R&S®CMW100
Communications Manufacturing Test Set
User Manual

1177541002

User Manual

Version 14

ROHDE&SCHWARZ

This user manual applies to the following variants of the R&S®CMW100 communications manufacturing test set:

- R&S®CMW100 variant K06 (1201.0002K06)
- R&S®CMW100 variant K03 (1201.0002K03)
- R&S®CMW100 variant K02 (1201.0002K02) plus R&S®CMW100Win (1201.0002K31)

The manual describes the base software and basic principles for manual operation and for remote control. The firmware applications are described in separate documents.

Mühldorfstr. 15, 81671 München, Germany
Phone: +49 89 41 29 - 0
Fax: +49 89 41 29 12 164
Email: info@rohde-schwarz.com
Internet: www.rohde-schwarz.com

1177.5410.02 | Version 14 | R&S®CMW100

The following abbreviations are used throughout this manual: R&S®CMW is abbreviated as R&S CMW.

Contents

1 Preface

1.1 Documentation Overview

This chapter provides an overview of the R&S CMW100 user documentation. Most documents are available in PDF format on the CMW customer web.

The CMW customer web is a section of GLORIS, the global Rohde & Schwarz information system, see https://gloris.rohde-schwarz.com. A registration is required.

1.1.1 Getting Started Manual

The getting started manual introduces the R&S CMW100 and describes how to set up and start working with the product.

A printed version of the manual is available as part of R&S CMW100-Z31.

A PDF file is available on the CMW customer web.

1.1.2 User Manual and Help

Separate user manuals are provided for the base unit and the firmware applications:

- The base unit manual describes the base software and common features of the firmware applications. It also describes basic principles for manual operation and remote control. The manual includes the contents of the getting started manual.
- Additional PDF files describe the firmware applications, including the graphical user interface of each application and the remote control commands.

All user manuals are available on the CMW customer web.

The contents of the user manuals are also available as help in the R&S CMW100. The help offers quick, context-sensitive access to the information needed for operation and programming. If you perform a software update, the help is also updated.

1.1.3 Features & Functions

The features & functions documents list the available software options and the related features, settings and measurement results. The documents are available on the CMW customer web.

1.1.4 Basic Safety Instructions

Contains safety instructions, operating conditions and further important information. The printed document is delivered with the instrument.

1.1.5 Data Sheet and Brochure

The data sheet contains the technical specifications of the R&S CMW100. It also lists the options and their order numbers and optional accessories.

The brochure provides an overview of the instrument and deals with the specific characteristics.

The documents are available on the CMW customer web.

1.1.6 Release Notes and Open-Source Acknowledgment (OSA)

The release notes list new features, improvements and known issues per software version and installation package.

The open-source acknowledgment provides verbatim license texts of the used open-source software.

The documents are available on the CMW customer web.

1.1.7 Application Notes, Application Cards, White Papers, etc.

These documents deal with special applications or background information on particular topics.

The documents are available on the CMW customer web.

1.2 Key Features

Modern cell phone production lines typically perform RF tests in non-signaling mode. The used RF analyzers and generators work in list mode, processing a list of predefined receiver and transmitter test steps as efficiently as possible. This approach results in a much better production line efficiency than the traditional approach of performing RF tests in signaling mode.

The R&S CMW100 follows this test approach. It is a non-signaling tester, supports list mode measurements and even allows parallel testing of several cell phones connected to a single radio test head.

A signal generated by the radio test head can be sent to up to eight receivers, so that the receivers can be tested in parallel (for example bit error tests). The received signals of up to eight transmitters can be measured sequentially, one signal at a time, in

parallel to the receiver tests. Measurements are typically performed in multi-evaluation list mode.

The radio test head is controlled by test software installed on a standard computer.

Key facts for the R&S CMW100:

- R&S CMWrun based production solution for different chipset suppliers
- Continuous frequency range up to 6 GHz
- Multitechnology solution, supporting for example:
  LTE, WCDMA/HSPA+, GSM/GPRS/EGPRS
  CDMA2000® 1xRTT/EV-DO, TD-SCDMA
  WLAN, Bluetooth®, LR-WPAN
- Parallel testing on up to eight RF ports
- High measurement performance
- High measurement accuracy
- Wide range of methods for reducing test time and maximizing capacity utilization
- High MTBF
- Minimum space requirements and footprint
- Silent and low weight
- Low energy consumption

1.3 R&S CMW Models

The R&S CMW software supports several instrument models. Depending on the model, only a subset of the software features is supported. The supported hardware options depend also on the instrument model.

The following table provides an overview of the differences between the R&S CMW100 and the R&S CMW500, the model that supports most features.

The table lists only the differences that are relevant in the context of this documentation. It helps you to identify features that are described in the R&S CMW documentation, but are not supported by the model R&S CMW100.

Table 1-1: General and hardware-related differences

	R&S CMW500	**R&S CMW100**		
		K06	**K03**	**K02+K31**
Smart card for licenses	internal	internal	internal	external
Support of external trigger signals	yes	yes	no	no
Support of external synchronization	yes	yes	no	no
RX bandwidth	80 or 160 MHz	160 MHz	40 MHz	
TX bandwidth			80 MHz	
Basic frequency range, minimum	70 or 150 MHz	RX 150 MHz TX 70 MHz	70 MHz	

	R&S CMW500	R&S CMW100		
		K06	K03	K02+K31
Basic frequency range, maximum	3.3 GHz	4 GHz		
Frequency range extension up to 6 GHz	R&S CMW-KB036	R&S CMW-K046		
Concept	embedded PC	separate PC		
Non-signaling mode (analyzer plus generator)	yes	yes		
Signaling mode (network emulation)	yes	no		
Audio testing	yes	no		
Data E2E testing	yes	no		
Internal fading	yes	no		
Instrument version with built-in display	yes	no		
Instrument version without built-in display	yes	yes		
Number of RF connectors	3 or 6	8+1[1)]		
Number of TX/RX paths	1, 2, 4	1[1)]		
Number of I/Q boards	0, 1, 2	0		
ARB generator	optional	basic functionality		
Support of R&S CMWS	yes	no		
[1)] The numbers apply per radio test head.				

The R&S CMW100 supports most measurement applications and the GPRF generator. It is a non-signaling tester.

The following table provides an overview of the supported firmware applications.

Table 1-2: Support of firmware applications

Firmware application	Support by R&S CMW100
5G NR sub-6 GHz measurements (KM6000)	yes
Bluetooth® measurements (KD611, KM61x, KM721)	yes
CDMA2000® measurements (KM8xx)	yes
GPRF generator, measurements[1)]	yes
GSM UL measurements (KM2xx)	yes
LR-WPAN measurements (KM680)	yes
LTE measurements (KM5xx, KN5xx)	yes
NB-IoT measurements (KM300)	yes
SCPI concurrency support tools	yes
TD-SCDMA measurements (KM75x)	yes
WCDMA measurements (KM4xx, KN4xx)	yes

Firmware application	Support by R&S CMW100
WLAN TX measurements (KM65x)	yes
Signaling applications (KSxxx)	no
In this table, all options are abbreviated for better readability ("R&S CMW-" omitted for R&S CMW-KMxxx). 1) Not supported: EPS measurements	

1.4 Using the Online Help System

This chapter describes how to use the online help system as context-sensitive help.

Opening the help system

You can open the online help via one of the following methods:

- Press the [HELP] key on the soft-front panel in single-window mode.
- Press ? on the soft-front panel in multiple-window mode.
- Right-click a screen element and select "Help".
- Press F1 on a connected keyboard.

The online help is context-sensitive. It opens at a position that is related to the currently displayed view, tab or dialog. You get better results if you select the relevant screen object before you open the help. If you want for example information about a specific tab, tap into that tab before you open the help.

Navigating within the help system

The help dialog box is divided into two parts. The right part displays the help contents (contents pane). The left part provides tabs for navigation within the help system (navigation pane with "TOC" tab, "Index" tab and "Search" tab).

The "TOC" tab lists the table of contents as a dynamic tree.

► Select a tree entry to display the topic in the contents pane.

The "Index" tab lists all index entries.

1. Type text into the field at the top and press ENTER to show only index entries starting with this text.
2. Select an index entry to display the topic in the contents pane.

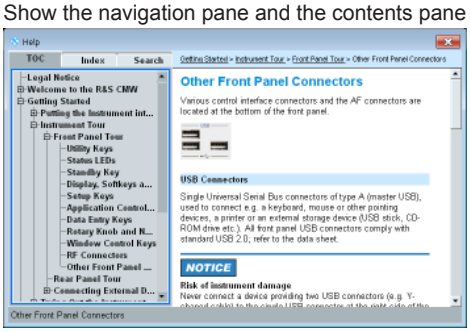
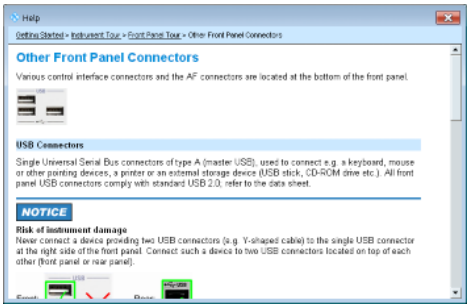
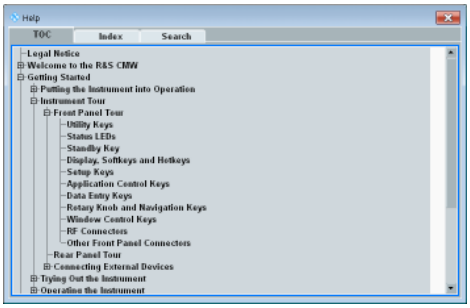
The "Search" tab provides a full-text search function.

1. Enter a search string. Then press ENTER to start the search.

   A list of results is displayed.
2. Select a result entry to display the topic in the contents pane.

Softkeys

The following softkeys are available on the right.

Toolbar button	Action
Home	Go to the start page
Back Forward	Browse the topics you visited before
Zoom In Zoom Out	Increase or decrease the text size
Tile	Show the navigation pane and the contents pane 
Content	Hide the navigation pane / show only the contents pane, for example to display large images or tables 
Index	Show only the navigation pane / hide the contents pane, for example to display long topic titles in the navigation tree 

2 Safety Information

The product documentation helps you to use the R&S CMW100 safely and efficiently. Follow the instructions provided here and in the "Basic Safety Instructions" brochure. Keep the product documentation nearby and offer it to other users.

Intended use

The R&S CMW100 is designated for the development, production and verification of electronic components and devices in industrial, administrative, and laboratory environments. Use the R&S CMW100 only for its designated purpose. Observe the operating conditions and performance limits stated in the data sheet.

Where do I find safety information?

Safety information is part of the product documentation. It warns you about the potential dangers and gives instructions how to prevent personal injuries or damage caused by dangerous situations. Safety information is provided as follows:

- The printed "Basic Safety Instructions" brochure provides safety information in many languages and is delivered with the R&S CMW100.
- Throughout the documentation, safety instructions are provided when you need to take care during setup or operation.

3 What's New in This Version

This documentation describes version 3.7.110 and higher of the R&S CMW100 base software. Compared to version 3.7.80, it provides the new features listed below.

- Remote commands for the system tray configuration, see Show System Tray Icon

4 Getting Started

This chapter contains the same information as the getting started manual.

4.1 Putting the Instrument into Operation

This chapter describes the basic steps to be taken when setting up the R&S CMW100 for the first time.

WARNING

Risk of injury due to disregarding safety information

Observe the information on appropriate operating conditions provided in the data sheet to prevent personal injury or damage to the instrument. Read and observe the basic safety instructions provided with the instrument, in addition to the safety instructions in the following sections. In particular:

- Do not open the instrument casing.

4.1.1 Unpacking and Checking the Instrument

Check the equipment for completeness using the delivery note and the accessory lists for the various items. Check the instrument for any damage. If there is damage, immediately contact the carrier who delivered the instrument.

Packing material

Retain the original packing material. If the instrument needs to be transported or shipped later, you can use the material to protect the connectors.

NOTICE

Risk of damage during shipment

Insufficient protection against mechanical and electrostatic effects during shipment can damage the instrument.

- When shipping an instrument, use the original packaging. If you do not have the original packaging, use sufficient padding to prevent the instrument from moving around inside the box.
- Pack the instrument in antistatic wrap to protect it from electrostatic charging.
- Secure the instrument to prevent any movement and other mechanical effects during transportation.

4.1.2 Positioning the Instrument on a Bench

Place the radio test head on a horizontal bench with even, flat surface. For details, observe the following sections.

Alternatively, you can mount the radio test head in a rack, see Chapter 4.1.3, "Mounting the Instrument in a Rack", on page 20.

4.1.2.1 Mounting the Feet

The delivery includes four instrument feet plus two M4 screws per foot.

Mount the feet as follows:

- For vertical positioning, you must mount two feet to the bottom of the instrument. The remaining two feet are not needed.
  Mount one foot to the bottom front end and one foot to the bottom rear end.

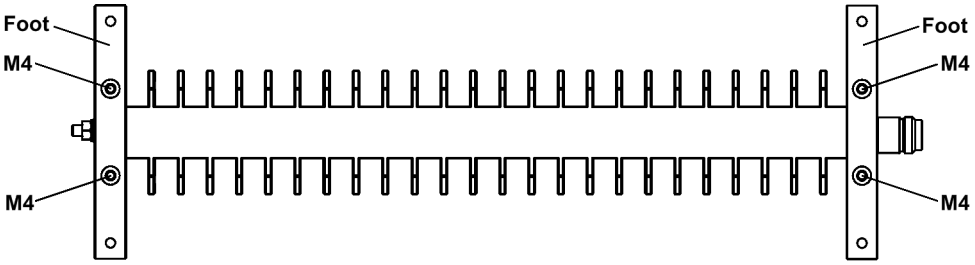


Figure 4-1: View from below with two mounted feet

- For horizontal positioning, you must mount all four feet.
  Mount two feet to the bottom of the instrument, as for horizontal positioning. Mount the other two feet in a similar way to the top of the instrument.
  When you lay the instrument on one side, it is now supported by all four feet.

The following figure shows vertically and horizontally positioned instruments with mounted feet.

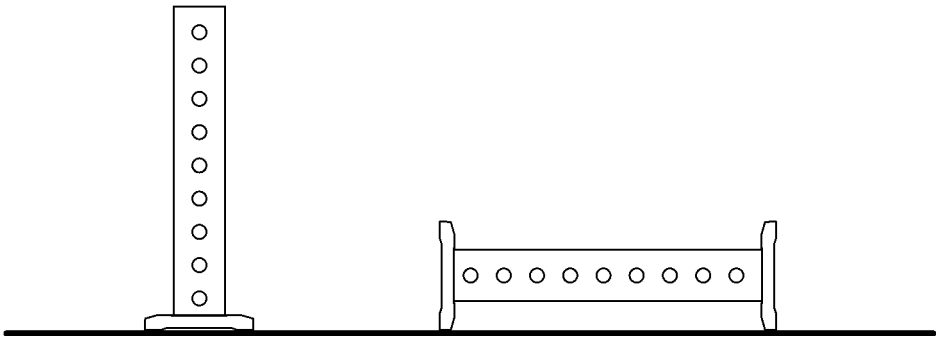


Figure 4-2: View from the front, vertical and horizontal positioning

4.1.2.2 Securing the Instrument against Toppling

CAUTION

Risk of injury due to toppling instrument

A vertically positioned instrument can tilt over and cause injury.

Ensure a stable stand of the instrument. Ensure that the instrument cannot accidentally be knocked over.

For vertical positioning, it is recommended, to mount the instrument feet to the bench. For this purpose, each foot provides two holes, suitable for M4 screws.

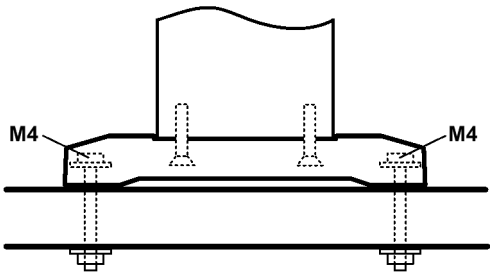


Figure 4-3: Foot mounted to bench

For securing the top of the instrument, the top rear provides a hole with an M4 thread.

4.1.2.3 Ensuring Sufficient Airflow and Cooling

NOTICE

Risk of instrument damage due to overheating

An insufficient airflow can cause the radio test head to overheat, which can impair the measurement results, disturb the operation, and even cause damage.

Follow the instructions in this section to ensure sufficient cooling.

Preferably, use the instrument in vertical position. The vertical position allows better cooling than the alternative horizontal position.

The instrument is air-cooled via its surface. Both sides carry cooling fins.

Observe the following rules to ensure sufficient cooling:

- Keep the cooling fins clean. Dust impedes cooling.
- Ensure that cool air can reach the bottom of the instrument and rising warm air can easily move away from the instrument. Especially do not place the instrument in a closed compartment.
- A minimum horizontal distance is required between the instrument sides and other objects, instruments or walls.

For vertical positioning, the minimum distance is 5 cm. For horizontal positioning, it is 15 cm.

- A minimum vertical distance of 15 cm is required between the instrument top and obstacles that block rising warm air.

The following figure illustrates the minimum distances.

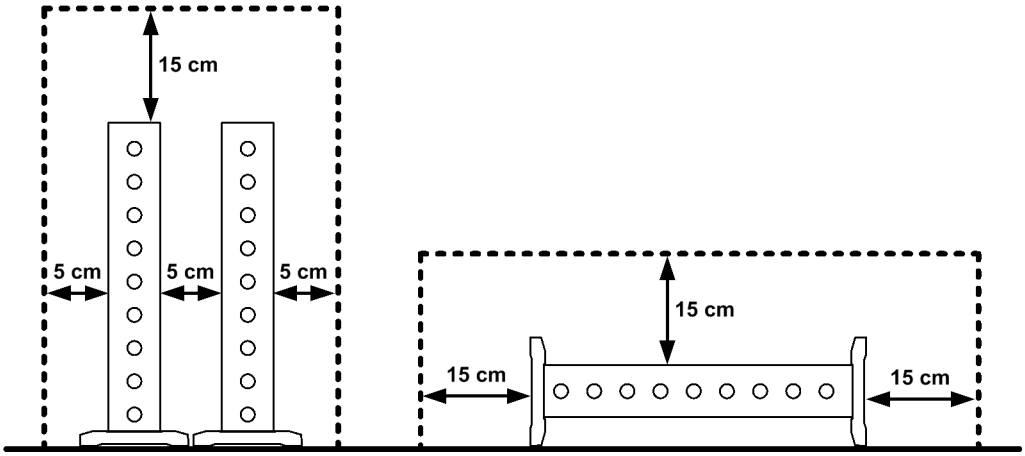


Figure 4-4: Minimum distances for horizontal and vertical positioning

4.1.3 Mounting the Instrument in a Rack

You can mount the radio test head in a 19" rack, using the rack mount kit R&S CMW100-Z19 (order number 1210.7470.xx).

NOTICE

Risk of instrument damage due to insufficient airflow in a rack

An insufficient airflow can cause the radio test head to overheat, which can impair the measurement results, disturb the operation, and even cause damage.

Observe the instructions provided by your rack manufacturer and the installation instructions delivered with the rack mount kit.

4.1.4 Grounding the Instrument

NOTICE

Risk of instrument damage caused by ESD or surge voltages

The instrument is susceptible to damage from an ESD event (electrostatic discharge) and surge voltages, for example caused by power system transients or lightning transients.

To protect the instrument, ground it via a low-resistance connection. Use the ground terminal labeled ⏚ for that purpose.

4.1.5 Connecting the Power Supply

NOTICE

Risk of instrument damage due to inappropriate operating conditions

An unsuitable operating site or test setup can damage the instrument and connected devices. Ensure the following operating conditions before you connect the power supply:

- The instrument is positioned correctly, see:
  - Chapter 4.1.2, "Positioning the Instrument on a Bench", on page 18
  - Chapter 4.1.3, "Mounting the Instrument in a Rack", on page 20
- The instrument is dry and shows no sign of condensation.
- The ambient temperature does not exceed the range specified in the data sheet.
- For any connected cables:
  - Signal levels at the input connectors are within the specified ranges.
  - Signal outputs are correctly connected and are not overloaded.

An external AC/DC power supply adaptor is included in the delivery. It adapts itself automatically to the AC supply voltage. The supported voltage range is stated on the rear panel and in the data sheet. It includes, for example, the typical AC voltages 110 V and 220 V.

⚠ WARNING

Risk of injury and instrument damage due to wrong power supply

Connecting an inappropriate power supply can cause an electric shock and damage the instrument.

Use only the power supply R&S AC adaptor 3592.7717.00. It is delivered with the radio test head and can be ordered as spare part R&S CMW-Z13.

NOTICE

Risk of instrument damage when connecting or disconnecting the power supply

If the power supply is connected to a voltage source, connecting the power supply to the radio test head can damage the instrument. The same applies to disconnecting the power supply from the radio test head.

Ensure that the power supply is disconnected from the voltage source, before connecting the power supply to the radio test head and before disconnecting the power supply from the radio test head.

Connect the power supply as follows:

1. Connect the external power supply to the DC IN jack of the radio test head.
2. Connect the external power supply to a power outlet.
3. Check that the Power LED is on.
   If it is off, check the power outlet, the external power supply and the connections. The external power supply has an LED, indicating whether the power supply works.

You can leave the power supply connected permanently. However, to save energy and costs, you can disconnect the external power supply from the power outlet, if the instrument is not in use.

After connecting the instrument to the power supply, the internal OCXO needs some minutes to warm up. For accurate measurements, wait until the LED OCXO Ready is lit.

⚠ CAUTION

Risk of burn when touching the radio test head

Take care when touching the radio test head. The surface of the instrument can become hot during operation.

4.1.6 Connecting the Computer

The measurement software is running on a computer, connected to the radio test head.

4.1.6.1 Computer Requirements

The computer to be connected to the radio test head must fulfill the minimum requirements stated in the following table.

A compatible computer is available as option R&S CMW100CTR.

	Minimum requirements	Recommended
Processor (CPU)	Intel Core i3, 3rd generation	Intel Core i7, quad-core, 4th generation
Memory (RAM)	8 GB	16 GB
Storage medium[1)]	64 GB free space, HDD or SSD	128 GB free space, SSD
Graphics	Minimum resolution 1024 x 768	
Operating system	Windows 7 Professional 64-bit or Windows 10, supporting English language The compatibility of other Windows versions cannot be guaranteed.	
USB socket for connection of radio test head	1x USB 3.0 Use a socket mounted to the motherboard, not a socket connected to the motherboard via a cable. Sockets on the PC front are often not suitable.	
Only for variant K02: USB socket for license smart card reader	1x USB 2.0 or USB 3.0	
[1)] The free disk space on the storage medium is, for example, used for installation of the test software, storage of measurement results and ARB files.		

Virus protection

An efficient virus protection is a prerequisite for secure operation in a network. Notice the following recommendations when using anti-virus software:

- During tests, do not download signature files from the Internet and do not perform system drive scans. Such actions can affect the performance of the instrument.
- During tests, disable virus protection of the system drive.

Compass version conflicts

Note that the test software is based on the R&S Compass platform. To avoid version conflicts, do not install other R&S Compass-based software on the same computer.

For example, do not install the R&S WinIQSIM2™ software.

4.1.6.2 Connecting the Computer to the Instrument

A USB cable is included in the delivery. Connect it to the Control jack of the radio test head and to a USB 3.0 jack of the computer.

The entire communication between the radio test head and the controlling computer is done via the USB connection. Thus the performance of the USB connection is important.

Observe the following hints:

- Use a USB jack mounted to the motherboard of the computer, not a jack connected to the motherboard via a cable. USB jacks at the front of a computer are often not suitable.
- Use only the USB cable included in the delivery or a USB cable delivered by Rohde & Schwarz as accessory for the R&S CMW100. Do not use another cable, especially not a longer cable or a cable with poor shielding.

You can connect up to four radio test heads to the same computer. The radio test heads must be of the same variant (for example, only K06).

4.1.6.3 Connecting the Smart Card Reader (Variant K02+K31)

This section applies only to variant K02. All other variants come with an integrated smart card.

For variant K02, you need a K31. The K31 includes a smart card with the purchased licenses and a smart card reader. You need only one K31 per control computer, even if you connect several radio test heads to the same computer. The smart card must contain all required licenses for all connected radio test heads.

The smart card is delivered as full-size smart card (credit card size) containing a mini smart card (SIM size).

You can use the full-size smart card with a suitable card reader, for example the OMNIKEY® 3121 USB smart card reader. Or you can break out the mini smart card and use it with the portable USB smart card reader included in the delivery.

Figure 4-5: Delivered smart card reader with partly inserted smart card

NOTICE

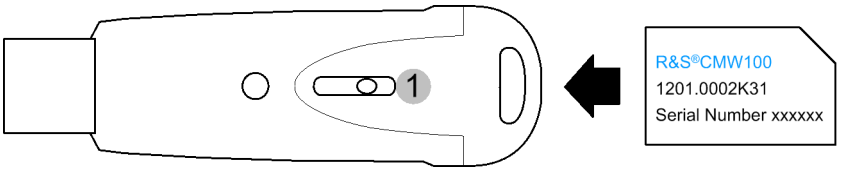
Risk of smart-card damage

The mini smart card can be damaged by adhesives or if too much stress is applied to the chip.

Avoid strong bending of the chip when breaking-out the mini smart card. Never try to reinstall the mini smart card into the full-size smart card.

If you want to use the delivered USB smart card reader, insert the mini smart card into the slot at the rear end of the reader. Observe the correct orientation of the smart card

as indicated in the following figure. The cut corner points away from the card reader. The contacts of the smart card are facing down, so they are not visible in the figure.



In the case that you need to remove the smart card from the smart card reader, use the small pusher labeled (1) in the figure.

Connect the smart card reader with inserted smart card to a USB socket of the computer.

NOTICE

Risk of theft or loss

The smart card contains the purchases licenses. So it is valuable.

Secure the smart card against theft, especially when it is not used.

4.1.7 Installing the Test Software

This section describes the first installation of the test software and the first software start, including the verification of the successful installation.

4.1.7.1 Software Packages

The R&S CMW100 software consists of the mandatory CMW base software package plus optional packages for firmware applications and utilities. These packages are integrated in setup files named `Setup_CMW1xx_<Scope>(Release)_<Version>.exe`.

For the base software package, `<Scope>` equals BASE, resulting in the file name `Setup_CMW1xx_BASE(Release)_<Version>.exe`.

You can purchase a USB memory stick with the setup files. It is included in the delivery if you order an R&S CMW-S31A (K02 only) or an R&S CMW100-Z31 (all other variants).

You can also download the software from the CMW customer web.

Additional waveform (ARB) files are grouped into library packages and provided as self-extracting files or zip-files. They are only relevant if you want to use the functionality of the arbitrary RF generator. Download the needed libraries from the CMW customer web and unpack them to the system drive. The preferred file location is the folder `waveform` in the CMW user data directory (for example `C:\ProgramData\Rohde-Schwarz\CMW\Data\waveform`).

4.1.7.2 Installation Procedure

This section describes how to install and start the test software for the first time. For a software update, you must first uninstall the software parts to be updated, see Chapter 4.5, "Software Update", on page 44.

User account

Use the same user account for installation and usage of the test software. This account must have administrator rights.

Proceed as follows to install the test software:

1. Ensure that the Microsoft user account control (UAC) is inactive.
   a) In the Windows "Start" menu, enter the search term "UAC".
   b) Click the search result "Change User Account Control Settings".
   A dialog box opens.
   c) In the dialog box, move the slider to the lowest position ("Never notify"). Close the dialog box via the OK button.
   d) If you have changed the setting, perform a reboot.
2. Copy all relevant setup files to the same directory. Use a directory on the system drive or on a USB memory stick.
3. Start the installer:
   a) Right-click one setup file, for example `Setup_CMW1xx_BASE(Release)_<Version>.exe`.
   b) Select "Run as administrator".

   The "R&S Software Distributor" opens.

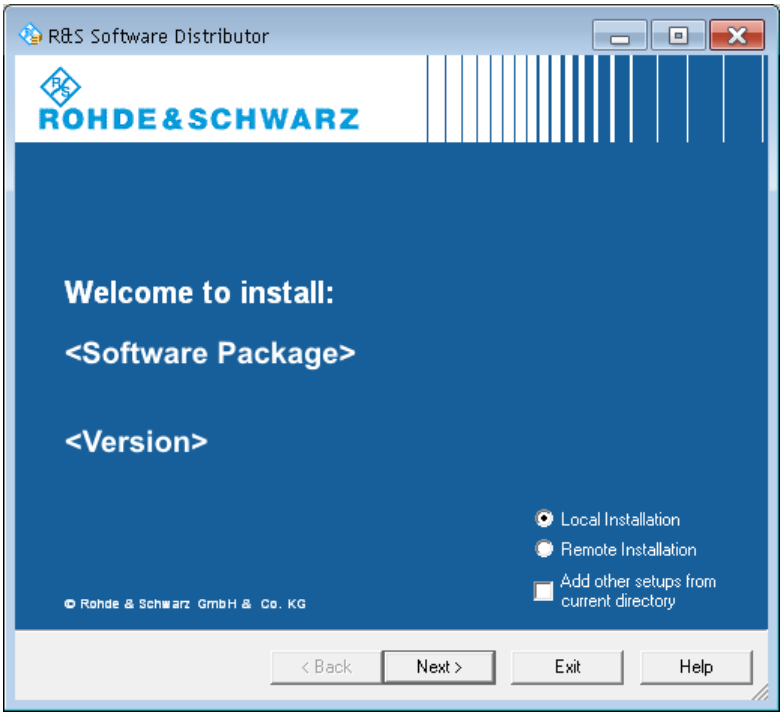


Figure 4-6: R&S software distributor

4. Enable "Add other setups from current directory".
5. Click "Next".
6. Select which packages you want to install.
7. Click "Install".
8. Wait until the installation is complete.
9. Reboot the computer.
10. Right-click the "CMW" icon on the desktop.

11. Select "Properties > Compatibility" and enable "Run this program as an administrator".

The test software is now ready to operate.

4.1.7.3 Starting the Software for the First Time

Before you start the test software, ensure that:

- The radio test head is connected to the power supply, see Chapter 4.1.5, "Connecting the Power Supply", on page 21.

- The USB cable is connected, see Chapter 4.1.6.2, "Connecting the Computer to the Instrument", on page 23.
- For variant K02, that the smart card reader with inserted smart card is plugged in, see Chapter 4.1.6.3, "Connecting the Smart Card Reader (Variant K02+K31)", on page 24.

If you encounter problems during the following steps, see Chapter 4.1.7.4, "Trouble-shooting", on page 29.

To start the test software:

1. Double-click the "CMW" icon on the desktop.
2. The first start of the software performs a firmware update of the radio test head. The update can take some time. Wait until it is complete.
   If you do not see a "Firmware Update" dialog box during the first start, the software does not recognize the radio test head.
3. After the firmware update, wait until the normal startup procedure is complete.

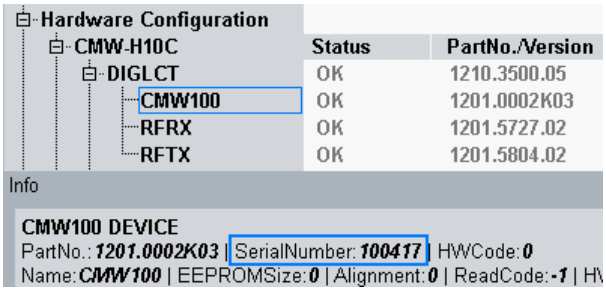
To check the successful installation and hardware setup:

1. Press the button.

   The "Setup" dialog box opens.
2. On the left, select "License Keys".
3. Check that a "Device Identification" is displayed.

   If no ID is displayed, the smart card with the licenses is not recognized.
   The licenses on a detected smart card are listed under "Active License Keys".
4. On the left, select "SW/HW-Equipment".
5. Check the section "Hardware Configuration".

   There must be an entry "CMW-H10x" for each connected radio test head.
   The subentry "CMW100" shows the part number including the variant. The info section of the subentry shows the serial number.



6. Check the section "Installed Software".

   If the installed firmware applications are recognized, they are listed.

Installed Software	Version	Status
CMW 1xx BASE	V 3.5.20	OK
CMW BLUETOOTH Meas	V 3.5.10	OK
CMW CDMA2000 Meas	V 3.5.11	OK
CMW GPRF Gen	V 3.5.10	OK
CMW GPRF Meas	V 3.5.10	OK

To perform first tests, see Chapter 4.3, "Trying Out the Instrument", on page 33.

4.1.7.4 Troubleshooting

This section provides troubleshooting hints for the first software start.

Instrument not recognized by test software

If the radio test head is not recognized, proceed as follows:

1. Power off and on the radio test head:
   a) Disconnect and reconnect the external power supply.
   b) Wait some seconds to let the radio test head start up.
   c) Restart the test software.
   d) Check the "Setup" dialog.
2. If that does not help:
   a) Try another USB port at the computer. For troubleshooting, you can even try a USB 2.0 port to check whether it works at all. For testing, always use a suitable USB 3.0 port (see Chapter 4.1.6.2, "Connecting the Computer to the Instrument", on page 23).
   b) Restart the test software.
   c) Check the "Setup" dialog.

Access to SCardSvr service failed

If the test software aborts the startup and indicates "access to SCardSvr service failed", you have started the test software without administrator rights.

Configure the start icon so that the software is always started with administrator rights, see end of Chapter 4.1.7.2, "Installation Procedure", on page 26.

No license keys available (variant K02 only)

If no licenses are listed in the "Setup" dialog, the licenses smart card is not detected properly.

1. Check that the smart card reader is plugged in.
2. Check the correct insertion of the smart card into the reader. See Chapter 4.1.6.3, "Connecting the Smart Card Reader (Variant K02+K31)", on page 24.

3. Ensure that the test software starts with administrator rights.
4. Check whether the Windows device manager detects the smart card reader (OMNIKEY entry or smart card reader entry).
   You must have started the test software with administrator rights before checking the device manager.

4.1.8 EMI Suppression

Electromagnetic interference (EMI) can affect the measurement results.

To suppress electromagnetic radiation:

- Use suitable shielded cables of high quality. For example, use double-shielded RF cables.
- Always terminate open RF ports and cable ends.
- Note the EMC classification in the data sheet.

4.2 Instrument Tour

This chapter provides an overview of all connectors and control elements of the radio test head.

The front panel of the instrument is equipped with nine RF ports. The rear panel provides a USB jack, four status LEDs, a power input jack and a ground terminal.

The rear panel of the variant K06 provides also two trigger ports and two system synchronization ports.

Figure 4-7: Front panel, variant K02/K03 rear panel and variant K06 rear panel

1 = RF Ports, page 31
2 = Control Jack, page 31
3 = Status LEDs, page 31
4 = DC IN Jack, page 32

5 = Ground Terminal, page 32
6 = M4 Thread, page 32
7 = Ref 10 MHz Ports (K06), page 32
8 = Trigger Ports (K06), page 33

4.2.1 RF Ports

The SNAP N-type radio frequency (RF) ports are used as inputs and outputs for RF signals. The impedance of all RF ports is 50 Ω.

RF 1 to RF 8 are bidirectional ports for RF connections to and from the devices under test (DUT).

TX AUX is a unidirectional output port, supporting a higher maximum output level than the other ports. You can use this port, for example, when you measure external attenuations in your test setup.

In output direction, you can provide a generated signal at all RF n ports in parallel, or at a configurable subset of the RF n ports, or at TX AUX.

In input direction, you can measure a signal at a single RF n port at a time.

The LED above each port indicates the port state:

- LED green: output direction active, signal available at the port
- LED red: input direction active, ready to receive a signal via this port
- LED orange: both directions active

NOTICE

Risk of instrument damage due to high input power

Ensure that the maximum input levels of the RF ports are not exceeded. They are stated in the data sheet.

Exceeding the maximum levels can damage the instrument.

4.2.2 Control Jack

The Control jack is a USB 3.0 type B jack. It is used to establish a USB connection between the controlling computer and the radio test head, see Chapter 4.1.6.2, "Connecting the Computer to the Instrument", on page 23.

4.2.3 Status LEDs

The status LEDs have the following meaning:

- Status:
  Used by Rohde & Schwarz personnel only.
- OCXO Ready:

The internal oven quartz has warmed up. This state is a prerequisite for accurate measurements. After connecting the instrument to the power supply, the OCXO needs some minutes to warm up.

- Power:
  The instrument is correctly supplied with power.
- Power Fail (K02, K03), Power Rail 1/2 Fail (K06):
  If at least one of these LEDs is lit, there is a failure in the power input module. For repair, contact the Rohde & Schwarz service.

4.2.4 DC IN Jack

The instrument is supplied with power via the DC IN jack, see Chapter 4.1.5, "Connecting the Power Supply", on page 21.

4.2.5 Ground Terminal

Ground the instrument via the ground terminal (labeled ⏚) and a low-resistance connection, see Chapter 4.1.4, "Grounding the Instrument", on page 21.

4.2.6 M4 Thread

There is a hole with an M4 thread at the top of the instrument rear. You can use it to secure a vertically positioned instrument against toppling.

See also Chapter 4.1.2.2, "Securing the Instrument against Toppling", on page 19.

4.2.7 Ref 10 MHz Ports (K06)

At the instrument variant K06, there is one BNC port for the input of reference signals and one BNC port for the output of reference signals. The arrows indicate the direction (signal into the port or out of the port).

If you want to synchronize the R&S CMW100 to an external reference signal, feed a 10-MHz signal to the input port. The external reference signal must meet the specifications in the data sheet.

Whether the internal source or the input port is used, is configurable. See "Setup" dialog > "Sync" > "Frequency Source".

The LED labeled "Ref In 10 MHz Locked" indicates that the synchronization to an external reference signal is successful. If the LED is off, possible reasons are:

- The instrument is configured to use its internal reference frequency source, not the input port.
- There is no external signal or its level is too low or its frequency is out of range.

The output port provides the used 10-MHz reference signal, coming from an internal source or from the input port.

Output cabling

Use double-shielded cables and match the signal with 50 Ω to comply with EMC (electromagnetic compatibility) directives.

For settings, see Chapter 6.5.8, "Sync Settings (K06)", on page 106.

4.2.8 Trigger Ports (K06)

At the instrument variant K06, there is one BNC port for the input of trigger signals and one BNC port for the output of trigger signals. The arrows indicate the direction (signal into the port or out of the port).

Input signal: An external trigger input signal must be an LVTTL/LVCMOS (3.3 V) signal with a rise/fall time below 5 ns. The trigger input is high impedance.

Output signal: The trigger output has an LVTTL/LVCMOS (3.3 V) signal. The output impedance is approximately 50 Ω. Evaluate the rising edge of the generated positive pulses.

Trigger cables

Long trigger cables can cause signal reflections.

For settings, see Chapter 6.5.9, "Trigger (K06)", on page 106.

4.3 Trying Out the Instrument

This chapter helps you to get familiar with the R&S CMW100. It provides some sample sessions with manual control of the instrument.

As a prerequisite, the instrument must be set up and started up. The required devices must be connected.

For additional information, see:

- Chapter 4.1, "Putting the Instrument into Operation", on page 17

Reset

To obtain predictable results, it is recommended to reset the R&S CMW100 to a definite state before you try out the examples in this chapter. Press [RESET] to open the "Reset" dialog and perform a preset.

4.3.1 Generating an RF Signal

Generators provide RF signals for test purposes. The following example uses the general-purpose RF (GPRF) generator.

The GPRF generator provides an RF signal at constant frequency or at a series of configurable frequencies and levels. It is also possible to generate an RF signal that is modulated using a waveform file. All RF signals are configured in a similar way.

As an example, we generate a dual-tone signal at constant frequency. Proceed as follows:

1. Enable the "GPRF Generator" application:
   In the main window, enable the checkbox behind the "General Purpose RF Generator" entry.
2. In the "GPRF Generator" application, configure the RF routing. In this example, we use only port RF 1 for signal output. You could also feed the generated signal to several RF ports in parallel.
   a) For "Scenario" and "Routing", use the default settings.
   b) If your test setup contains a known, frequency-independent attenuation, enter the value as "Ext. Att. (Output)".
   c) Disable the signal output for all but the first RF port.

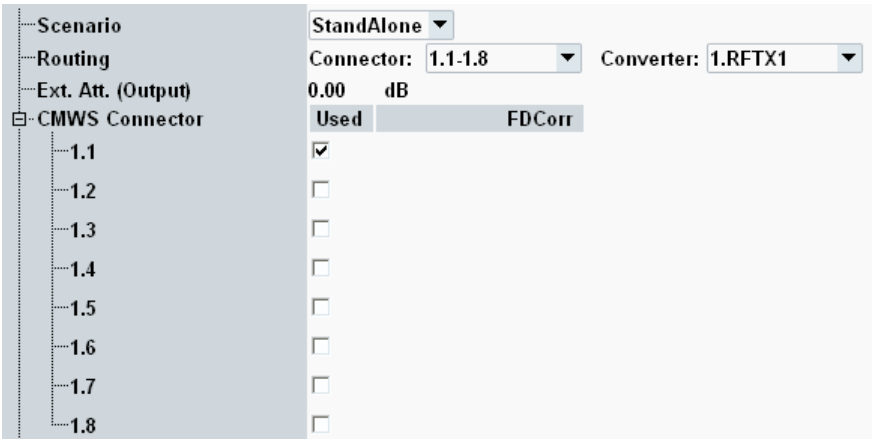


Figure 4-8: Resulting settings

3. Configure the signal properties.
   a) Select the "Frequency" (1200 MHz) and "Level (RMS)" (-30 dBm) of the RF output signal.
   b) Ensure that the "List Mode" is off.
   c) Select the "Baseband Mode" "Dual Tone".
   d) In the "Baseband Configuration > Dual Tone" section, configure the properties of the dual-tone signal. To superimpose two CW signals at different frequencies and levels, set the "Offset Frequency" of both signals and define a "Ratio" that is different from 0 dB; see figures below.

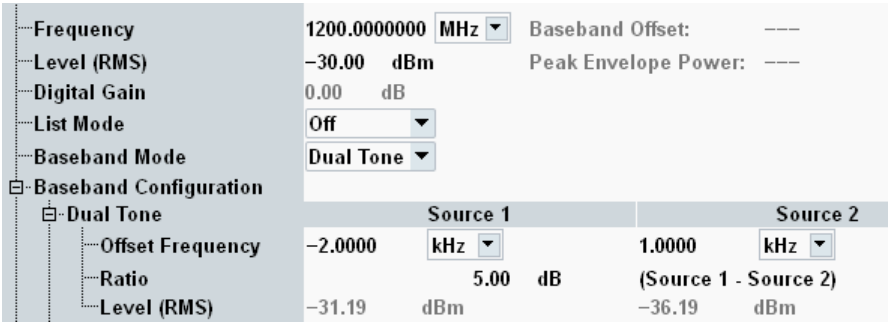


Figure 4-9: Resulting settings

4. To switch on the RF generator, right-click the "GPRF Generator" softkey and click ON.

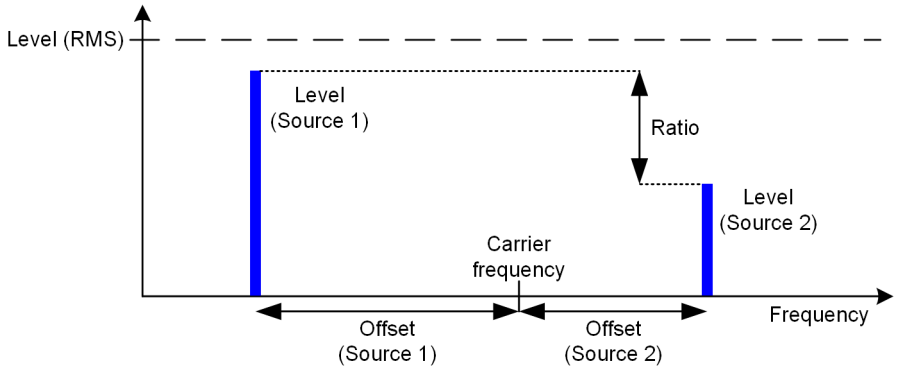


Figure 4-10: Dual-tone generator signal

To tap the resulting RF generator signal, proceed with Chapter 4.3.2, "Measuring an RF Signal", on page 35.

ARB generator

As an alternative, the GPRF generator application provides the arbitrary ("ARB") baseband mode. The ARB generator signal is based on a "waveform file" which is loaded and processed at runtime. For details, refer to the "GPRF Generator" description.

4.3.2 Measuring an RF Signal

The R&S CMW100 provides various general purpose and network-specific measurements. All measurements are controlled in an analogous manner. The following example uses the general-purpose RF (GPRF) power measurement.

The GPRF power measurement measures a series of power steps at (possibly) different powers and frequencies and performs a statistical evaluation. As a simple example, we can measure the RF signal generated by the GPRF generator of the R&S CMW100. You need a cable suitable to connect two RF ports.

Proceed as follows:

1. Configure the GPRF generator as described in Chapter 4.3.1, "Generating an RF Signal", on page 33.
2. Connect a coax cable between the RF ports RF 1 and RF 2 on the front panel of the R&S CMW100 to feed the generator signal to RF 2.
3. Enable the "GPRF Measurements" application:
   In the main window, enable the checkbox behind the "General Purpose RF Measurements" entry.
4. Press the softkey-hotkey combination "RF Settings > RF Routing". Select RF 2 as connector.
5. To start the power measurement, right-click the "Power" softkey and click ON.
6. In the "Settings" panel of the measurement dialog, adjust the "Frequency", the "Expected Nominal Power", and the filter "Bandwidth" to the properties of your input signal.
7. Observe the measurement result on the screen.

In the present example, the upper tone (at 1200.001 MHz) of the generated dual-tone signal is observed in a 1 kHz bandwidth. Due to the cable loss, the measured power is a bit smaller than the tone level shown in the generator dialog box.

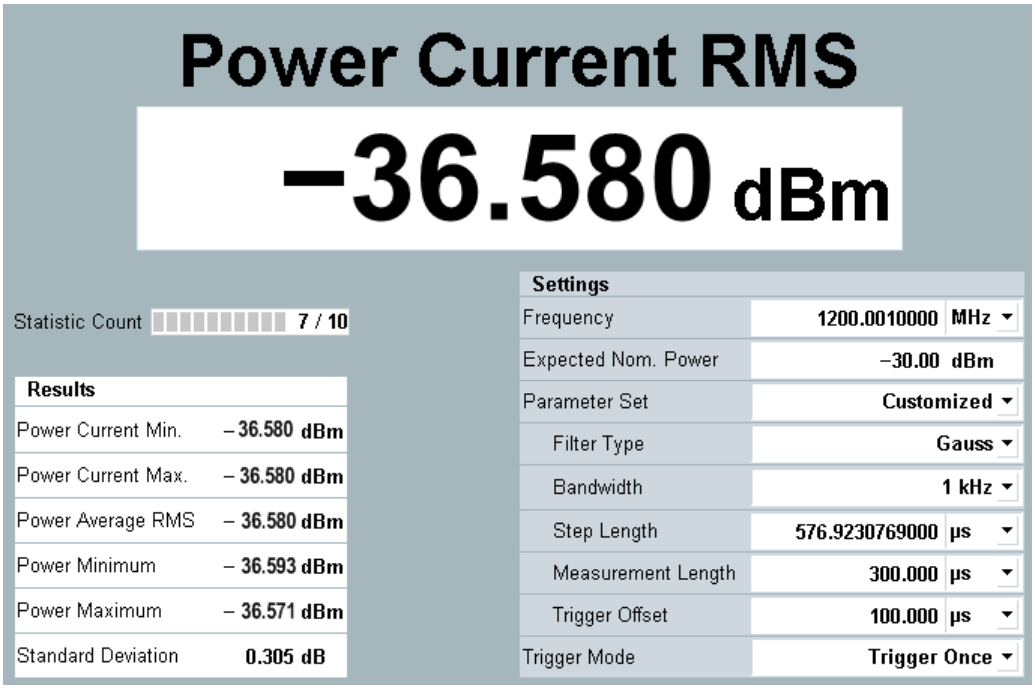


Figure 4-11: RF power results

4.4 Operating the Instrument

You can operate the instrument manually or via remote control:

- Remote control:
  You create scripts to automate repeating settings, tests and measurements. The scripts are executed on a computer controlling the instrument.
  For an introduction, see Chapter 7, "Remote Control", on page 125.
- Manual operation:
  You control the instrument via a graphical user interface (GUI), using a mouse and a keyboard.
  The principles of manual operation are explained in this chapter.

4.4.1 Accessing Applications

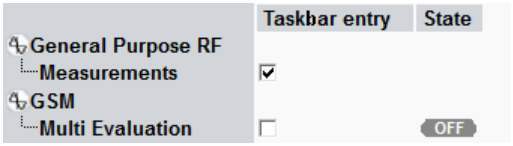
Before you can use an application, for example a measurement, you must add the application to the task bar. By default, all applications are disabled and the task bar is empty.

The following sections describe how to add applications to the task bar, depending on the used mode.

Accessing applications in multiple-window mode

In this mode, the installed and licensed applications are listed directly in the main window.

► Select the applications for which you want to display an entry in the task bar.



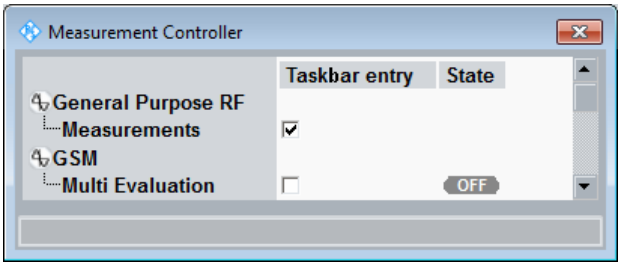
For each selected application, a separate window is opened. To rearrange these windows, see Chapter 4.4.2, "Organizing Multiple Windows", on page 38.

Accessing applications in single-window mode

In this mode, the applications are listed in two dialog boxes. One dialog box lists applications that measure an input signal. The other dialog box lists applications that generate an output signal. Both dialog boxes are used in the same way.

1. Press [MEASURE] to open the "Measurement Controller" dialog box or [SIGNAL GEN] to open the "Generator/Signaling Controller" dialog box.

   The dialog boxes list the installed and licensed applications.



2. Select the applications for which you want to display an entry in the task bar.
3. To close the dialog box, press [MEASURE] / [SIGNAL GEN] again or press [ESC]. Alternatively, select an application at the bottom of the GUI. As a result, the dialog box closes and the main view of the selected application opens.

In single-window mode, only one application is displayed at a time. To switch between the applications, proceed as follows.

1. To display the task bar across the bottom of the GUI, press the [TASKS] key. Alternatively, move a connected mouse to the lower border of the GUI. Wait some seconds.
   Or right-click the hotkey-bar at the bottom and select "Show tasks".
2. Press one of the hotkeys to access the associated application.

4.4.2 Organizing Multiple Windows

In multiple-window mode, several windows can be open in parallel. This section describes how to rearrange and resize these windows.

The main window provides the following buttons for window management:

- Buttons for main window toolbar handling
  - hides the main window. The soft-front panel buttons are still accessible via a vertical toolbar at the edge of the screen.
  - restores the main window.
  - moves the vertical toolbar to the other side of the screen.
- "Bring to front" button

  Brings all foreign windows to the back. Foreign windows are windows that have not been opened by the R&S CMW100 software, for example windows of the operating system or of other software programs.
- "Cascade" button

  Cascades all application windows. The main window is displayed separately.
- "Tile fixed size" button

Tiles all application windows and the main window. All windows get the same predefined size. If there are more windows than fit on the screen, the remaining windows are cascaded.

- "Tile maximum size" button
  

  Similar as the previous button, but using the entire screen for tiling. The effect is like using the previous button and then increasing the size of all windows.

Cascading and tiling affects all main application windows, for example generator and measurement main windows and some other selected windows, like the "Setup" dialog box.

Other windows and dialog boxes are not rearranged or resized automatically. Examples for such windows are the help window, the print dialog box and the main configuration dialog box of all applications. Position and resize them manually as desired. Their position and size is maintained even if you press a tile or cascade button.

It can be especially helpful to move the main configuration dialog box out of the related application window and to increase the size of the dialog box.

Reordering tiled windows

If the window sequence after tiling is not as desired, you can reorder the windows as follows.

1. Roughly move the windows to their desired position, for example exchange two windows.
2. For alignment of the windows, press the tile button again.

   For each target position, the tile mechanism uses the nearest window. So the manually created window order is maintained.

4.4.3 Accessing Dialogs

Accessing dialogs within an application

Depending on the application, dialogs are accessed via softkey-hotkey combinations, via the "Config" hotkey or via buttons within the GUI.

Most settings of an application are listed in a single dialog box, opened via the "Config" hotkey. Or they can be configured directly in the main view of the application.

► Press the "Config" hotkey in the bottom right corner of the GUI to open the main configuration dialog box of an application.

Using softkeys and hotkeys

The softkeys are on the right of the GUI, the hotkeys at the bottom of the GUI. The present softkeys and hotkeys are adapted dynamically, depending on the application and the context, for example on configured parameters. Hotkeys also change depending on the active softkey.

To use a softkey-hotkey combination, proceed as follows:

1. Press the softkey.

   The hotkeys associated with the softkey are displayed across the bottom of the GUI.

2. Press the hotkey.

   The hotkey triggers an action like opening a dialog box or toggling a setting.

4.4.4 Entering Data via the On-Screen Keyboard

You can enable an on-screen keyboard in the "Setup" dialog (system settings section). The keyboard is especially useful if you have a touchscreen connected. You can also use the on-screen keyboard with a mouse. This section assumes a connected touchscreen and uses the corresponding terms (for example "tap" instead of "click").

There are several variants of the keyboard. This section explains the two most important variants: An on-screen keypad for entry of numeric values and units. And an on-screen keyboard with English layout for text input.

To enter values with the on-screen keypad

1. Tap a numeric entry field to open the on-screen keypad.

2. Enter the numeric value as you would on a normal keypad.

   Special buttons:
   - "±" changes the sign of the value.
   - The unit buttons convert the number to another unit.
   - "Insert" toggles between insert and overwrite mode.
   - The bold arrow button acts as backspace.
   - "CLR" deletes the entire entry. "Del" deletes the character at the cursor position.
   - The arrow left/right buttons move the cursor left or right.

- The arrow up/down buttons increase/decrease the value at the cursor position.

3. Tap "Enter" to apply or "Esc" to discard the changes. The keypad closes in both cases.

To enter data with the on-screen keyboard

1. Tap a text entry field to open the on-screen keyboard.

2. Enter the text as you would on a normal keyboard.

   Special buttons:
   - "Last" discards the changes and displays the text applicable before opening the keyboard.
   - "Shift" toggles between two sets of characters. The keyboard displays the currently active set.

   - "CLR" deletes the entire entry.
   - "Insert" toggles between insert and overwrite mode.
   - The arrow up/down buttons increase/decrease the value at the cursor position.

3. Tap "Enter" to apply or "Esc" to discard the changes. The keyboard closes in both cases.

4.4.5 Entering Data via an External Keyboard

This section describes how to access input fields, enter numeric values or character data and toggle checkboxes using only a keyboard (no mouse).

Navigating within a dialog

► Press the UP / DOWN and LEFT / RIGHT arrow keys on the keyboard to step through the elements in the dialog.

Navigating between tabs

► Press SHIFT + TAB / TAB to switch to the previous tab to the left or to the next tab to the right.

Selecting a value from a pull-down list

1. Press ENTER to activate the pull-down list.
2. Use the UP / DOWN arrow keys to step through the list.
3. Press ENTER to select the current entry and deactivate the pull-down list.

Entering a numeric value or character data

1. Press ENTER to activate the input field.
2. Enter/modify the field contents:
   - Use the corresponding keys to enter characters or numbers.
   - Use the UP / DOWN arrow keys to increment/decrement a digit.
   - Use the LEFT / RIGHT arrow keys to move the cursor within the input field.
   - Use the minus key to change the sign of a number.
   - Press CTRL + F9 / F10 / F11 / F12 to multiply a numeric value with factors of 1, $10^{(-)3}$, $10^{(-)6}$, $10^{(-)9}$ and add the appropriate physical unit.
   - Press BACKSPACE to correct an entry.
3. Press ENTER to confirm the entry and deactivate the input field.

Selecting or clearing a checkbox

Two types of checkboxes must be distinguished. There are standalone checkboxes and checkboxes combined with a data input field.

1. To select or clear a combined checkbox:
   a) Press ENTER to activate the data input field.
   b) Press CTRL + ENTER to select or clear the checkbox.
2. To select or clear a standalone checkbox, press ENTER.

4.4.6 Using the Soft-Front Panel and Keyboard Shortcuts

The test software supports instruments with and without built-in display. The front panel of an instrument with display contains several important control elements.

If you display the GUI on a monitor, not on a built-in instrument display, these control elements are emulated by a soft-front panel. The panel contains buttons corresponding to the most important front panel keys.

As an alternative to the soft-front panel, you can also use keyboard shortcuts.

The following table lists the most important front panel hardkeys, the corresponding soft-front panel keys and keyboard shortcuts.

There are two variants of the soft-front panel, one for the multiple-window mode and one for the single-window mode.

Table 4-1: Mapping table for hardkeys, buttons and keyboard shortcuts

Hardkey	Keyboard	Soft-front panel Single window	Soft-front panel Multiple windows
[RESET]	CTRL + R	RESET	
[INFO]	CTRL + I	INFO	
[SAVE | RCL]	CTRL + S	SAVE	
[SETUP]	CTRL + E	SETUP	
[PRINT]	CTRL + P	PRINT	
[HELP]	F1	HELP	
[SYS]	Win + D, CTRL + ALT + D	SYSTEM	
[DEVICE]	CTRL + D	DEVICE	
[WIZARD]	CTRL + W	WIZARD	
[BLOCK VIEW]	CTRL + B	BLOCK VIEW	
[MEASURE]	CTRL + M	MEASURE	
[SIGNAL GEN]	CTRL + G	SIGNAL GEN	
[ON | OFF]	CTRL + ENTER	ON | OFF	Use mouse [1]
[RESTART | STOP]	CTRL + SHIFT + ENTER	RESTART | STOP	Use mouse [1]
[TASKS]	CTRL + TAB	TASKS	
[PREV]	SHIFT + TAB		
[NEXT]	TAB		
Hotkeys (left to right)	ALT + 1 to 8		
Softkeys (top down)	CTRL + 1 to 8		
[G/n]	CTRL + F12		
[M/µ]	CTRL + F11		
[k/m]	CTRL + F10		
[x1]	CTRL + F9		
[CLOSE]	CTRL + F4		

Hardkey	Keyboard	Soft-front panel Single window	Soft-front panel Multiple windows
[MIN]	CTRL + F5		
1) To switch an application on or off in multiple-window mode (for example a measurement), right-click the control softkey of the application, then click the desired action.			

4.5 Software Update

The following sections are related to software updates.

4.5.1 Software Packages

The R&S CMW100 software consists of the mandatory CMW base software package plus optional packages for firmware applications and utilities. These packages are integrated in setup files named `Setup_CMW1xx_<Scope>(Release)_<Version>.exe`.

For the base software package, `<Scope>` equals BASE, resulting in the file name `Setup_CMW1xx_BASE(Release)_<Version>.exe`.

Additional waveform (ARB) files are grouped into library packages and provided as self-extracting files or zip-files. They are only relevant if you want to use the functionality of the arbitrary RF generator. Unpack the desired libraries to the system drive. The preferred file location is the folder `waveform` in the CMW user data directory (for example `C:\ProgramData\Rohde-Schwarz\CMW\Data\waveform`).

4.5.2 Compatibility of SW Versions and Parallel Installation

The version indicator consists of three numbers. The first two numbers indicate the so-called software branch, for example 3.5 or 3.7.

For the initial release of a software branch, all software packages have the same version. For update packages compatible to this branch, the third number of the version indicator is increased.

Example: An installation for branch 3.5 can comprise, for example, software packages with version V3.5.10 (initial release), V3.5.11 and V3.5.20 (update packages). Within one software branch, only one version of each software package can be installed, not several versions in parallel.

Several software branches can be installed in parallel, for example 3.5 and 3.7. All software branches are stored on the system drive. But only one of the installed soft-

ware branches is active at a time, that means used by the instrument. To change the active software branch, see Chapter 4.5.4, "R&S Version Selector", on page 47.

4.5.3 Update Procedure

You can download software updates from the CMW customer web.

To install an update of one or several software packages, perform the following steps.

1. Shut down a running test software.
2. If you want to update software packages within an installed branch, uninstall the old packages using the following steps.
   You can use these steps also to uninstall a complete software branch.
   a) Open the "R&S Version Selector", for example via the corresponding icon on the desktop.
   b) In the "R&S Version Selector", disable the option "with Restart".
   c) Select the packages that you want to uninstall and press "Uninstall".
   During the uninstall process, confirm possible popup dialogs. Wait until the process is complete.
   d) If you want to uninstall additional software, repeat the previous step.
   e) Close the "R&S Version Selector".
3. Install the software updates.
   For a local installation, the required steps are the same as for the initial installation, see Chapter 4.1.7.2, "Installation Procedure", on page 26.
   For a remote installation, proceed as described in the following.

Remote installation

The first software installation is always a local installation.

Subsequent software updates can also be installed remotely. That means, you start the installer at any computer and install the software on one or several other remote computers. The computers are typically connected via LAN.

Proceed as follows to install packages remotely:

1. Ensure the following for all computers that you want to update:
   - Connect the computer to a radio test head.
   - Supply the radio test head with power.
   - For variant K02, connect your smart card to the computer.
   - Ensure that the "Device Group" setting in the "Setup" dialog is the same on all computers (only upper case letters, "DEFAULT" is recommended).
     Path to setting: [SETUP] or  > "System" > "Software Update" > "Device Group"

2. Copy all relevant setup files to the same directory. Use a directory on the system drive or on a USB memory stick.
3. Start the installer:
   a) Right-click one setup file, for example `Setup_CMW1xx_BASE(Release)_<Version>.exe`.
   b) Select "Run as administrator".

   The "R&S Software Distributor" opens.

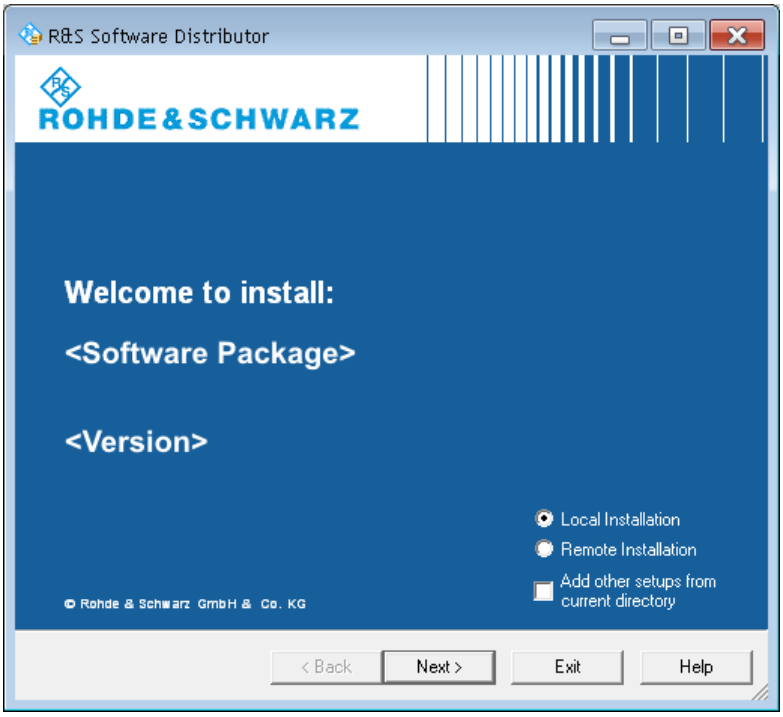


Figure 4-12: R&S software distributor

4. Select "Remote Installation".
   If you want to install additional packages provided by other setup files, enable "Add other setups from current directory".
5. Click "Next".
6. Select which packages you want to install.
7. Click "Next" to initiate a hardware scan.
   Wait until the "R&S Software Distributor" displays a list of computers connected to the LAN.
   If the hardware scan does not yield the desired results, check the "Options..." settings. The "Device Group" setting must match the corresponding setting in the "Setup" dialogs of the computers, see step 1. For further troubleshooting instructions, refer to the online help of the "R&S Software Distributor".
8. Select one or more computers where you want to install the software. Click "Install".

9. Wait until the installation is complete.

   The remote computers are rebooted automatically.

10. If you have updated the base software, perform the following action on each computer:

    a) Right-click the "CMW" icon on the desktop.

    b) Select "Properties > Compatibility" and enable "Run this program as an administrator".

Tests during type approval

It is not required to perform a recalibration of the R&S CMW100 after a software upgrade or downgrade.

The following tests are performed during the type approval of every new software release. The tests ensure that the R&S CMW100 TX and RX is working according to the specification after a software change.

Test 1:

- Adjustment (measurement of correction values) and verification with the latest official software.
- Upgrade to the new software and verification.

Test 2:

- Adjustment and verification with the new software.
- Downgrade to the latest official software and verification.

The tests are passed if all verification results show comparable values.

Rohde & Schwarz is certified according to ISO 9001 since May 1995.

4.5.4 R&S Version Selector

You can install several software branches in parallel. But only one of the installed software branches is active at a time and used by the instrument.

The "R&S Version Selector" allows you to change the active software branch, to uninstall optional software packages and to repair (reinstall) software packages.

Open the "R&S Version Selector" via the corresponding icon on the desktop.

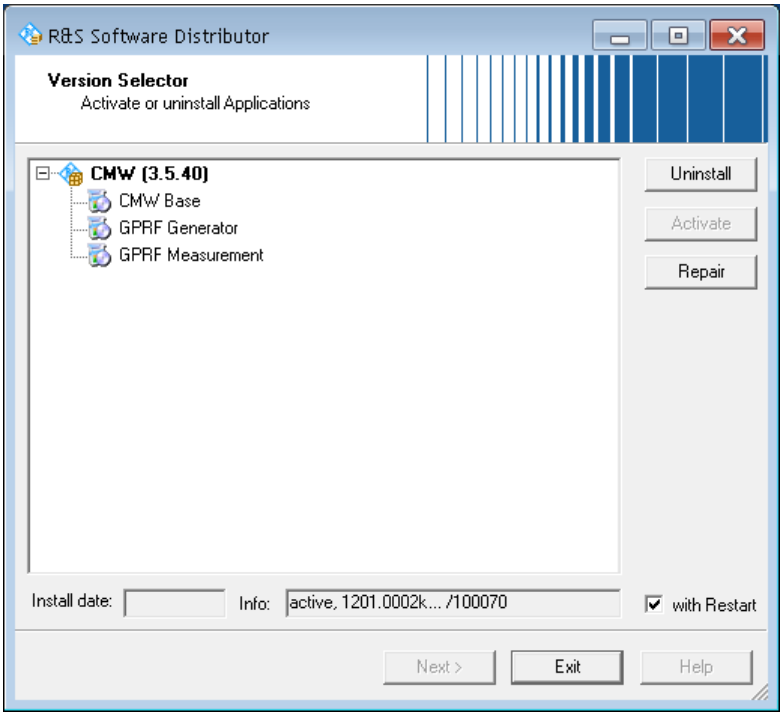


Figure 4-13: R&S version selector

To perform an uninstall, activate or repair action, follow these steps:

1. Select the relevant entry to the left. You can select a complete node (branch) or a single entry within a node.
   If you want to perform an automatic restart when the action is complete, enable "with Restart".
2. Click the relevant action button to the right.
3. Follow any instructions of the "R&S Version Selector".

4.6 Cleaning

NOTICE

Instrument damage caused by cleaning agents

Cleaning agents contain substances such as solvents (thinners, acetone, etc.), acids, bases, or other substances. Solvents can damage the labeling or plastic parts, for example.

Never use cleaning agents to clean the outside of the instrument.

NOTICE

Risk of instrument damage due to dirty surface

Dust on the instrument surface impedes cooling and increases the instrument temperature. Overheating can damage the instrument.

Regularly check the cooling fins at the sides of the instrument. Clean them if necessary.

► Clean the outside of the instrument using a soft, dry, lint-free dust cloth.

5 System Overview

This chapter provides an overview of the capabilities of the R&S CMW100 and their use. It includes a description of the basic concepts that the tester uses to organize, process and display measurement data. These basic concepts are valid for all firmware applications.

5.1 Generators

The R&S CMW100 provides a "General Purpose RF" (GPRF) generator. The following topics describe the principles of generator control and RF path configuration. For a sample session, refer to Chapter 4.3.1, "Generating an RF Signal", on page 33.

5.1.1 Generator Control

A generator can be in the "ON" or "OFF" state. In the default configuration, the generator is switched off. No output signal is available. The generator state is shown in the generator control softkey.

To turn the generator on or off:

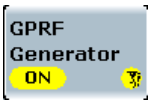
1. Select the generator control softkey and press [ON | OFF].
2. Alternatively, right-click the generator control softkey. Click "ON" or "OFF".

Wait until the control softkey indicates the "ON" state:

"Generator pending" state

Depending on the configuration, the R&S CMW100 requires some time to provide the generator signal. Example: The ARB generator signal is available only after a waveform file has been loaded.

While the generator is turned on but still waiting for resource allocation, adjustment or hardware switching, a yellow sandglass symbol in the generator control softkey indicates the "pending" state.



The yellow symbol disappears when the generator signal is available.

The "pending" state is also indicated while the generator is turned off but the resources have not yet been released.

5.1.2 RF Path Settings (Generators)

The following generator settings control the routing of signals and the generator level.

Signal routing settings (output)

The R&S CMW100 provides several RF connectors on the front panel. The RF output connector and the TX module to be used are selected in the "RF Routing" section at the beginning of the generator configuration dialog.

You can route the generated signal to up to eight RF output connectors in parallel.

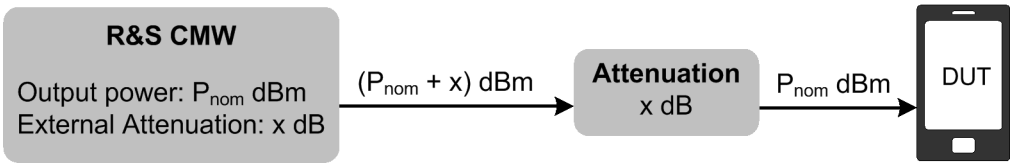
External attenuation (output)

Defines the value of an external attenuation (or gain, if the value is negative) in the output path. Use this setting to compensate the effect of a frequency-independent attenuating component or amplifier in the test setup, for example a cable, a test fixture or an RF shield box.

Additional settings for compensation of a frequency-dependent attenuation/gain are provided by the base system of the instrument. They allow you to define correction tables containing pairs of frequencies and associated attenuation/gain values.

If you activate a correction table for an output connector, the correction value derived from the table and the "External Attenuation" from the generator settings are added. Correction values for intermediate frequencies between two frequency entries are calculated using linear interpolation. For frequencies higher than the highest frequency entry in the table, or lower than the lowest frequency entry, the correction value associated with the highest / lowest frequency entry is used.

With a total external attenuation of x dB, the generator power is increased by x dB so that the generator power differs from the displayed output power. The displayed output power is available at the input of the DUT. Negative values of the external attenuation decrease the effective generator power.



Frequency-independent attenuations are defined as part of the generator settings, refer to the description of the generator application. Frequency-dependent correction tables are administrated in the "Setup" dialog or via remote commands (`...:FDCorrection:...`).

While a correction table is active for the connector currently used by an application, the GUI of the application displays the table name together with the frequency-independent attenuation setting.

You can display the table entries by clicking the button "FDCorr!".

5.1.3 Real-Time and Arbitrary (ARB) Generator Signals

The properties of real-time generator signals are based on the settings in the generator configuration dialogs. An example for a real-time signal (dual-tone GPRF signal) is described in Chapter 4.3.1, "Generating an RF Signal", on page 33.

In contrast, the arbitrary (ARB) generator signal is based on a "waveform file" (typically, a file generated with R&S WinIQSIM) which is loaded and processed at runtime. The R&S CMW100 supports waveform files for many network standards.

5.2 Measurements

The R&S CMW100 provides several measurement applications. Most measurement applications are dedicated to a specific network standard. Some are general-purpose applications, for example for power measurements.

All measurements are controlled in an analogous way. The following topics describe the principles of measurement control and measurement results that are similar in many measurement applications. For a sample session, refer to Chapter 4.3.2, "Measuring an RF Signal", on page 35.

5.2.1 Measurement Control

Measurements can be in the "RUN", "RDY", or "OFF" states. In the default configuration, all measurements are switched off; no results are available. The measurement state is shown on the measurement control softkey.

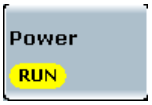
To turn the measurement on or off:

1. Select the measurement control softkey and press [ON | OFF] or [RESTART | STOP].
2. Alternatively, right-click the measurement control softkey. Click "ON", "OFF", "RESTART" or "STOP".

The behavior of the measurement control softkey depends on the "Repetition" mode selected in the configuration dialog:

- If the measurement is turned on in single-shot repetition mode, it enters the "RUN" state and returns to "RDY" when a single-shot result has been acquired.

- If the measurement is turned on in continuous repetition mode, it remains in the "RUN" state until it is turned off explicitly.



5.2.2 Connection Control (Measurements)

The R&S CMW100 provides several settings that are similar in different measurements but can be configured independently. These settings control the routing of input signals, the correction of the input power, the RF analyzer and the trigger system.

Signal routing settings (input)

The R&S CMW100 provides several RF connectors on the front panel. The RF input connector and the RX module to be used are selected in the "RF Routing" section at the beginning of the measurement configuration dialog.

External attenuation (input)

Defines the value of an external attenuation (or gain, if the value is negative) in the input path. This setting is suitable if the test setup contains a frequency-independent

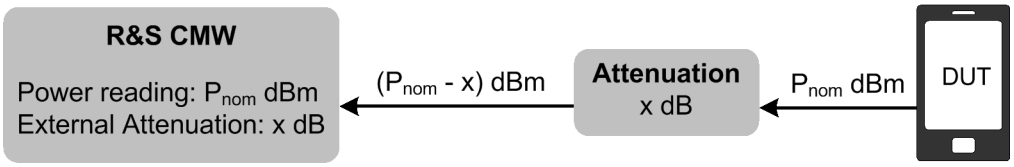
attenuating component or amplifier, for example a cable, a test fixture or an RF shield box.

Additional settings for compensation of a frequency-dependent attenuation/gain are provided by the base system of the instrument. They allow you to define correction tables containing pairs of frequencies and associated attenuation/gain values.

If you activate a correction table for an input connector, the correction value derived from the table and the "External Attenuation" from the measurement settings are added. Correction values for intermediate frequencies between two frequency entries are calculated using linear interpolation. For frequencies higher than the highest frequency entry in the table, or lower than the lowest frequency entry, the correction value associated with the highest / lowest frequency entry is used.

The total correction value (frequency-independent "External Attenuation" + frequency-dependent correction) modifies the power reading of the measurement and ensures that the measured powers are referenced to the output of the DUT.

- Positive values increase the power reading, compensating for an attenuation.
- Negative values reduce the power reading, compensating for an amplification factor (gain).



The external attenuation also enters into the internal calculation of the maximum input power that the R&S CMW100 can measure (see "Expected Nominal Power" below).

Frequency-independent attenuations are defined as part of the measurement settings, refer to the description of the measurement application. Frequency-dependent correction tables are administrated in the "Setup" dialog or via remote commands (`...:FDCorrection:...`).

While a correction table is active for the connector currently used by an application, the GUI of the application displays the table name together with the frequency-independent attenuation setting.

You can display the table entries by clicking the button "FDCorr!".

Expected nominal power

Defines the nominal power of the RF signal to be measured. Set the nominal power in accordance with the actual transmitter output power of the DUT. An additional "External Attenuation" (see above) can be used to compensate for the loss in the test setup. Some measurements provide additional parameters to account for variations of the signal power (e.g. the "User Margin" for the GPRF power measurement).

With an inappropriate setting of the expected nominal power, the measurement results generally deteriorate:

- If the "Expected Nominal Power" setting is too low, the RF input connector is overdriven and some measurement results are not reliable.
- If the "Expected Nominal Power" setting is too high, the RF input connector is underdriven, which also impairs the accuracy of the measurements.

Analyzer frequency

Sets the center frequency of the RF analyzer. This value must be in accordance with the measured RF signal to obtain meaningful measurement results.

5.2.3 Statistical Settings

Measurements generally cover a basic time interval and can be repeated periodically. The basic time interval is called measurement interval.

The number of measurement intervals that the R&S CMW100 repeats to calculate statistical results is called "statistic count". After one statistic count, the instrument has terminated a measurement cycle (single-shot measurement). Measurement cycles can be repeated for an unlimited number of times, resulting in the continuous repetition mode.

Most measurements provide sets of measurement results corresponding to the current measurement interval. And they provide statistical results like maximum, minimum and average over several consecutive measurement intervals. See also Chapter 5.2.4, "Statistical Results", on page 56.

The statistical settings described below are set in the configuration dialog of each measurement.

Statistic count / measurement cycle

The statistic count is the integer number of measurement intervals per measurement cycle. The duration of a measurement interval is measurement-specific. Conformance measurement specifications often request a certain number of repetitions of a particular measurement. Configure the statistic count accordingly. The required statistical depth for the measurement is then reached after one measurement cycle.

Measurement interval examples:

- The measurement interval for the GPRF power measurement is a configurable time interval termed the "Measurement Length"/"Step Length". The interval corresponds to either a single power step (if no sequence mode is active) or a step sequence.
- The measurement interval in most network test applications is related to a network-specific periodic time interval, for example a timeslot, burst or frame.

Depending on the repetition mode, a measurement extends over one or several measurement cycles.

Statistic count in continuous measurements

The statistic count is also relevant for continuous measurements, because it affects the averaging procedures, see Chapter 5.2.4.4, "Averaging", on page 59.

Repetition mode

The repetition mode defines how many measurement cycles are processed. Two modes are available:

- "Single-Shot": The measurement is stopped after one measurement cycle.
- "Continuous": The measurement is continued until you terminate it. The results are periodically updated.

Manual and remote control

The default repetition mode in manual control is "Continuous" (observe results over an extended period of time). The default mode in remote control is "Single-Shot" (measure one cycle and retrieve results).

Stop condition

For measurements providing a limit check, two stop conditions can be selected:

- "None": The measurement is performed according to its "Repetition" mode and "Statistic Count", irrespective of the limit check results.
- "On Limit Failure": The measurement is stopped when one of the limits is exceeded, irrespective of the repetition mode set. If no limit failure occurs, it is performed according to its "Repetition" mode and "Statistic Count". Use this setting for measurements that are intended for checking limits, e.g. production tests.

5.2.4 Statistical Results

The R&S CMW100 repeats the measurements according to the selected statistic count and repetition mode.

Consecutive measurement values are stored and used to calculate statistical results. The following sections describe the calculation of statistical results in detail.

5.2.4.1 Statistics Type

The statistics type defines how the R&S CMW100 calculates the displayed values if the measurement extends over several measurement intervals. Assume that a trace or a bar graph contains a series of different measurement points. After n consecutive measurement intervals, the instrument has collected n complete traces, corresponding to n measurement results at each point.

The different types of statistical results are calculated as follows:

- "Current": the current trace, i.e. the last result at all measurement points
- "Minimum": the smallest of the n collected values at each measurement point

- "Maximum": the largest of the n collected values at each measurement point
- "Average": a suitably defined average over all collected values at each measurement point
- "Standard Deviation": the root mean square deviation of all collected values at each measurement point from the "Average" value

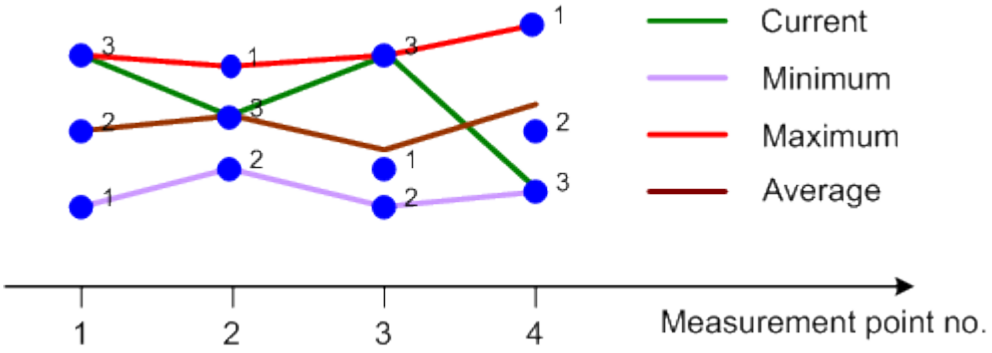


Figure 5-1: Types of statistical results

Differences between statistical calculations

"Minimum"/"Maximum" and "Average" results are calculated differently if the measurement extends over more than one measurement cycle (repetition mode "Continuous"):

- The "Minimum" and "Maximum" values represent the smallest and largest values since the start of the measurement.
- The "Average" result refers to the last measurement cycle.

The statistics type of the displayed trace generally belongs to the display configuration settings in the measurement configuration dialogs. For single measurement results, the R&S CMW100 often displays a table with all statistics types.

The statistics type is often combined with detector settings.

5.2.4.2 Detectors

The detector setting specifies how a single measurement result is calculated from a set of adjacent measurement points:

- "RMS": The displayed result represents the RMS average (e.g. the mean power) in a specified measurement interval. Over-estimation of stochastic signals (noise) is avoided.
- "Minimum": The displayed result represents the minimum value in a specified measurement interval. Narrow peaks cannot be smoothed out due to averaging.
- "Maximum": The displayed result represents the maximum value in a specified measurement interval.

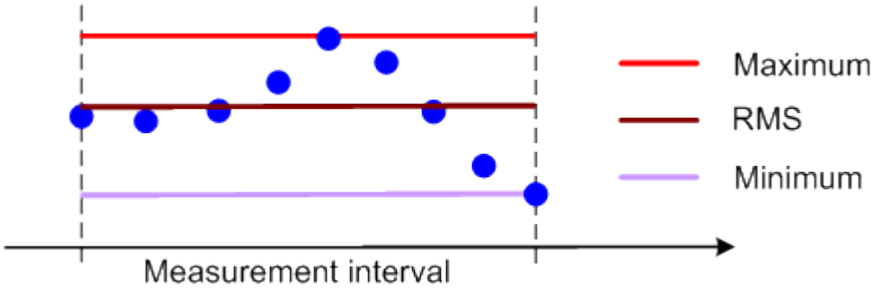


Figure 5-2: Detector type

The measurement interval varies from one measurement to another; it is typically a particular time or frequency interval.

Detector and statistics type settings can be combined. Examples:

- Statistics type "Current", detector "RMS":
  The current trace, calculated from RMS-averaged values over the specified "Measurement Length".
- **Statistics type "Current", detector "Minimum":**
  The current trace, calculated from minimum values within the "Measurement Length".
- **Statistics type "Current", detector "Maximum":**
  The current trace, calculated from maximum values within the "Measurement Length".
- **Statistics type "Average", detector "RMS":**
  The average trace, calculated from RMS-averaged values over the "Measurement Length".
- **Statistics type "Minimum", detector "Minimum":**
  The minimum trace, calculated from minimum values within the "Measurement Length".
- **Statistics type "Maximum", detector "Maximum":**
  The maximum trace, calculated from maximum values within the "Measurement Length".

5.2.4.3 Peak Values

"Peak" values are calculated as the maximum of the magnitude times the sign:

- For positive quantities such as the EVM, the peak value is equal to the maximum.
- For negative quantities such as the I/Q offset and the I/Q imbalance, expressed in dB, the peak value is equal to the minimum.
- For symmetric quantities with alternating sign such as the phase, frequency or timing error, the peak value is either the maximum or minimum, whichever has the larger magnitude.

5.2.4.4 Averaging

In single-shot measurements, "Average" values (traces and single values) are calculated as the arithmetic mean value over all measurement intervals since the start of the measurement. Assume that n measurement intervals have been measured. The average result at each measurement point is obtained recursively from the preceding (n - 1)st average result and the nth current result.

Equation 1:

$$Avg(n) = \frac{n-1}{n} \cdot Avg(n-1) + \frac{1}{n} \cdot Cur(n)$$

To obtain average traces, the R&S CMW100 calculates the average of consecutive measurement intervals at each trace point.

The formula above is modified for the magnitude error and the phase error, where positive and negative contributions tend to compensate each other. The "Average" value of these quantities is obtained as the average of the absolute values.

Equation 2:

$$Avg(n) = \frac{n-1}{n} \cdot Avg(n-1) + \frac{1}{n} \cdot |Cur(n)|$$

Logarithmic quantities are first averaged and then converted to a dB-value.

Note that the frequency error and timing error, although symmetric, is averaged according to equation 1.

For continuous measurements after the first cycle, equation 1 and equation 2 above are modified. For a statistic count c (c measurement intervals per cycle) and n > c, equation 1 is replaced by:

$$Avg(n) = \frac{c-1}{c} \cdot Avg(n-1) + \frac{1}{c} \cdot Cur(n) \quad (n > c)$$

As a consequence, the statistic count has an impact on average results obtained in continuous measurements.

5.2.4.5 Standard Deviation

The "Standard Deviation" σ_n indicates the spread of the n values at each measurement point. It is defined as the square root of the variance, which is the mean square of the deviation of the values from their own arithmetic mean.

$$\sigma_n^2 = \frac{\sum_{i=1}^{n} \left(x_i - \overline{x_n}\right)^2}{n}; \quad \sigma_n = \sqrt{\sigma_n^2}$$

The variance can be calculated using the following recursive equation:

$$\sigma_n^2 = \left[\left(x_n - \overline{x_{n-1}} \right)^2 \cdot \frac{1}{n} + \sigma_{n-1}^2 \right] \cdot \frac{n-1}{n}$$

With the arithmetic mean value:

$$\overline{x_n} = \frac{\sum_{i=1}^{n} x_i}{n} = x_n \cdot \frac{1}{n} + \overline{x_{n-1}} \cdot \frac{n-1}{n}$$

The formula above is modified for the magnitude error and the phase error, where positive and negative contributions tend to compensate each other. The arithmetic mean value and the standard deviation of these quantities is obtained from the absolute values.

5.2.5 Limit Check

Limits specify the allowed range for a particular set of measurement results. Typically, limits are used to check whether a DUT conforms to the rated specifications (conformance testing):

- An upper limit L_{upp} defines the maximum value for the measurement result R: $L_{upp} > R$.
- A lower limit L_{low} defines the minimum value for the measurement result R: $L_{low} < R$.
- A symmetric limit L_{sym} defines the maximum value for the absolute value of the measurement result R: R must be in the symmetric range $-L_{sym} < R < L_{sym}$.

A limit check consists of comparing the measurement results to the limits and displaying a pass/fail indication.

The R&S CMW100 provides different tools for viewing limits and limit check results.

- Limit lines show the upper, lower, or symmetric limits for a series of measurement results (measurement trace). In the measurement diagrams, limit lines are displayed in red color. A limit line consisting of different sections is termed a template.
- A pass/fail indication in a table of measurement results shows the limit check result for a single or a statistical result.

The following example shows the template for an 8PSK-modulated GSM burst: To pass the limit check, all "Power vs. Time" results must be between the upper and the lower limit lines.

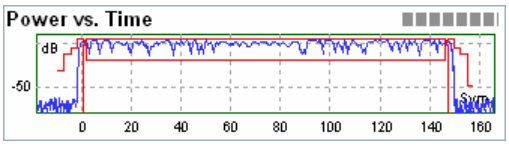


Figure 5-3: Example of limit lines (GSM)

The pass/fail indication in the result tables has the following meaning.

Table 5-1: Pass/fail indication in tables

Symbol	Meaning
(no indication)	Result passed
▲ 83.62	Result too large, exceeds upper limit
▼ 63.42	Result too small, below lower limit
▲ 60.35	Result too large but not reliable. Examples for reasons: • The analyzer is overdriven or underdriven. • The raw measurement data is possibly invalid.
▼ 29.30	Result too small but not reliable; see above
▲ ---	Result (probably) too large, but no valid result available
▼ ---	Result (probably) too small, but no valid result available

5.2.6 Measurement Triggers

The trigger system synchronizes a particular measurement with events.

The following sources of trigger events are used in many different measurements:

- The meaning of a "Free Run" trigger is measurement-specific. Usually, a "Free Run" measurement is not related to any trigger events. The R&S CMW100 measures as fast as possible. However, there are exceptions where "Free Run" trigger implies a synchronization to the RF input signal.
- With an "IF Power" trigger, the measurement is started when the level of the measured signal crosses a definite "Trigger Threshold" value. This trigger setting requires an RF input signal with variable power (power ramp, bursts). The trigger event can be set to occur at the rising or falling edge of the bursts.
- The ARB generator can provide marker signals, to be used for synchronization of a measurement.
- An "External" trigger is used to synchronize a measurement to external events. Example: A DUT providing a frame-periodic RF signal generates an additional trigger signal to indicate its frame timing.
  Feed an external trigger signal to the rear panel of the instrument.

Many measurements provide additional, specific trigger settings to improve their flexibility and performance.

5.2.6.1 Marker Signals

A marker signal provides events at specific points in time that can be used to synchronize the measurement. A typical example is a marker signal that is included in the waveform files generated by R&S WinIQSIM2.

Marker signals can be selected as trigger sources for measurements. The number and type of available marker signals depends on the installed firmware applications. The

R&S CMW100 detects all available marker signals, depending on its hardware/software equipment, and adds them to the list of available trigger sources.

5.2.6.2 Trigger Settings

Trigger settings enhance the flexibility of the trigger system and can help to avoid accidental trigger events. The trigger settings depend on the selected trigger source. No trigger settings apply to most "Free Run" measurements.

The following trigger settings are used in many firmware applications:

- The "Trigger Slope" setting specifies the edge (rising or falling edge) of the trigger signal that is to provide the trigger event. This trigger parameter is applicable to power trigger sources.
- The "Trigger Threshold" defines the power of the trigger signal where the R&S CMW100 generates a trigger event. Trigger signals below the trigger threshold are ignored by the trigger system. This trigger parameter is used for power trigger sources.
- A "Trigger Delay" delays the start of the measurement relative to the trigger event. Typically, a trigger delay is used to compensate for known propagation delays in the test setup or a known timing offset of the measured signal relative to the trigger signal.
- A "Trigger Timeout" is the maximum time after which an initiated measurement must have received a trigger event. If no trigger event is received, a trigger timeout is indicated in manual operation mode. In remote control mode, the measurement is automatically stopped.
- The "Minimum Trigger Gap" defines the minimum duration of the power-down periods (gaps) between two triggered power pulses. It can be used to prevent unwanted trigger events due to fast power variations.

5.2.7 TX Measurements

The purpose of a TX measurement is to assess the performance of an RF transmitter. Despite the differences in detail, TX measurements for different network standards have many properties in common.

5.2.7.1 Power Results

Power measurements are essential, e.g. for checking whether the transmitter output power complies with the power class of the device under test, or testing various power control mechanisms.

The R&S CMW100 provides two different types of power results:

- Most of the power results are averaged over an appropriate time/frequency interval (for example a burst). Average powers are used to check whether a transmitter produces the correct output power.

- Traces for the power versus time show a series of consecutive power steps or give detailed insight into the transmitted power. They show the structure of power ramps and possible effects of the modulation.

For details, refer to the description of the measurement firmware application.

5.2.7.2 Modulation Accuracy

Modulation accuracy is the ability of the UE transmitter to generate an ideally modulated signal. Modulation accuracy is assessed by several quantities which are analogous for all digital phase modulation schemes.

The modulation parameters are acquired in a single measurement process. The calculation is based on the comparison of the actual output signal Z of the transmitter under test with a reference signal R that is generated by the R&S CMW100. The reference signal represents an ideal error-free received signal.

An example for the process (WCDMA signals) is described in specification 3GPP TS 34.121, annex B.

Timing and Frequency Error

The R&S CMW100 must establish time synchronization with the input RF signal and estimate its timing, carrier frequency and power. The frequency error is the offset of the measured carrier frequency from the nominal RF frequency of the measured radio channel.

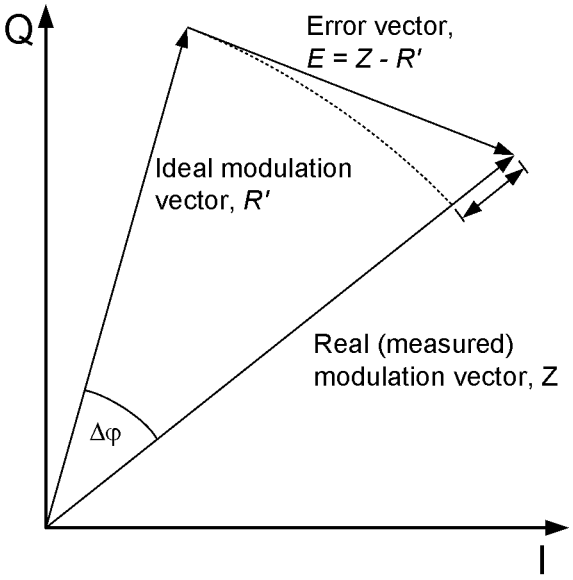
The calculated reference signal R is modified using the estimated timing, frequency error and power. The resulting corrected reference signal R' is used to determine the error vector magnitude, magnitude error and phase error.

The timing error is the deviation of the measured signal timing from the expected timing, which is derived from a trigger event.

Error Vector Magnitude, Phase Error, Magnitude Error, Code Domain Error

The error vector E = Z - R’ is calculated as an array at each sample in the measurement interval. From E and Z the following arrays can be calculated:

| E | = | Z - R’ |	**Magnitude of the error vector**, calculated at each sample in the measurement interval.
Δ φ	**Phase error**
| Z | - | R’ |	**Magnitude error**



In general, the measurement dialogs show the relative magnitude error and the relative EVM, i.e. the quantities defined above divided by the magnitude of the ideal modulation vector | R' |.

The **error vector magnitude** is calculated as the ratio of the RMS value of E to the RMS value of R' in percent or in dB:

*EVM in % = (RMS (E) / RMS (R')) * 100*

EVM in dB = 20 log (RMS (E) / RMS (R'))

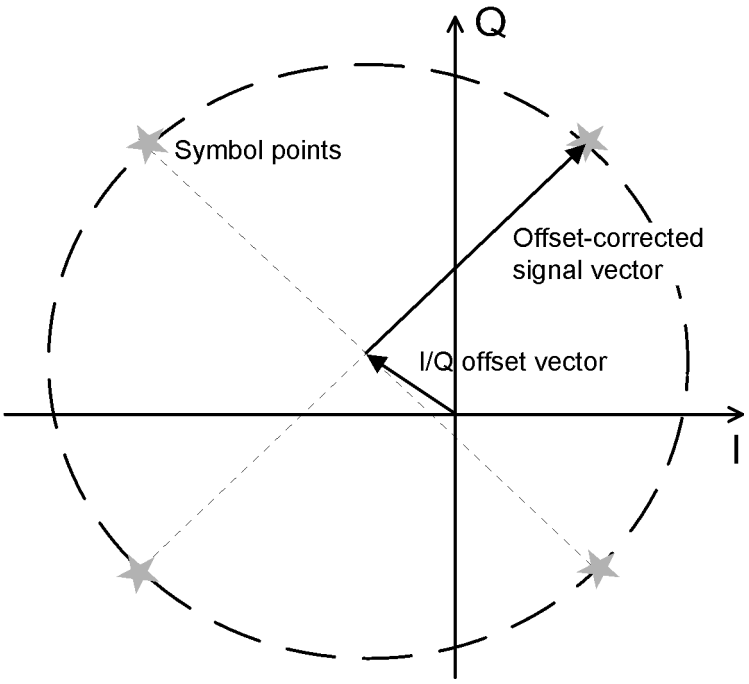
The **code domain error** (CDE) is calculated as follows: The error vector E = Z - R' is descrambled and projected onto all code channels of a specific spreading factor (SF). For each of the resulting projected error vectors E_k (k = 0 to SF - 1), the RMS value is calculated. The CDE is calculated as the ratio of this RMS value to the RMS value of R', i.e. PCDE = 20 * log (RMS (E_k) / RMS (R')) dB.

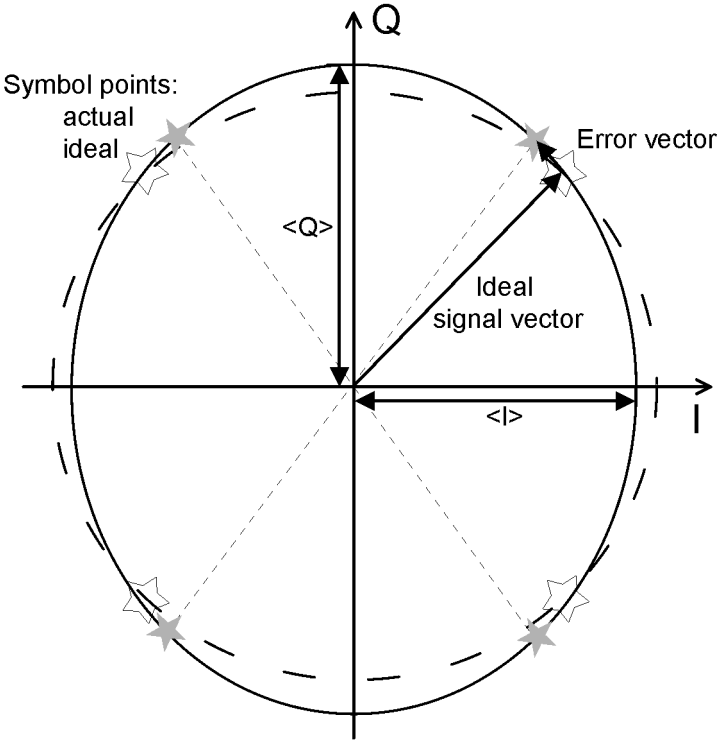
The **peak code domain error** (PCDE) is the maximum code domain error. It is calculated as:

*PCDE = 20 * log (max RMS (E_k) / RMS (R')) dB*.

I/Q Offset, I/Q Imbalance, Waveform Quality

The following figure is an idealized representation of the modulation errors. The effects of a pure origin offset (first diagram) and of a pure I/Q imbalance (second diagram) are disentangled.





The **I/Q offset** in dB is the logarithmic ratio of the I/Q offset vector (i.e. the estimated DC-offset of the measured signal) to the average offset-corrected signal vector:

$$\text{Origin Offset} = 20\log\frac{|\text{I/Q offset vector}|}{|\text{Offset-corrected signal vector}|_{\text{RMS}}}$$

In the equation above, | Offset - corrected signal vector $|_{RMS}$ denotes the magnitude of the offset-corrected signal vector that is RMS-averaged over all samples.

The **I/Q imbalance** in dB is equal to the difference between the estimated I and Q amplitudes of the measured signal, which are normalized and logarithmized as follows:

$$\text{I/Q Imbalance} = 20\log\frac{|<I> - <Q>|}{|<I> + <Q>|}$$

The **waveform quality** or rho factor is a measure for the modulation accuracy. It corresponds to the normalized correlated power between the actual waveform and the ideal waveform sampled at the constellation points. It is defined as:

$$\text{Waveform Quality} = \frac{|\sum_k R'_k Z_k^* |^2}{\sum_k |R'_k|^2 \sum_k |Z_k|^2}$$

Where R'_k is the k^{th} sample of the ideal signal, Z_k is the k^{th} sample of the measured signal (both in complex representation) and the sums run over all samples. For an ideal transmitter ($Z_k = R_k$ for all k), the waveform quality is equal to 1. For real transmitters, the waveform quality is a positive real number smaller than 1.

In some network applications (e.g. WCDMA), it is possible to select different algorithms for the modulation analysis:

- In the analysis "With Origin Offset", the modulation vectors R and Z for the EVM calculation are measured from the origin of the I/Q plane. So the results for the EVM, phase error and magnitude error include a possible origin offset.
- In the analysis "Without Origin Offset", the modulation vectors R and Z for the EVM calculation are measured from the coordinates of the I/Q offset vector. So the origin offset is subtracted out in the EVM, phase error and magnitude error results.

5.2.7.3 Adjacent Channel Power (Spectrum)

The R&S CMW100 measures the transmitter output spectrum emissions in a symmetric frequency range centered on the nominal RF carrier frequency. The spectrum emissions are a measure of the amount of energy that spills outside the designated radio channel. An excess amount of off-carrier power increases the interference with adjacent channels and decreases the system capacity.

The off-carrier power can be assessed by several complementary quantities:

- The adjacent channel leakage power ratio (**ACLR**) is the ratio of the power measured in an adjacent channel (adjacent channel power, **ACP**) to the transmitted carrier power, expressed in dB.
- In GSM networks, the "ACP modulation" is measured on a portion of the useful part of the burst, excluding the power ramps and the training sequence. The result is a measure for the part of the spectrum that is due to the modulation of the GSM signal. In contrast, the "ACP switching" result is the peak power within a minimum number of bursts. This result assesses the switching transients, i.e. the part of the spectrum that is due to the power ramp-up and down of the signal.

Sign conventions

According to the definition above, the sign of the ACLR values is often negative. This definition is in line with the ACLR definition for networks like GSM and TDMA but differs from the sign convention for WCDMA (3GPP/FDD); see specification 3GPP TS 34.121. To make results comparable, the R&S CMW100 uses the GSM sign convention for all network standards.

Spectrum Emission Mask

The spectrum emission mask is a template to limit the out-of-band emissions in a frequency range around the center carrier frequency. The spectrum emission mask complements the requirements for the adjacent channel power.

In the figure below, the red lines represent the spectrum emission mask for UL WCDMA signals (3GPP/FDD 3.84 MHz). The emission mask comprises different sections. In addition to the limit lines, the standard specifies IF filter settings for each section.

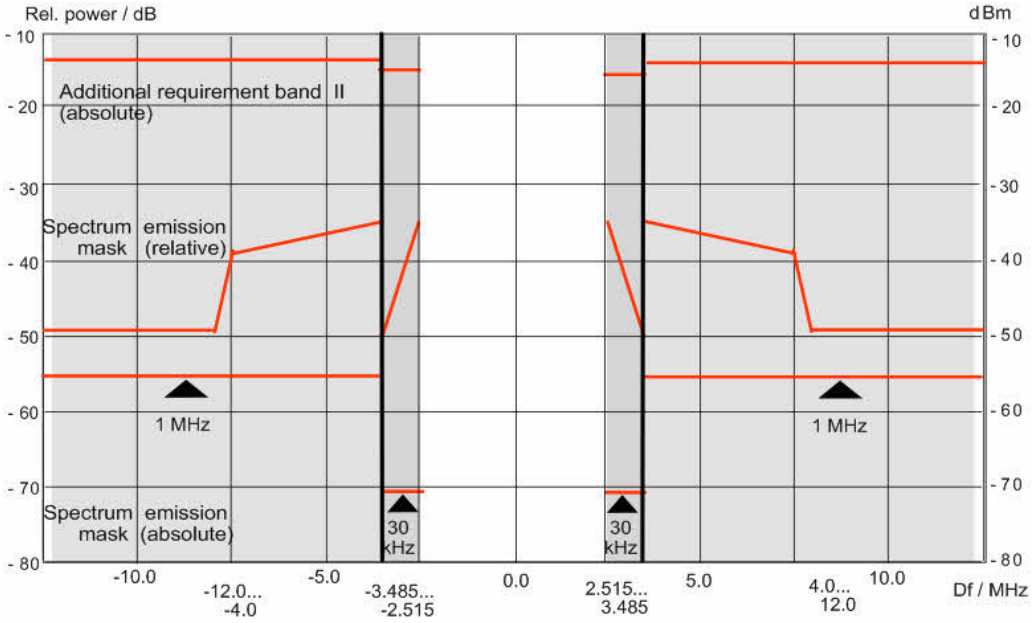


Figure 5-4: Spectrum emission mask (WCDMA/FDD 3.84 MHz)

Occupied Bandwidth (OBW)

For wideband and OFDM(A) signals, the occupied bandwidth is the width of a symmetric frequency interval around the nominal RF carrier frequency. It contains 99 % of the total integrated power of the transmitted spectrum. The occupied bandwidth shows whether the signal is confined to the assigned bandwidth of the channel. The following figure shows the occupied bandwidth for a WCDMA signal. The dark shaded area corresponds to the nominal bandwidth of 5 MHz.

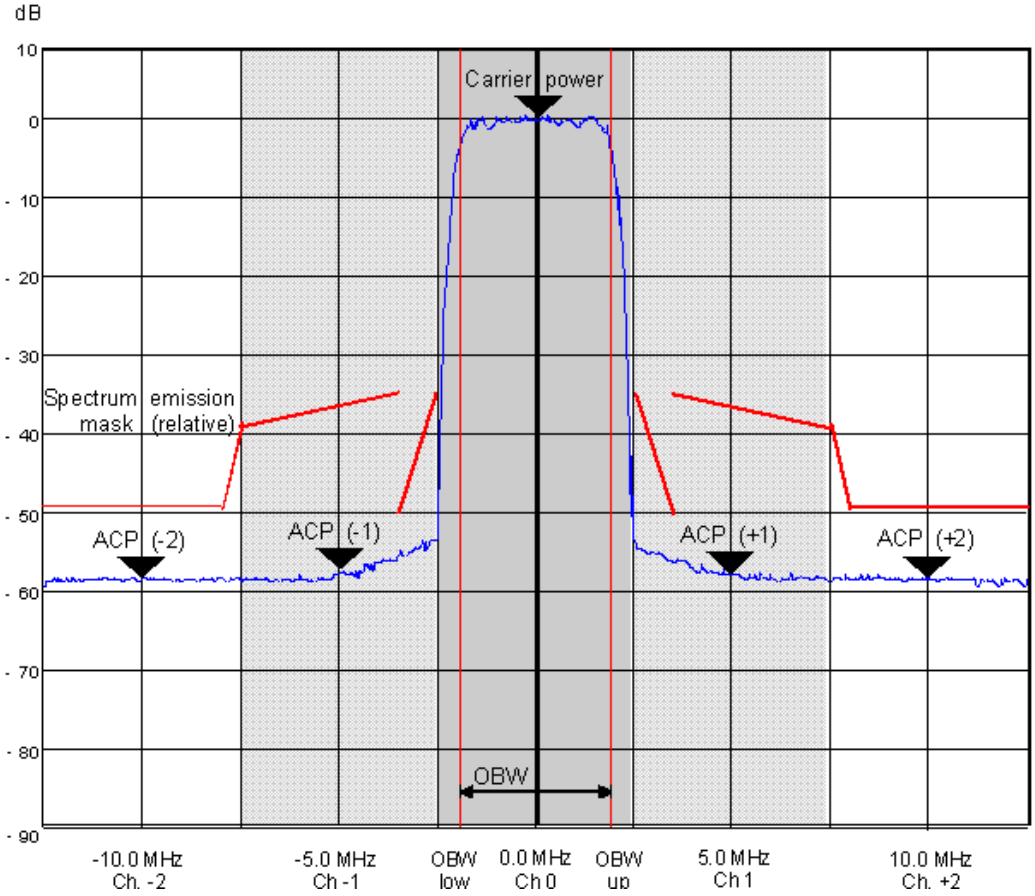


Figure 5-5: Occupied bandwidth (WCDMA/FDD 3.84 MHz)

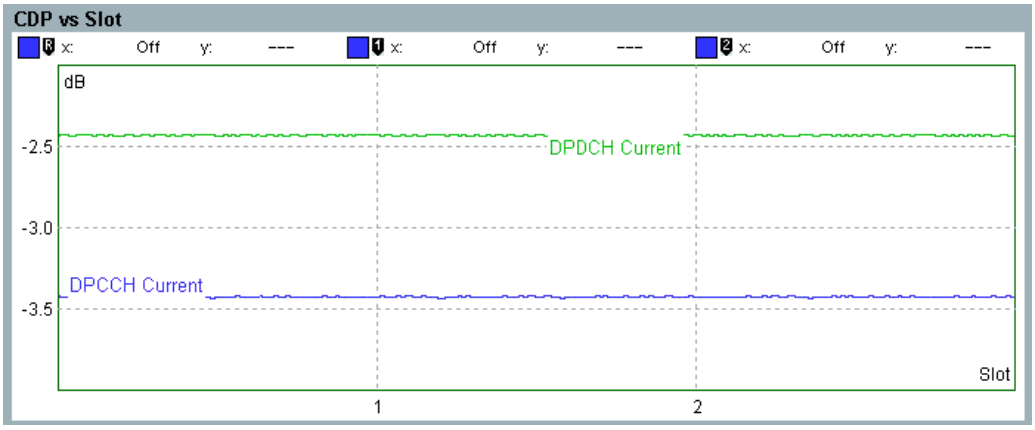
5.2.7.4 Code Domain Power

A code domain power (CDP) measurement provides the power of the individual code channels of a CDMA signal. The power in each code channel is averaged over a suitable time interval (e.g. a slot) and expressed in dB, relative to the power of the total, composite CDMA signal.

Typically, the following measurement tasks can be performed:

- Compare different physical channel powers within a CDMA signal
- Compare the observed channel powers with the signaled values (gain factors)
- Monitor active and inactive channels

In the following figure, the CDP of the DPCCH and the DPDCH in an uplink WCDMA signal is displayed over a measurement period of 120 WCDMA slots. The average DPDCH power is approximately 1 dB above the average DPCCH power.



5.2.7.5 I/Q Constellation Diagram

The constellation diagram shows the modulation symbols as points in the I/Q plane.

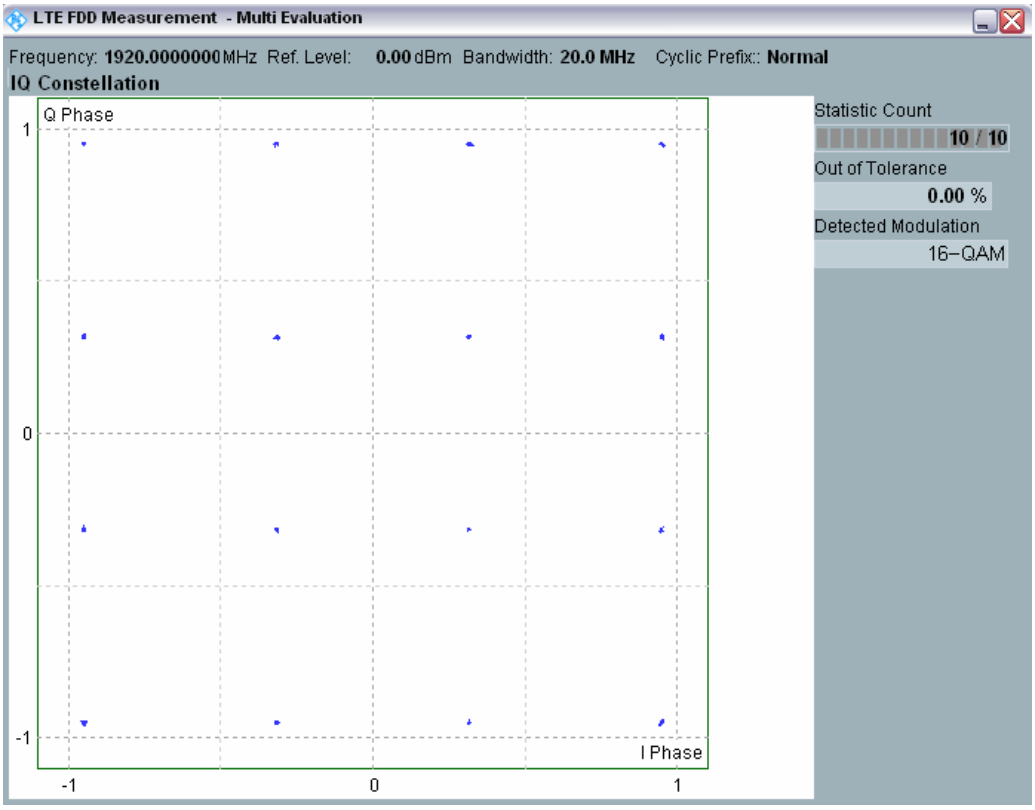


Figure 5-6: LTE multi-evaluation: I/Q constellation diagram

The constellation diagrams depend on the modulation type; for details refer to the description of the individual firmware applications. The diagrams are normalized such that the average distance of all points from the origin is 1.

Constellation diagrams give a graphical representation of the signal quality and can help to reveal typical modulation errors causing signal distortions, as shown in the

table below. In practice, the received signal shows a combination of the modulation errors listed.

Modulation error	Description / cause	Effect in the constellation diagram
I/Q imbalance	Caused by different gains of the I and Q components	One of the components is expanded, the other compressed
I/Q origin offset	Caused by an interfering signal at the RF carrier frequency	All constellation points are shifted by the same vector
Interferer	Non-coherent single-frequency spurious signal in the frequency band, superimposed on the modulated signal	Rotating pointer superimposed on each constellation point, causes circular constellation points
Gaussian noise	Uncorrelated interfering signals	Fuzzy constellation points
Phase error	Phase shift between I and Q components different from 90 deg	Non-orthogonal I and Q components
Phase noise	Uncorrelated phase error	Rotationally spreading constellation points
Amplitude compression	Large amplitudes below the nominal value, caused by non-linear components in the transmission path	Corner points move towards the center
Unused detected subcarriers (in OFDMA systems)	An unused/inactive subcarrier is measured, most likely due to a mismatch between the TX measurement settings and the measured signal	Unexpected constellation close to the origin (zero signal power)

5.2.7.6 Multi-Evaluation Measurements

In a multi-evaluation measurement, the R&S CMW100 acquires a wide range of measurement results at once.

The scope of the measurement depends on the network standard. Possible results are for example:

- Transmitter output power versus time
- Results that describe the modulation accuracy (EVM, phase error, frequency error, I/Q diagram, ...)
- Results that describe the output RF spectrum (ACP or ACLR, spectrum emission mask, spectrum flatness, ...)

Compared to independent TX measurements, multi-evaluation measurements provide several advantages:

- They ensure highest measurement speed.
- They provide a comprehensive picture of the performance of a tested RF transmitter with a minimum of effort for configuring the R&S CMW100.
- They provide "linked" results: The different measured quantities are all based on the same set of raw measurement data.

In remote control, it is possible to control each multi-evaluation measurement as a whole but retrieve the different types of results separately.

Controlling multi-evaluation measurements

A multi-evaluation measurement is controlled like any other measurement using the measurement control softkey.

Measurement results and settings

Multi-evaluation measurement results are displayed in a common measurement dialog which can comprise several diagrams (views) and other output elements. In general, it is possible to modify the display settings, e.g. to zoom in on a single diagram.

Measurement settings for the different views are also part of a common configuration dialog. Many parameters (e.g. the connection control settings and statistical parameters) affect the entire multi-evaluation measurement.

Measurement speed considerations

A multi-evaluation measurement ensures that the entire set of results is acquired and processed as quickly as possible. If only a part of the results is needed, it can be preferable to restrict the scope of the measurement to gain additional speed.

Example: Disabling the ACLR results in the GSM multi-evaluation measurement speeds up the measurement of the modulation accuracy.

5.2.8 RX Measurements

The purpose of an RX measurement is to assess the performance of an RF receiver. The R&S CMW100 transmits a definite bit pattern on the downlink (forward) RF signal. The device under test demodulates the received signals; the percentage of bits or data blocks received in error is counted.

There are different methods of assessing the receiver quality:

- In a single-ended receiver quality test, the received bit sequence is evaluated at the DUT. The measurement requires an ARB RF generator. No R&S CMW100 measurement application is needed.
- In a loop test, the DUT is commanded into an operating mode where it loops back the received data. The R&S CMW100 demodulates the uplink (reverse) signal and compares the received bits with the original pattern. The bit error rate is calculated, assuming no transmission errors in the uplink. The loop test requires an RF generator in combination with a measurement application.

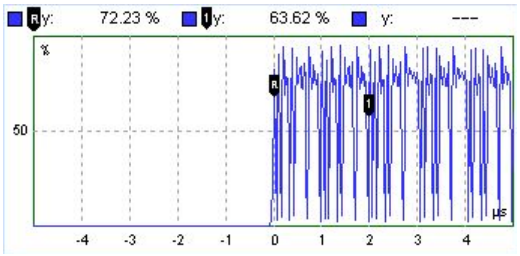
The bit error rate (BER) is defined as the percentage of bits that the DUT received in error:

*BER = <No. of Bits received in error> / <Total no. of bits> * 100 %*

Similar definitions apply to the packet error rate (PER), frame error rate (FER) etc., refer to the relevant applications.

5.2.9 Markers

Markers are tools for numerical readout of measured data in diagrams. A marker is displayed with a symbol on the trace. At the same time, the coordinates are displayed above the diagram.



In many views, the R&S CMW100 provides a reference marker (R) and additional markers labeled (1) and (2):

- The reference marker (R) indicates the coordinates of a trace point and defines the reference values for all relative markers.
- The markers (1) and (2) can be configured as absolute or relative markers. Absolute markers indicate the coordinates of a trace point. Relative markers indicate the coordinates relative to the position of the reference marker.

The "Trace Mode" defines whether all markers are always set to the same trace in the view ("Collective" mode) or positioned individually ("Individual" mode).

The marker settings are accessed via the "Marker" softkey. No remote control for markers is provided.

5.3 Smart Channel Mode

In the smart channel (SC) mode, a radio test head is split into several subinstruments. This feature requires option R&S CMW-K108. You need one option per radio test head.

Without the smart channel option, the smallest possible subinstrument is one complete radio test head with 8 bidirectional RF ports. Splitting the system into subinstruments is only possible if you have connected several radio test heads to the control computer.

With the smart channel option, you can split a single radio test head into 2, 4 or 8 subinstruments. The resulting subinstruments contain 4, 2 or 1 bidirectional RF ports.

The following table provides an overview of the possible splits.

	Number of subinstruments	
Controlled radio test heads	**Non-SC mode**	**SC mode**
1	1	2, 4, 8
2	1, 2	4, 8, 16
4	1, 2, 4	8, 16, 32

The purpose of the SC mode is to maximize the test capacity of a radio test head. Or in other words, to reduce the idle times to a minimum by running several test scripts on a single radio test head in parallel.

A radio test head can serve only one generator instance and one measurement instance at a time. A queuing mechanism serves parallel running scripts in turns.

Queuing mechanisms

The following applies to tasks that need the same resources, for example two measurement instances or two generator instances on the same radio test head.

A new measurement waits until an already running measurement has completed its measurement cycle. Then the running measurement enters the RDY state and the queued measurement starts.

A new generator instance waits until an already running generator instance enters the RDY state.

Generator state RDY

A GPRF generator instance enters the RDY state only if all the following preconditions are fulfilled:

- The SC mode is active (each subinstrument has less than eight RF ports).
- The generator is set to the ARB baseband mode.
- The ARB file is processed only once (repetition = single shot, retrigger = off).

In that case, the generator enters the RDY state when the ARB file processing is complete.

Trigger source names

The trigger source strings of the firmware applications contain a number indicating the firmware application instance.

In smart channel mode, the instance numbering is done per smart channel. For that reason, there is an additional prefix "Instr<m>" indicating the number of the smart channel.

Example: "Instr2: GPRF Gen1: User Defined Marker" is the user-defined marker of the GPRF generator instance 1 of smart channel number 2 of the radio test head.

Virtual RF connector names

In smart channel mode, a subinstrument comprises a subset of a single radio test head. So you need only the `RA...` virtual connector names.

To select a connector bench in the generator, you can always use `RA18`. You can enable up to eight connectors. The signal is output via all enabled connectors of the bench, even if they belong to different subinstruments.

To select a single connector, use `RA<n>`, identifying the <n>th connector of the subinstrument, as listed in the following table.

Table 5-2: Mapping of virtual connector names for the first radio test head

8 subinstruments			4 subinstruments			2 subinstruments		
Instr. no.	**Virtual**	**Physical**	**Instr. no.**	**Virtual**	**Physical**	**Instr. no.**	**Virtual**	**Physical**
1	RA1	1.1	1	RA1	1.1	1	RA1	1.1
2	RA1	1.2		RA2	1.2		RA2	1.2
3	RA1	1.3	2	RA1	1.3		RA3	1.3
4	RA1	1.4		RA2	1.4		RA4	1.4
5	RA1	1.5	3	RA1	1.5	2	RA1	1.5
6	RA1	1.6		RA2	1.6		RA2	1.6
7	RA1	1.7	4	RA1	1.7		RA3	1.7
8	RA1	1.8		RA2	1.8		RA4	1.8

For additional radio test heads, the mapping is similar. Continue counting the subinstruments and replace the first digit of the physical connector names.

5.4 Compensating Aging Effects

The level accuracy specifications of the R&S CMW100 as given by the data sheet are valid for a certain calibration interval, stated in the data sheet.

The accuracy when leaving the factory is significantly better than the data sheet values.

The following table lists typical values obtained during the manufacturing process based on the 95th percentile (2σ) statistical analysis.

RF generator output level		RF analyzer level	
Frequency range	**Typical uncertainty**	**Frequency range**	**Typical uncertainty**
400 MHz to 2700 MHz	0.4 dB	150 MHz to 3300 MHz	0.25 dB
2700 MHz to 4000 MHz	0.7 dB	3300 MHz to 4000 MHz	0.4 dB
4000 MHz to 6000 MHz	0.7 dB	4000 MHz to 6000 MHz	0.65 dB

During the time interval until the next regular calibration, the accuracy can drift towards the limits given in the data sheet. This behavior is due to the "aging" effect, which is inherent to all measurement instruments.

To get best accuracy throughout the whole calibration interval, the R&S CMW100 is equipped with the so-called internal path correction (IPC) function, see Chapter 6.5.6, "Internal Path Correction", on page 103.

The IPC function uses an internal high-precision level meter to measure the levels in the RX path and the TX path. It stores correction values to compensate the drift. Thus,

the initial accuracy can be maintained until the next regular calibration is due. However, please note that the IPC is not a replacement for the regular calibration.

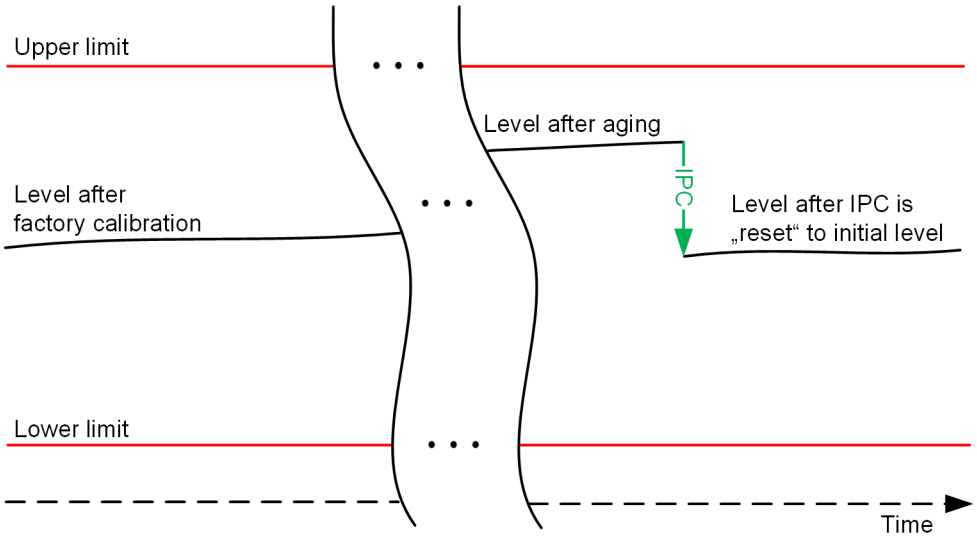


Figure 5-7: Level vs. time, effects of aging and IPC

6 Basic Instrument Functions

The following sections describe dialogs and settings for general use. The dialogs are not related to a particular general purpose or network test application.

6.1 Startup Dialog

The "Startup" dialog shows the progress of the startup procedure and the available options. Once the startup procedure has been terminated, it is automatically replaced by the dialog opened in the last session.

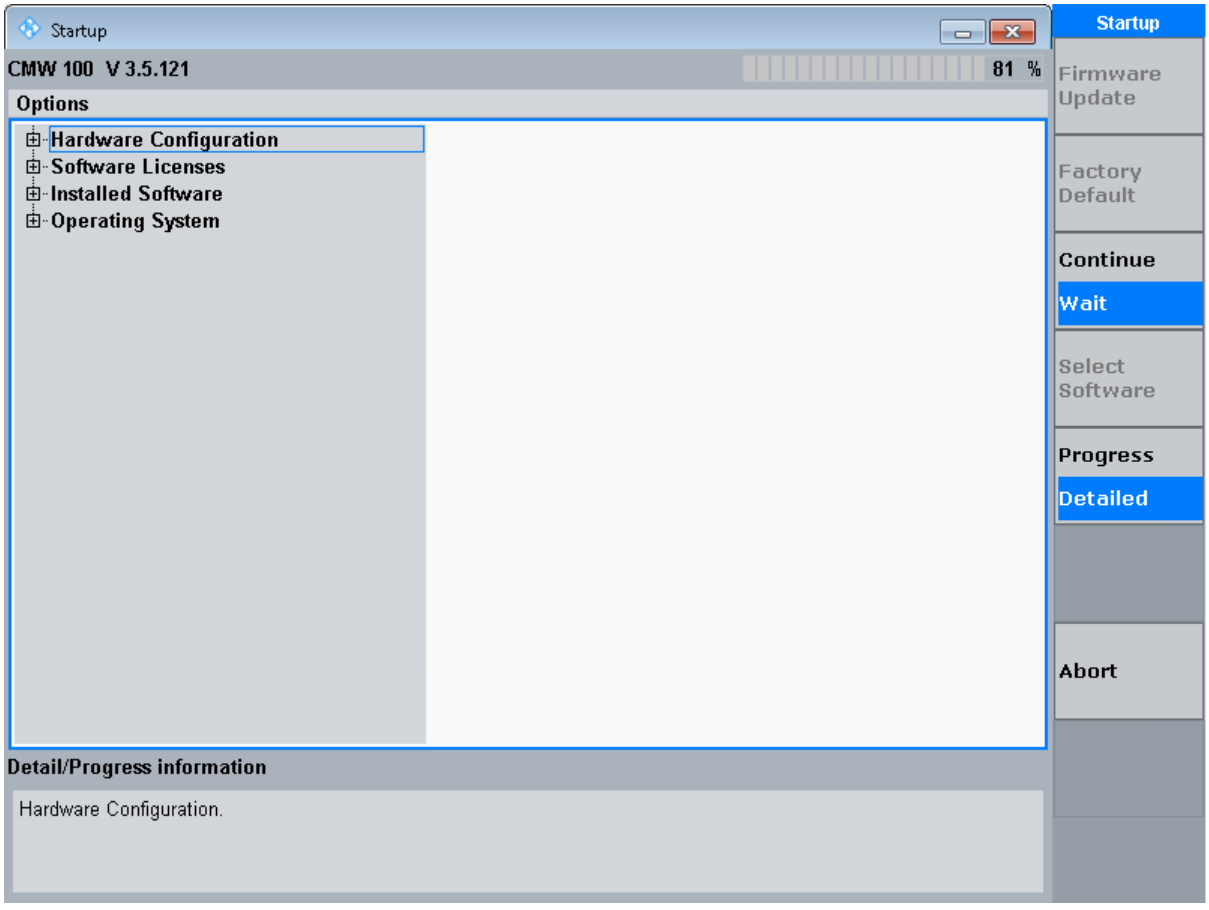


Figure 6-1: Startup dialog

The "Startup" dialog provides the following control softkeys that you can activate while the startup is in progress.

Firmware Update
Checks whether the data stored on the internal hardware of the R&S CMW100 is up-to-date. If necessary, copies new firmware-specific data to the internal hardware.

Factory Default
Sets the instrument to its factory default state.

This state comprises the following settings:
- All Preset settings. The instrument is optimized for local/manual operation.
- The subinstrument configuration: Only one subinstrument.

Continue / Wait
Resume or interrupt the startup procedure. "Wait" is appropriate e.g. for a check of the installed hardware and software options, see "Progress Info / Detailed Info" below.

Select Software
For future use

Progress / Detailed
Toggles between alternative information types in the "Option/progress information" panel across the bottom of the screen.
- The "Progress Info" is a log of the startup procedure including the loaded software components.
- The "Detailed Info" describes the selected hardware or software option. It contains the name, version, product code and essential technical data of the option.

It is preferable to press "Continue / Wait" to view the "Detailed Info". After the startup procedure is finished and the startup dialog is closed, the detailed option information can be accessed via the "Setup" dialog, see Chapter 6.5.4, "SW/HW Equipment", on page 98.

Abort
Aborts the startup procedure.

6.2 Main Window in Multiple-Window Mode

The multiple-window mode can only be used on external monitors, not on built-in instrument displays. To enable or disable the mode, see "Multiple Window" on page 90.

If the multiple-window mode is enabled, the following main window is displayed after the startup.

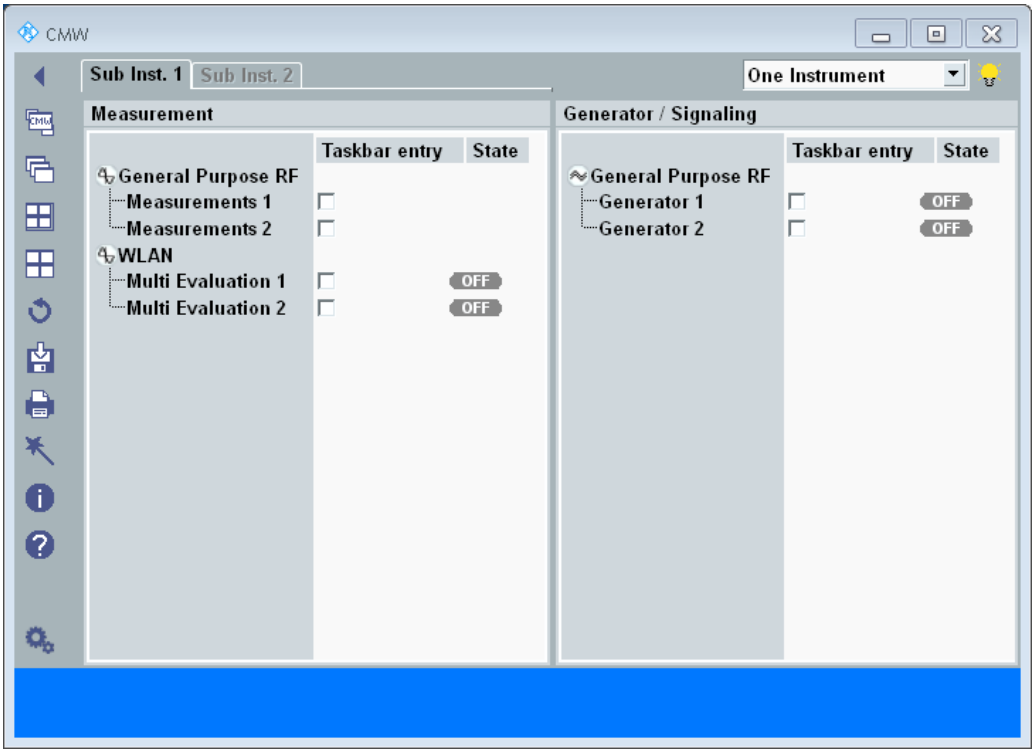


Figure 6-2: Main window in multiple-window mode

This main window serves as control center in multiple-window mode. From here, you can open additional windows, for example a measurement window, a generator window, a window showing the setup dialog or the help system.

You can keep all these windows open in parallel. You can, for example, modify generator settings in one window and at the same time monitor measurement results in a second window. Or you can keep the help window open while configuring settings.

Freely arrange the windows on your monitor and resize windows as desired. The position and the size of the windows is remembered and restored after a restart or after switching between subinstruments.

Subinstruments

There is at least one subinstrument tab. If your instrument can be split into subinstruments, there is one tab per possible subinstrument (depends on the instrument model and the installed options).

Each tab lists the firmware applications assigned to the subinstrument. Select a subinstrument tab to display all open windows related to that subinstrument.

To configure the split of the instrument into subinstruments, use the setting on the upper right ("One Instrument", "Two Sub Instruments", ...).

See also Chapter 6.8, "Instrument Setup Dialog", on page 119.

Firmware application windows

For manual control of firmware applications, you must enable the applications in the main window. For each enabled application, an entry is added to the task bar at the bottom and a new window with the application is opened.

Remote control of a firmware application is possible without enabling it in the main window.

The list of available firmware applications in the main window is dynamic. It depends on your installed software packets, the available licenses and the available hardware.

You can enable firmware applications also via the "Connector View" and at the same time configure the RF paths, see Chapter 6.2.2, "Connector View", on page 80.

6.2.1 Vertical Toolbar

The toolbar on the left of the main window allows you to rearrange the open windows comfortably and it serves as soft-front panel, see also Chapter 4.4.6, "Using the Soft-Front Panel and Keyboard Shortcuts", on page 42.

You can minimize the main window so that only the toolbar is visible.

Table 6-1: Toolbar buttons

Button	Purpose
	Buttons for minimizing the main window, restoring it and moving the toolbar, see Chapter 4.4.2, "Organizing Multiple Windows", on page 38
 to	Buttons for arranging multiple windows, see Chapter 4.4.2, "Organizing Multiple Windows", on page 38
	Opens the "Reset" dialog box, see Chapter 6.3, "Reset Dialog", on page 82
	Opens the "Save / Recall" dialog box, see Chapter 6.4, "Save/Recall Dialog", on page 84
	Opens the "Print" dialog box, see Chapter 6.6, "Print Dialog", on page 114
	Opens the wizard provided by the active firmware application (if available)
	Opens the "Info" dialog box, see Chapter 6.7, "Info Dialog", on page 115
	Opens the "Help" dialog box, see Chapter 1.4, "Using the Online Help System", on page 13
	Opens the "Setup" window, see Chapter 6.5, "Setup Dialog", on page 86

Button	Purpose
	Visible if the main window is minimized Quick-access to the center of the main window, where you can enable/disable applications
	Visible if the main window is minimized Quick-access to the subinstrument settings and selection of the displayed subinstrument

6.2.2 Connector View

You can configure the main window, so that it shows a connector view.

Path to the setting: "Setup" dialog > "System > Startup > Connector View as Main Menu". See also "Startup" on page 89.

The following figure shows the main window with enabled connector view.

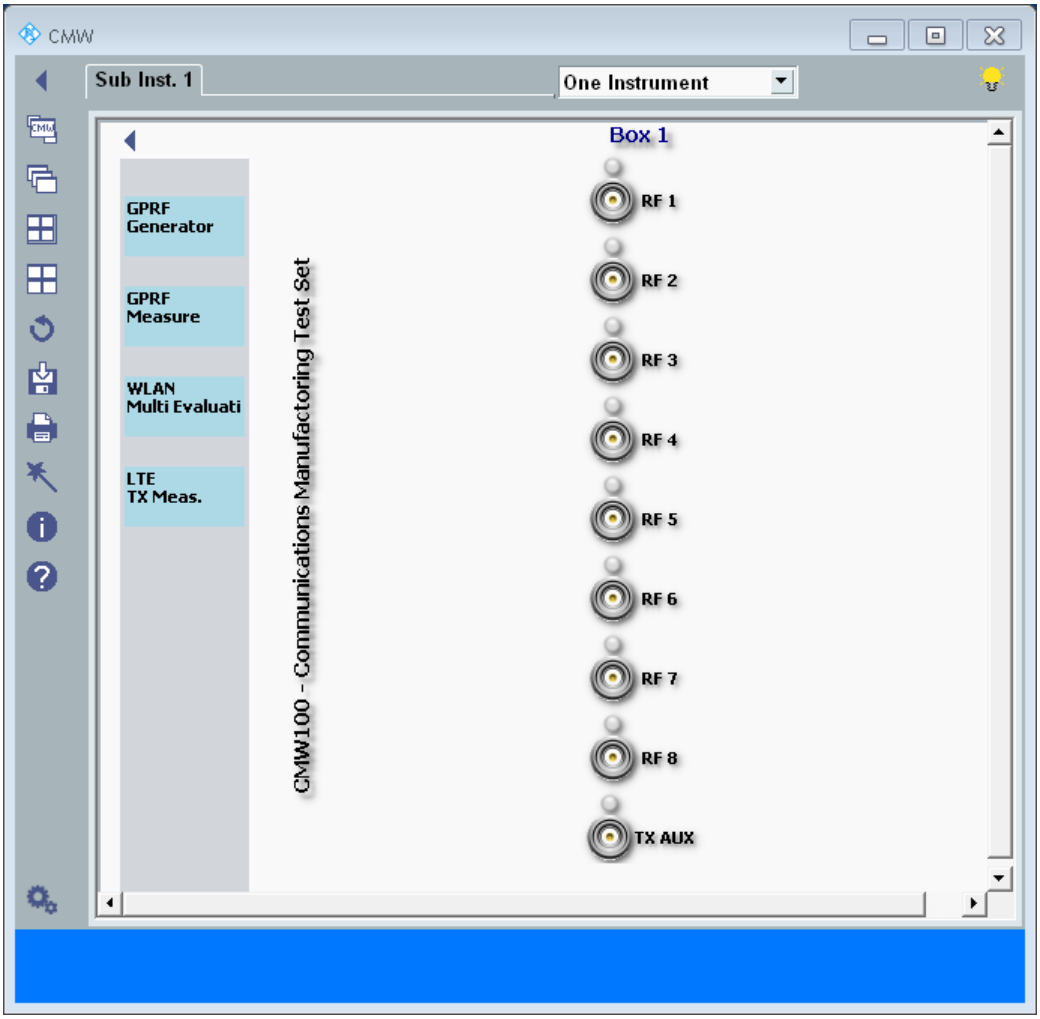


Figure 6-3: Connector view, initial situation

In this view, you can enable/disable applications and configure the RF connectors of active applications.

Each active measurement uses a single RF input connector. An active generator uses up to eight RF output connectors.

Most important actions:

- The disabled applications are listed on the left.
  To enable an application, double-click it. It is assigned to connector RF 1.
  Alternatively, drag an application to a specific connector.
- To assign an application to another connector, drag it to the new connector.
- To configure the active output connectors for a generator, enable/disable the checkboxes.
- To disable an application, close its window or drag the application to the list on the left.
- To bring the window of an application to the foreground, click the application.
- After you have enabled all relevant applications, you can hide the list on the left via the arrow button above the list.

The following figure shows a situation where a generator and two measurements have been enabled. The generator uses RF 1 to RF 4. The GPRF measurement uses RF 5. "GPRF Measure Power" means that the power measurement has been started last. Additional measurements like the "FFT Spectrum" measurement can also be running (started earlier). The WLAN measurement is enabled, but currently not running.

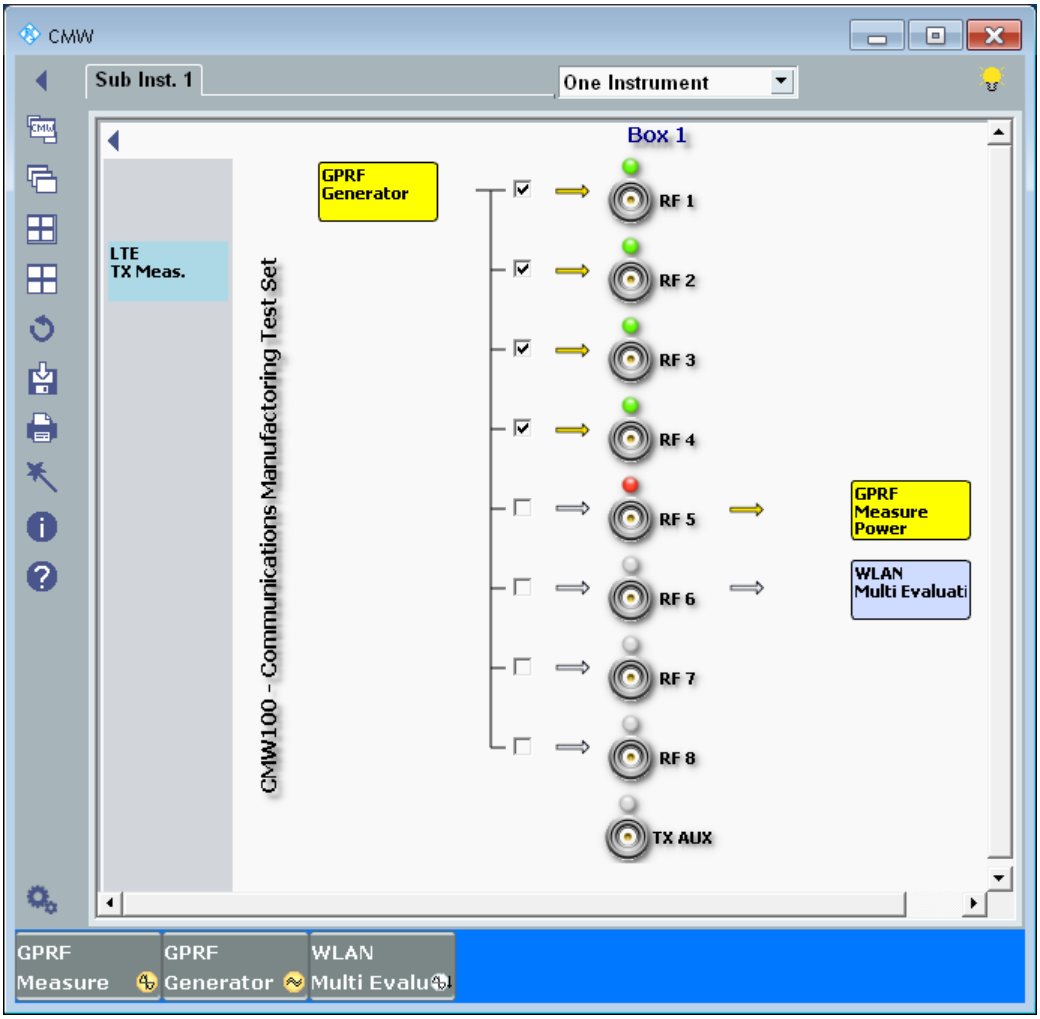


Figure 6-4: Connector view, with enabled applications

The colors of the connector LEDs are the same as on the physical front panel, see Chapter 4.2.1, "RF Ports", on page 31.

The colors of the applications indicate the states OFF/ON or OFF/RUN/RDY, see Chapter 5.1.1, "Generator Control", on page 50 and Chapter 5.2.1, "Measurement Control", on page 53.

If you have several subinstruments, there is a separate tab per subinstrument. If you have several radio test heads in one subinstrument, they are all displayed side-by side. Use the horizontal scrollbar.

6.3 Reset Dialog

The "Reset" dialog forces the software to return to a definite reset/preset state.

To open the dialog box, press the [RESET] key on the (soft-) front panel, or the button.

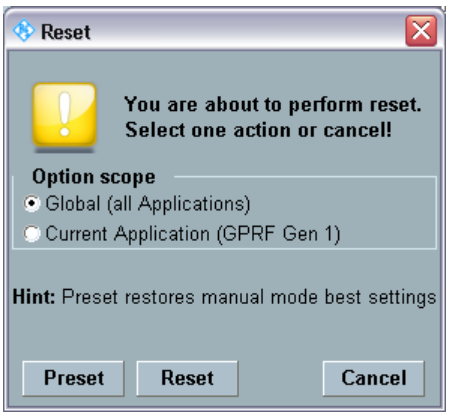


Figure 6-5: Dialog for one subinstrument

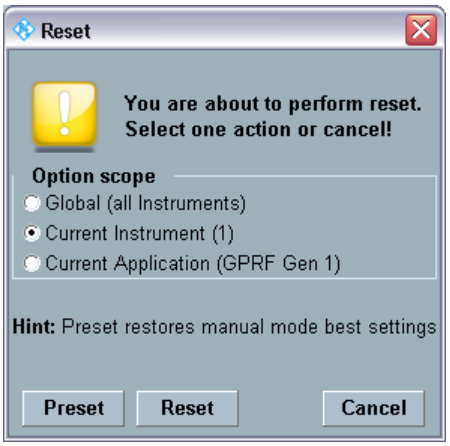


Figure 6-6: Dialog for several subinstruments

The reset/preset can be performed for the entire R&S CMW100 or it can be limited to the current firmware application.

If the R&S CMW100 has been split into subinstruments, the reset/preset can also be limited to the current subinstrument.

In addition to the reset and preset states described here, the instrument supports also a factory default state, see "Factory Default" on page 77.

Preset
Sets the instrument parameters to values suitable for local/manual interaction.

In particular, the preset state comprises the following settings:
- All measurements are performed in continuous repetition mode.
- The R&S CMW100 uses long statistics cycles (for reliable statistical evaluations).

The following R&S CMW100 settings are not affected by a preset:
- Address information assigned to the instrument (e.g. the IP address)
- The instrument setup

Remote command:
SYSTem:PRESet:ALL
SYSTem:PRESet:BASE
SYSTem:PRESet

Reset
Sets the instrument parameters to values suitable for remote operation.

In particular, the reset state comprises the following settings:
- All measurements are performed in single-shot repetition mode.
- The R&S CMW100 uses short statistics cycles (for benchmarks).
- In multi-evaluation measurements, some time-consuming evaluations are skipped to gain measurement speed.

The following R&S CMW100 settings are not affected by a reset:
- Address information assigned to the instrument (e.g. the IP address)
- The instrument setup
- The contents of the status registers

Remote command:
*RST
SYSTem:RESet:ALL
SYSTem:RESet:BASE
SYSTem:RESet

6.4 Save/Recall Dialog

The "Save/Recall" dialog stores the current instrument setup and recalls setups.

To open the dialog box, press the [SAVE | RCL] key on the (soft-) front panel, or the button.

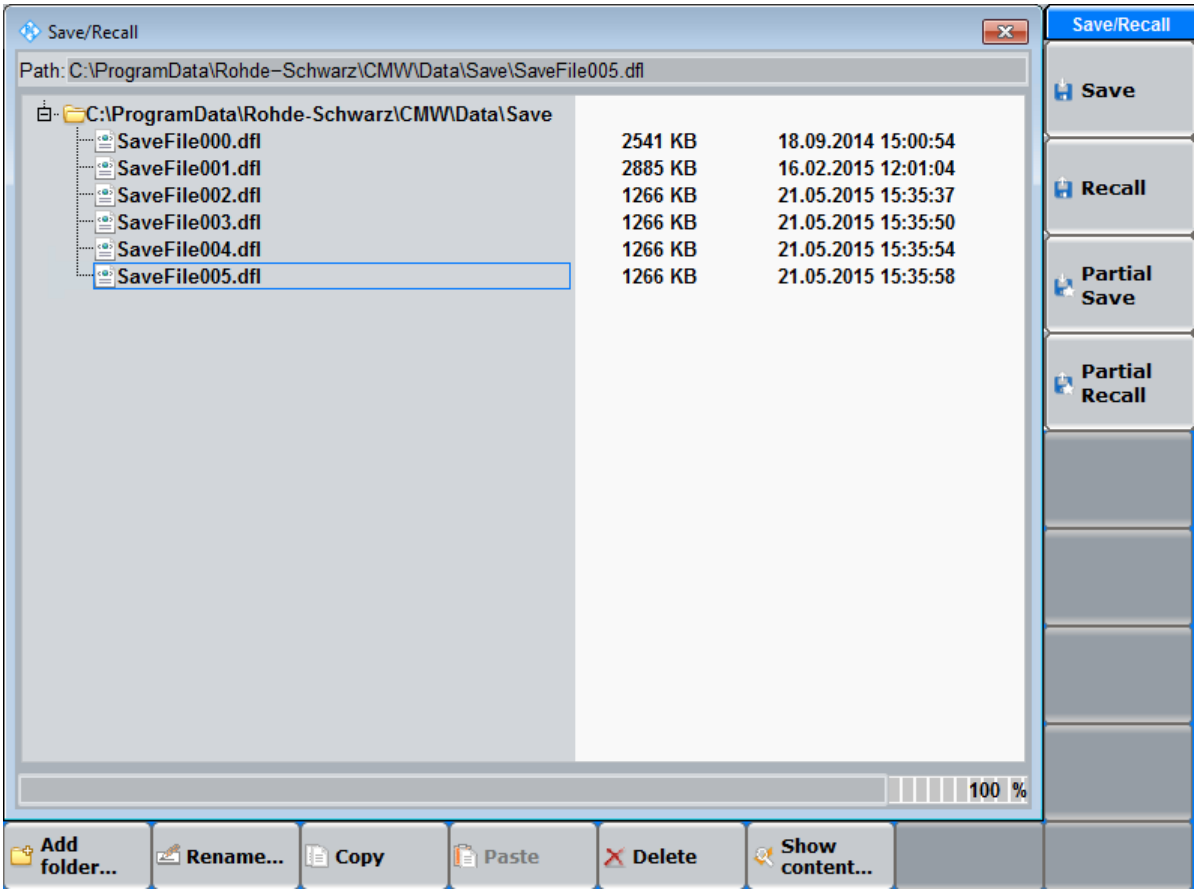


Figure 6-7: Save/Recall dialog

The "Save/Recall" dialog provides the following control softkeys and hotkeys.

Save / Partial Save
Saves the current configuration (or a part of it) to a file. Save files are stored in a default directory on the system drive.

- If a folder is selected, a new save file `SaveFile<no>.dfl` is created. `<no>` is 000 or the number of the last save file plus one.
- If a save file is selected, this file is overwritten.

"Partial Save" opens a dialog for selection of the information to be saved. For a description of selectable parts, see "Show content" on page 86.

Remote command:
`MMEMory:SAV`
`MMEMory:STORe:ITEM`

Recall / Partial Recall
Recalls the selected save file (or a part of it) and activates the stored settings.

"Partial Recall" opens a dialog for selection of the information to be recalled. For a description of selectable parts, see "Show content" on page 86.

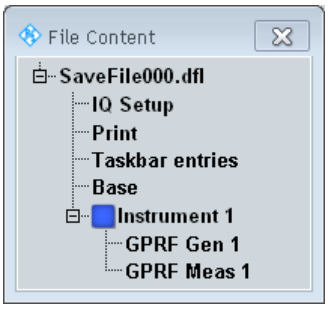
Remote command:
MMEMory:RCL
MMEMory:LOAD:ITEM

Add Folder
Adds a subfolder to the selected folder.

Rename / Delete
Renames / deletes the selected folder or file.

Copy / Paste
Copies / pastes a folder or file. An object can only be copied from one folder to another. The destination folder must not yet contain an object of the same type with the same name.

Show content
Opens a dialog showing which information is contained in the currently selected file.



Possible information categories are for example:
- "IQ Setup": "Digital IQ" settings in the "Setup" dialog
- "Print": "Print" dialog settings
- "Taskbar entries": information which firmware applications are active, i.e. are listed in the task bar
- "Base": most settings in the base system (except the settings listed above)
- "Instrument <n> – <Firmware Application>": all settings of the firmware application

6.5 Setup Dialog

The "Setup" dialog helps you to perform various basic and administrative tasks.

To open the dialog box, press the [SETUP] key on the (soft-) front panel, or the button.

6.5.1 System Settings

The "System" section of the "Setup" dialog configures the display, the network adapters, settings for remote software update and startup of the instrument.

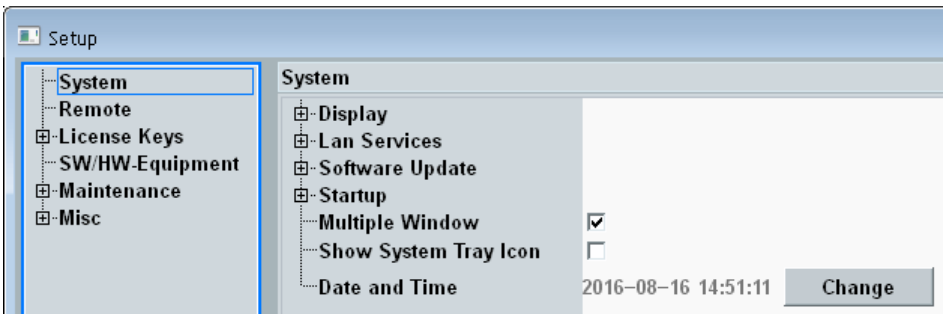
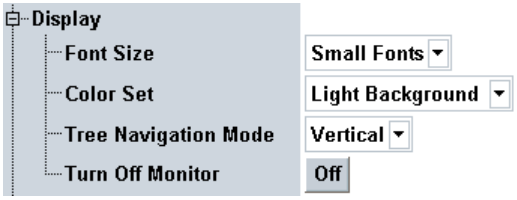


Figure 6-8: System section

Display
The "Display" section contains the following settings.



Font Size, Color Set ← Display
Configure the look of the GUI at the display.

Remote command:
`SYSTem:BASE:DISPlay:FONTset`

Tree Navigation Mode ← Display
Selects how trees are traversed when you turn the rotary knob on the (soft-) front panel.

- Zigzag: Traverse trees in a zigzag way, moving through all elements line by line.
- Vertical: Traverse trees vertically. You can use the [⇦] / [⇨] keys to traverse horizontally.
- Cursors: The [⇦] / [⇨] keys change to "Zigzag" mode, the [⇧] / [⇩] keys to "Vertical" mode.

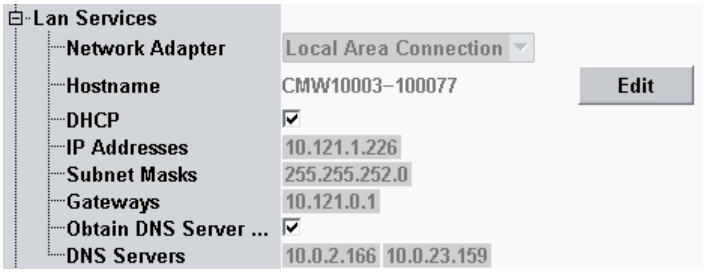
Remote command:
`SYSTem:DISPlay:ROLLkeymode`

Turn Off Monitor ← Display
Press "Off" to turn off the monitor or display showing the GUI.

You can reactivate the monitor/display via a connected keyboard or a connected mouse.

Remote command:
`SYSTem:DISPlay:MONitor`

Lan Services
The "Lan Services" section contains the following settings.



Each network adapter has an own set of settings.

Network Adapter ← Lan Services
Selects the network adapter to be displayed. All other parameters are related to the selected adapter.

Remote command:
`SYSTem:COMMunicate:NET:ADAPter`

Hostname ← Lan Services
Specifies the hostname of the R&S CMW100.

Remote command:
`SYSTem:COMMunicate:NET:HOSTname?`

DHCP ← Lan Services
Enables or disables DHCP. Before disabling DHCP, contact your network administrator.

Remote command:
`SYSTem:COMMunicate:NET:DHCP`

IP Addresses, Subnet Masks, Gateways ← Lan Services
With enabled DHCP, this information is assigned automatically and displayed for information.

With disabled DHCP, configure the settings compatible to your network. Contact your network administrator for appropriate settings.

Remote command:
`SYSTem:COMMunicate:NET:IPADdress`
`SYSTem:COMMunicate:NET:SUBNet:MASK`
`SYSTem:COMMunicate:NET:GATeway`

Obtain DNS Server Address Automatically ← Lan Services
Enables or disables the configuration of DNS addresses via DHCP.

Remote command:
`SYSTem:COMMunicate:NET:DNS:ENABle`

DNS Servers ← Lan Services
With enabled dynamic DNS, this information is assigned automatically and displayed for information.

With disabled dynamic DNS, enter the DNS server addresses manually.

Remote command:
`SYSTem:COMMunicate:NET:DNS`

Software Update

The "Device Group" must match the settings in the "R&S Software Distributor" to perform a remote software update. The entry is case-sensitive and the software distributor does not allow lower case letters.

Remote command:
`SYSTem:UPDate:DGRoup`

Startup

The settings in this section take effect during/after a reboot of the instrument.

- "Boot Factory Default":
  The instrument is booted with factory default settings. Enabling this setting has the same effect as pressing the "Factory Default" softkey each time the instrument is started, see Chapter 6.1, "Startup Dialog", on page 76.
- "Onscreen Keyboard":
  Activates the on-screen keypad / keyboard. If you access a numeric entry field, the on-screen keypad is displayed. If you access a text entry field, the on-screen keyboard is displayed.
  This feature is useful if you have a touchscreen connected. If you want to use an external keyboard, disable the setting.
  See also Chapter 4.4.4, "Entering Data via the On-Screen Keyboard", on page 40
- "Connector View as Main Menu":
  Configures the main window of the multiple-window mode. If the checkbox is off, the main window shows a tree of firmware applications. If the checkbox is on, the window shows a connector view.
  See also Chapter 6.2.2, "Connector View", on page 80.

Remote command:
`SYSTem:STARtup:PREPare:FDEFault`

Multiple Window

The graphical user interface provides a single-window mode and a multiple-window mode.

If the checkbox is disabled, the single-window mode is used.

If the checkbox is enabled and the screen resolution is sufficient, the multiple-window mode is used. The screen must have a horizontal resolution of at least 1600 pixels, or a vertical resolution of at least 1080 pixels.

See also Chapter 6.2, "Main Window in Multiple-Window Mode", on page 77.

Remote command:
`SYSTem:BASE:DISPlay:MWINdow`

Show System Tray Icon

Adds an icon to the system tray of the operating system. If the icon is shown, the CMW software starts up minimized. To access the CMW application after the startup, you must perform the "Open" action.

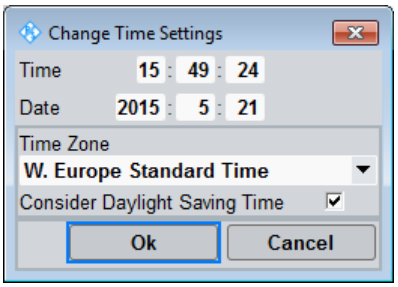
The context menu of the icon provides the following entries:

- "Close" or "Open":
  "Close" hides all windows and taskbar entries of the CMW application.
  "Open" restores the windows and taskbar entries.
  The context menu toggles between the two entries. You can also click the system tray icon to trigger these actions.
- "Shutdown":
  Shuts down the CMW application.

Remote command:
SYSTem:BASE:STICon:ENABle
SYSTem:BASE:STICon:OPEN
SYSTem:BASE:STICon:CLOSe

Date and Time
This section displays the date and time settings of the operating system.

Press "Change" to open a dialog box for configuration of the settings.



Time ← Date and Time
Sets the local time of the operating system clock.

Remote command:
SYSTem:TIME:LOCal

Date ← Date and Time
Sets the local date of the operating system calendar.

Remote command:
SYSTem:DATE:LOCal

Time Zone ← Date and Time
Sets the time zone in the date and time settings of the operating system.

Remote command:
SYSTem:TIME:DSTime:RULE:CATalog?
SYSTem:TIME:DSTime:RULE

Consider Daylight Saving Time ← Date and Time
Configures whether the operating system automatically adjusts its clock for daylight saving time (DST) or not.

The rules defining when exactly the clock must be adjusted by which offset, depend on the configured time zone.

Remote command:
SYSTem:TIME:DSTime:MODE

User Mode

The user mode can be changed using the softkey "User Mode Login".

- The "User" mode provides the complete measurement functionality of the R&S CMW100 and most selftests.
- The "User (Extended)" mode provides additional selftests.
- The remaining user modes are for service purposes.

Remote command:
n/a

6.5.2 Remote Settings

The "Remote" section of the "Setup" dialog lists all available interface and protocol types for remote control of the instrument and their parameters.

The highest level lists a subset of the following values:

- HI-SLIP (LAN connector, HiSLIP protocol)
- VXI-11 (LAN connector, VXI-11 protocol)
- TCPIP (LAN connector, VISA socket resource)

For remote control via LAN, it is recommended to use the HiSLIP protocol.

The entries at the second level correspond to one remote channel each. Each channel is assigned to one subinstrument.

See also Chapter 7.1.6, "Multiple Channels for Remote Access", on page 131

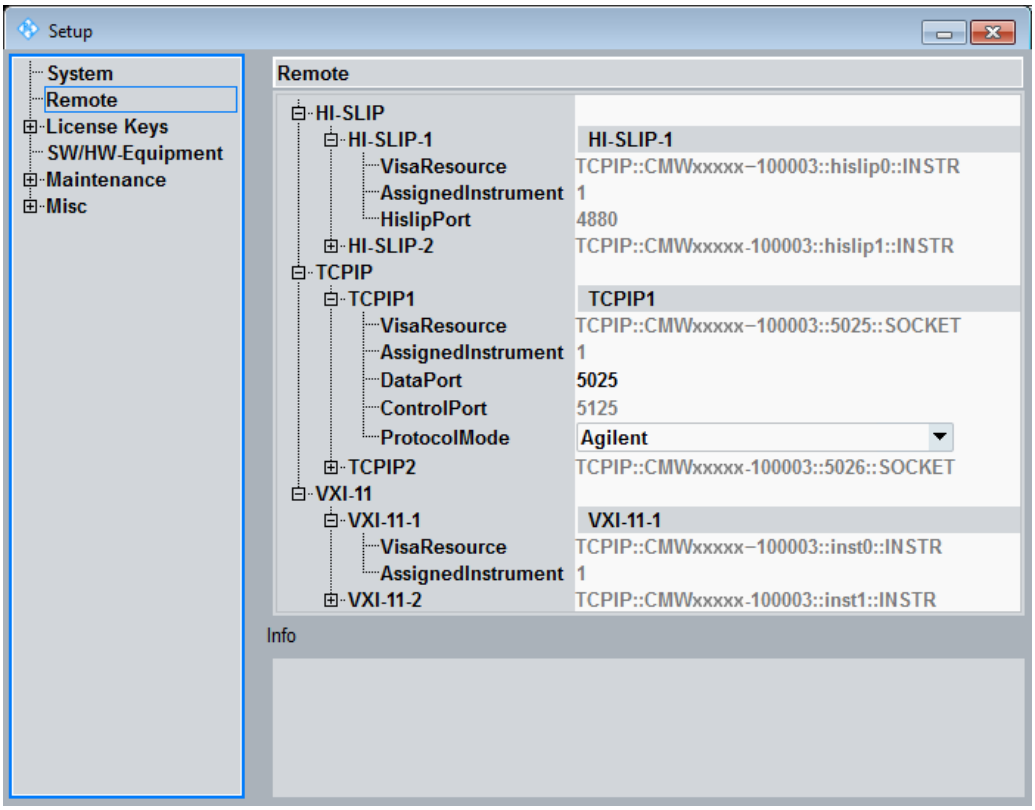


Figure 6-9: Remote section

Most connection parameters are fixed and displayed for information only. The most important parameters are described below.

Visa Resource (all protocol types)
The resource string depends on the protocol type and the assigned address information. Each remote channel is identified by a different VISA resource string.

The string is composed as follows:

- HI-SLIP: *"TCPIP[board]::<IP_address>[::<device_name>[,<IP_port>]][::INSTR]"*
  - *<device_name>*: equals *hislip<n-1>* for subinstrument n
  - *<IP_port>*: default port number is 4880
- VXI-11: *"TCPIP[board]::<IP_address>[::<device_name>][::INSTR]"*
  - *<device_name>*: equals *inst<n-1>* for subinstrument n
- TCPIP: *"TCPIP[board]::<IP_address>::<data_port>[::SOCKET]"*

Remote command:

```
SYSTem:COMMunicate:HISLip<i>:VRESource?
SYSTem:COMMunicate:VXI<i>:VRESource?
SYSTem:COMMunicate:SOCKet<i>:VRESource?
```

Assigned Instrument (all protocol types)
Identifies the subinstrument to be controlled. This parameter is fixed. If only one subinstrument is available, it is addressed as "Assigned Instrument 1".

For VXI-11 and HiSLIP, the "Assigned Instrument" number is also part of the VISA resource string, with instrument 1 mapped to 0, instrument 2 mapped to 1, and so on.

Remote command:
n/a

Data Port (TCPIP only)
The data port number is part of the TCPIP VISA resource string. Use different data ports for different channels (to address different subinstruments). The data port is used for all protocol modes.

Consider the following rules:

- To modify the port number, enter a 0. A free port in the range 1024 to 32767 is then assigned automatically.
- Alternatively, enter a port number in that range manually.
- Never configure a port number in the range 1 to 1023. These "well-known ports" are reserved for specific services.

Remote command:

```
SYSTem:COMMunicate:SOCKet<i>:PORT
```

Control Port (TCPIP only)
The control port is only relevant for protocol mode "Agilent". It can be used to set up an optional control connection for transfer of emulation codes. The control port number is displayed for information. It cannot be modified.

Remote command:
n/a

Protocol Mode (TCPIP only)
The protocol mode defines the support of control messages, e.g. polling or service request:

- **RAW**: no support for polling and service request (but best performance)
- **Agilent**: emulation codes via control connection (control port)
- **IEEE1174**: emulation codes via data connection (data port)

Remote command:
`SYSTem:COMMunicate:SOCKet<i>:MODE`

6.5.3 Option Activation and Deactivation

The "License Keys" section of the "Setup" dialog allows you to manage licenses. You can apply a delivered activation key or deactivation key. You can also perform steps required to move a portable license.

If you install a new software option, you must enable the option via an activation key, before you can use the option.

Unregistered licenses
Only registered licenses are accepted. If your license is delivered unregistered, use the online tool R&S License Manager to register the license. Note that the registration of a non-portable license is irreversible, so ensure that you register it with the correct device ID. The address of the tool is https://extranet.rohde-schwarz.com/service.

The R&S License Manager allows you also to move a portable license to another device ID.

For detailed step-by-step instructions, refer to the installation instructions delivered together with the license and to the documentation available in the R&S License Manager (in the online tool, press ?).

For an example, see Chapter 6.5.3.2, "Moving a Portable License", on page 96.

The "License Keys" section provides the following subsections:

- **"Active License Keys"**: lists non-portable active licenses. The list can be sorted alphabetically or by category (toggle parameter "Show Category View").
- **"Portable License Keys"**: lists portable active licenses
- **"Deactivation Keys"**: lists applied deactivation keys
- **"Inactive License Keys"**: lists licenses that have been explicitly deactivated by a deactivation key or are for any other reason inactive, for example expired licenses

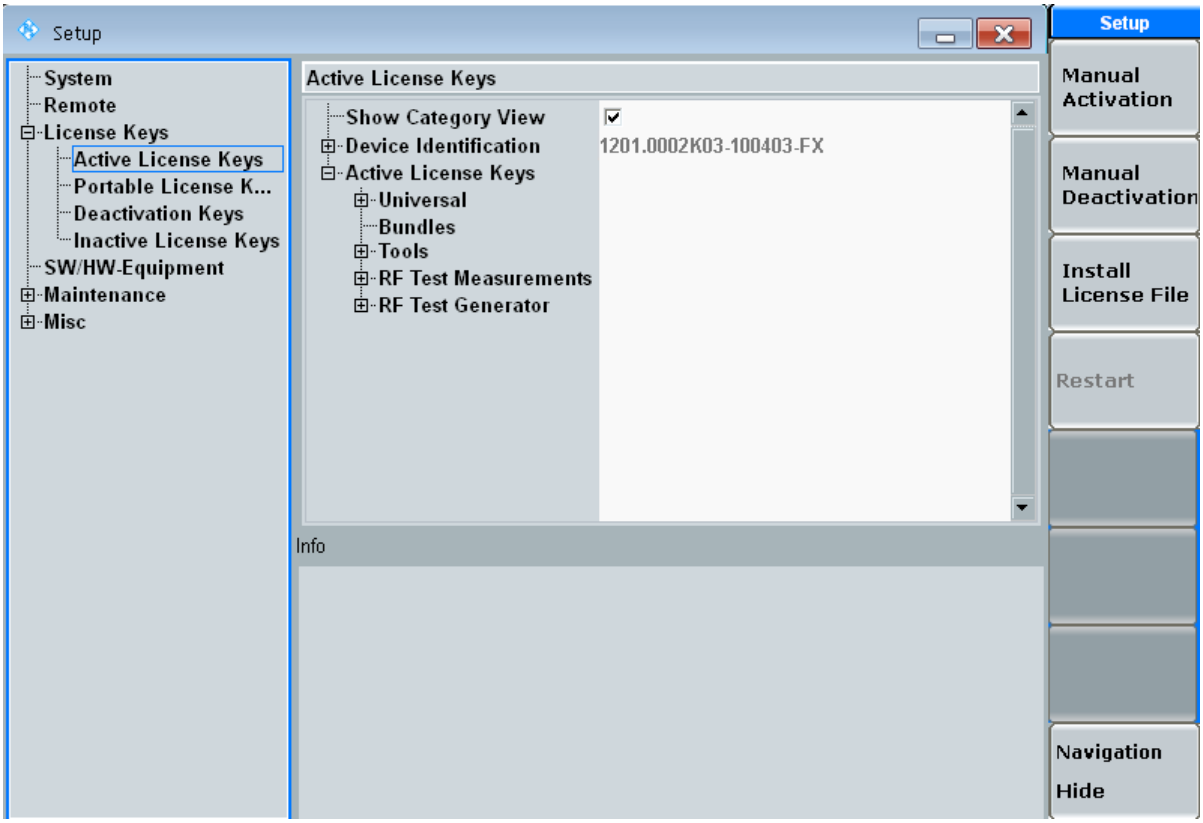


Figure 6-10: License Keys section

The "Device Identification" at the top of the sections is required for identification in the online tool R&S License Manager.

For variant K03 and K06, this ID identifies a complete radio test head with integrated smart card. For variant K02, the ID identifies only the smart card (K31).

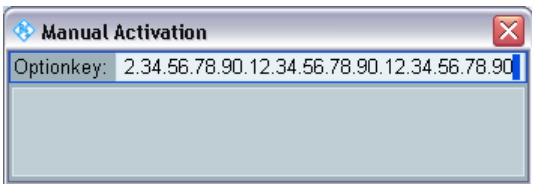
To identify all connected radio test heads, see Chapter 6.5.4, "SW/HW Equipment", on page 98.

6.5.3.1 Softkeys

The "License Keys" section provides the following softkeys.

Manual Activation
Use this softkey to apply an activation key by manually typing in the activation key code.

A dialog box is opened for this purpose. The separating dots are entered automatically.



Manual Deactivation
Use this softkey to apply a deactivation key by manually typing in the deactivation key code.

A dialog box is opened. Enter the deactivation key code and save the response key to a file.

You can save the response key from the dialog box, or alternatively later on, see "Save Response Key to File" on page 96.

Save the response key to a USB memory stick or to the CMW user data directory on the system drive, folder `ResponseKeys` (for example `C:\ProgramData\Rohde-Schwarz\CMW\Data\ResponseKeys`).

The key confirms that you have deactivated the license. The online tool R&S License Manager requests the key when moving a portable license.

Install License File
Use this softkey to apply an activation or deactivation key via a registered license file instead of typing in the key code manually.

You can open registered license files from a USB memory stick or from the CMW user data directory on the system drive (for example `C:\ProgramData\Rohde-Schwarz\CMW\Data`).

Press the softkey to open a dialog box, navigate to the file location and select one of the listed keys. Press "Apply" or "OK" to install the key. "OK" closes the dialog box, while "Apply" keeps it open, so that you can install additional licenses.

Restart
Restarts the CMW software. Perform this action after you have applied license keys.

Save to License File
This softkey is available in subsection "Portable License Keys" only.

First select a portable license in the list, then press the softkey to save the license to a file. You can save it to a USB memory stick or to the CMW user data directory on the system drive, folder `ExportedLicenseKeys` (for example `C:\ProgramData\Rohde-Schwarz\CMW\Data\ExportedLicenseKeys`).

The online tool R&S License Manager requests the license file to identify a portable license to be moved.

Save Response Key to File
This softkey is available in subsection "Deactivation Keys" only.

First select a deactivation key in the list, then press the softkey to save the response key for this deactivation key.

Usually, the response key is saved at the end of the deactivation procedure. This softkey allows you to save it later on.

6.5.3.2 Moving a Portable License

As an example for license management, the following procedure describes how to move an active portable license to another device ID.

The example procedure involves the transfer of files between the R&S License Manager and the instruments. You can either use a USB memory stick for this transfer, or a LAN connection. Alternatively you can enter the key codes manually, both in the R&S License Manager dialogs and in the instrument dialogs. The example procedure uses a USB memory stick.

At the beginning of each step, a keyword indicates where the action must be performed:

- Source: instrument/device ID from which the license is moved
- Target: instrument/device ID to which the license is moved
- Manager: R&S License Manager, opened in a browser

Proceed as follows:

1. Source: Saving the portable license to a file.
   a) Open the setup dialog, section "License Keys", subsection "Portable License Keys".
   b) Select the portable license that you want to move.
   c) Press the softkey "Save to License File". Store the portable license to a USB stick.
2. Manager: Identifying the source and target.
   a) Open the R&S License Manager: https://extranet.rohde-schwarz.com/service
   b) Select "Move Portable License".
   c) Enter the device identifications of the source and target.
   You find the IDs in the setup dialogs of the devices, section "License Keys".
3. Manager: Identifying the portable license that you want to move and creating a deactivation key for the source.
   a) Open the portable license file saved in step 1.
   b) Generate and save a deactivation key file. Store it on the USB stick.
4. Source: Deactivating the license and generating a response key.
   a) Press the softkey "Install License File" and install the deactivation key file generated in the previous step.
   b) Save the deactivation response key to a file on the USB stick.
   You can perform this action at the end of the key installation process.
   If the dialog box closes without saving the response, open the subsection "Deactivation Keys". Select the applied deactivation key in the list. Press the softkey "Save Response Key to File" to save the response key to the USB stick.
5. Manager: Verifying that the license is inactive and generating an activation key for the target.
   a) Enter the deactivation response key generated in the previous step. You can copy the key from the stored response key file.
   As a result, you get a portable license file registered for the target.
   b) Store the license file on the USB stick.

6. Target: Press the softkey "Install License File" and install the activation key file generated in the previous step.

   The portable license is now active at the target.

6.5.4 SW/HW Equipment

The "SW/HW Equipment" section of the "Setup" dialog shows the installed software and hardware and lists the available options.

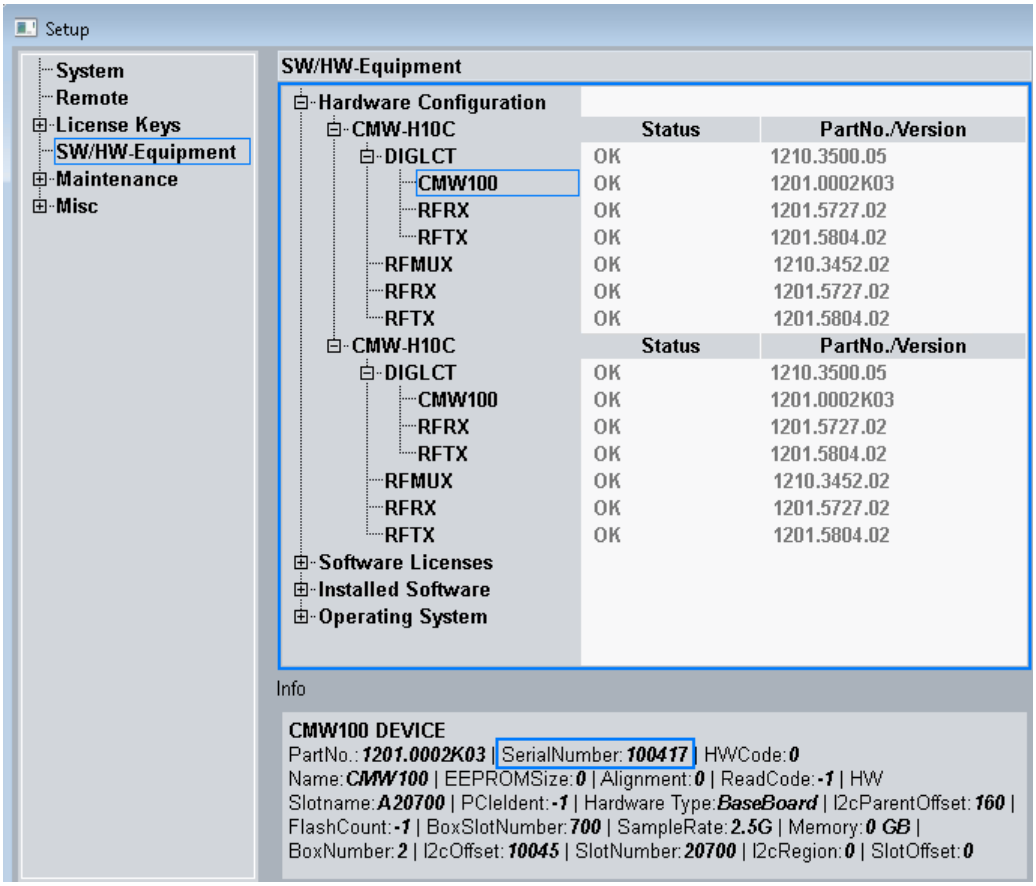


Figure 6-11: SW/HW-Equipment section

The radio test head is listed as "CMW100". You can distinguish several units via their serial numbers. To display the serial number, select the entry CMW100. The number is then displayed in the "Info" area at the bottom. For identification of radio test heads, see also Chapter 6.5.11, "Multi-CMW", on page 112.

Related Commands
Most contents of the "SW/HW Equipment" section can also be queried via remote commands.

Remote command:

```
SYSTem:BASE:OPTion:LIST?
SYSTem:BASE:OPTion:VERSion?
SYSTem:CMW:DEVice:ID?
```

6.5.5 Selftests

The R&S CMW100 provides extensive selftest procedures on module and system level. The selftests are primarily intended for production and service purposes; they are not needed during normal operation of the instrument. The following description serves as a general overview.

6.5.5.1 General Test Features

All selftests are arranged in the "Setup" dialog, section "Maintenance > Selftest".

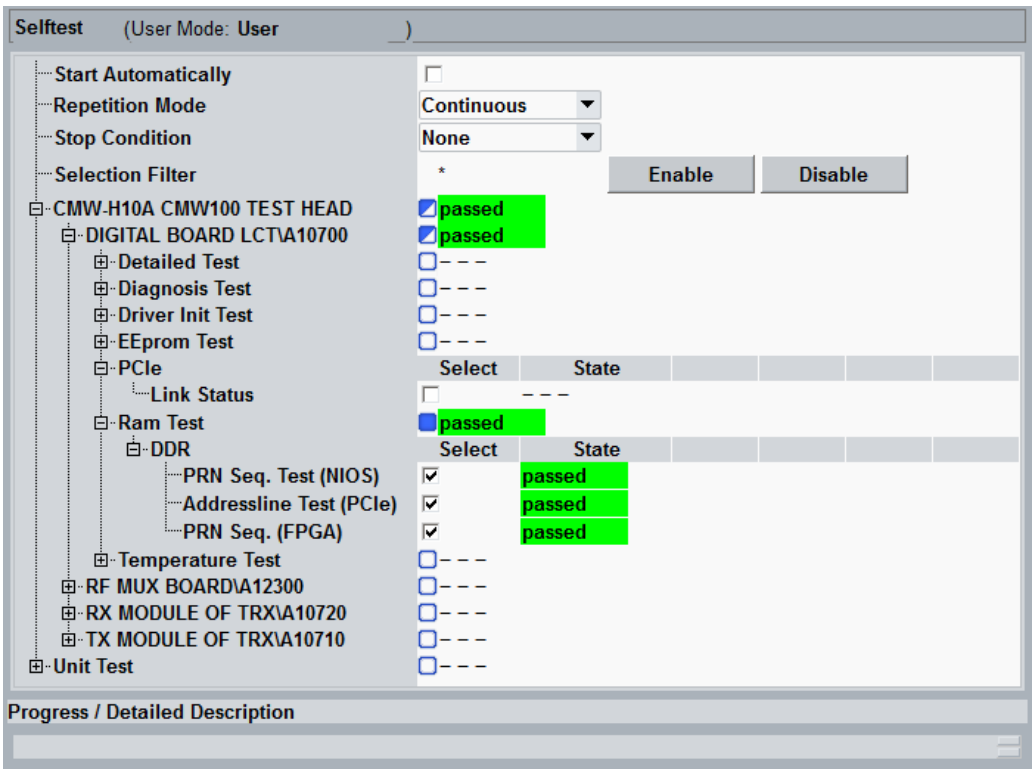


Figure 6-12: Selftest dialog

In the dialog, you can select one or several tests for execution and configure the repetition mode (once or continuously). Moreover, you can save any configuration of the "Selftest" dialog to a profile for later reuse.

The result of each test ("State") is displayed in the individual test section:

- **Invalid**: No test performed yet, no result available
- **In progress**: Test running, no result available yet

- **Passed**: Test passed, all measurements within the factory-set tolerances
- **Failed**: Test failed, one or more measurements out of tolerances
- **Skipped**: Test skipped because it could not be performed with the current instrument configuration

6.5.5.2 Board Tests

A board test verifies that a particular hardware module works properly. Depending on the board, the R&S CMW100 provides different types of board tests described below.

RAM Test
The R&S CMW100 tests one or more of the memories (SRAM, SSRAM, ASRAM, DDR) on the board. Two different types of memory tests are provided.

- For an address line test, the individual address lines of the RAM are activated one after another to transfer a definite number to the RAM. If the instrument can read the transferred and stored numbers correctly after testing all address lines, the address lines must be independent (not interconnected), as required.
- For a pseudorandom (RRAN) test, the R&S CMW100 generates a pseudo-random bit sequence (PRBS) which is first written to the RAM and then read. The R&S CMW100 checks whether the read PRBS is equal to the generated PRBS.

An address line test is faster than a PRAN test. Notice that a failed address line test does not necessarily prove that two address lines are connected. The problem can be due to one of the other components involved in data transmission and storage.

Download Test
The R&S CMW100 loads program data into a DSP or FPGA module and verifies that the module responds properly.

EEprom Test
The R&S CMW100 verifies that the I2C EEPROM data of the board is well-formed (i.e. the syntax is correct, the contents are logically compatible). If no access to the EEPROM data is possible, the board is not displayed in the "Selftest" dialog.

An EEPROM test is provided for all modules.

Driver Init Test
The R&S CMW100 checks whether the operating system has properly initialized the hardware driver for a specific board.

If driver initialization failed, or if the board is defective, the "Driver Init Test" is the only board test shown. The test status is "failed". Contact your Rohde & Schwarz service representative for assistance.

Detailed Test
Detailed tests are board-specific tests that cannot be grouped into one of the previous categories. In general, detailed tests are intended to be performed by R&S service representatives.

6.5.5.3 Unit Tests

A unit test verifies that the communication between several internal modules of the instrument is uninterrupted. A passed unit test usually proves that several modules (e.g. boards and bus systems connecting these boards) work properly.

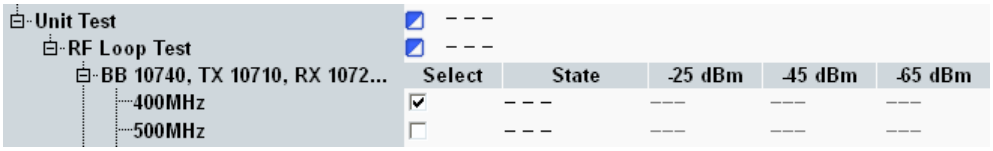
Unit tests and board tests

A unit test represents an efficient method for testing the instrument's overall functioning. After a failed unit test, you can use one of the module tests to pin down the source of the malfunction. Some examples of unit tests are described below.

RF Loop Test

A signal which is generated on a TRX module is routed to an RF connector and back to the TRX module. The main purpose of the test is to verify the connections between the TRX module and the frontend. The RF loop is measured at different frequencies and TX levels.

The header row of the overall loop test shows the output levels in dBm at the RF connectors. The actual measured levels are approximately 6 dB above the equivalent levels at the RF connectors.



6.5.5.4 Performing Selftests

Selftests are controlled like any other measurement. The following example shows an "RF Loop Test" which is appropriate for a general check of the RF path.

To perform the overall loop test:

1. Press the [SETUP] key. Select "Maintenance > Selftest" from the "Setup" dialog.

   The "Selftest" dialog is opened.
2. Select "Repetition Mode: Single-Shot" to perform the test only once.
3. Click the "Disable all" hotkey to clear the predefined selftest selection.
4. Open "Unit Test" > "RF Loop Test" > first "BB..." section.
5. Select the RF frequencies you want to measure.

   Each frequency is measured at several RF output levels.
6. Press the [ON | OFF] key to start the selftest.

The LEDs at the used RF connector light to indicate that an output/input signal is applied. After the selftest is completed, the result at each frequency is shown in the

"State" column. The expected measured levels are approximately 6 dB above the RF output levels.

6.5.5.5 Selftest Control Softkeys

The selftest is turned on or off using the [ON | OFF] or [RESTART | STOP] keys. The measurement control softkey shows the current measurement state.

Save Profile, Load Profile
A profile is a particular selftest configuration that you store for later reuse. The profile contains all enabled selftests together with their user-defined parameters (if there are any) and the Selftest Configuration parameters.

6.5.5.6 Selftest Configuration

The selftest measurement is configured using the parameters at the beginning of the "Selftest" section in the "Setup" dialog.

Figure 6-13: Selftest configuration

Start Automatically
Starts the selected selftests automatically when the R&S CMW100 software is started next time (service feature).

Repetition Mode
Defines how often the selftest measurement is repeated if it is not stopped explicitly or by a stop condition "Halt on Failure".

- In "Continuous" mode, all selected selftests are repeated until the selftest is explicitly aborted; the results are updated after each test cycle: The R&S CMW100 performs a long-term test.
- A "Single Shot" measurement is stopped after all selected tests have been completed. This mode is appropriate for verifying the correct functioning of the instrument.

Stop Condition
Specifies the conditions for an early termination of the selftest measurement.

"None" All selftests are executed according to the selected "Repetition" mode, irrespective of their state.

"Halt On Failure"	Test execution is stopped when one of the selected selftests has reached the "Failed" state, irrespective of the repetition mode set. If none of the test fails, selftest execution is continued according to the selected "Repetition" mode.

Selection Filter
Enables or disables a particular selftest specified via its name. To enable or disable groups of selftests, use the wildcards ? (single character) and * (any character).

As an alternative to the selection filter, use the "Enable/Disable All" or "Enable/Disable Subtree" hotkeys.

Enable/Disable All, Enable/Disable Subtree
Selects all or a current group (subtree) of selftests for execution or clears the current selection.

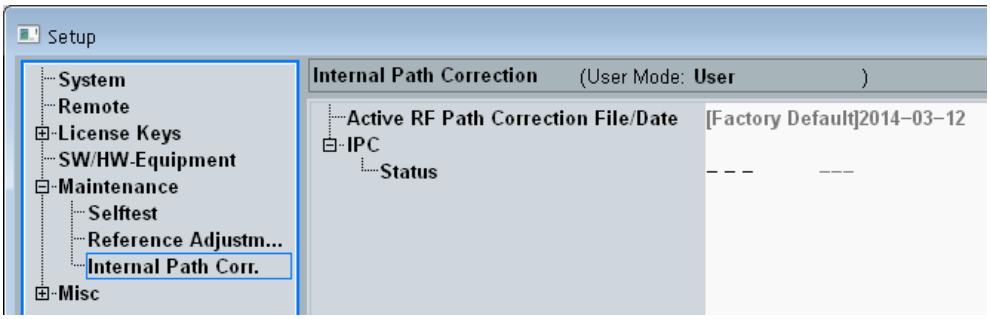
6.5.6 Internal Path Correction

The instrument is calibrated before it leaves the factory. Due to aging, the calibration must be repeated regularly. The recommended calibration interval is stated in the data sheet.

Between two calibrations, you can perform an internal path correction from time to time, to reach the highest level accuracy. For background information, see also Chapter 5.4, "Compensating Aging Effects", on page 74.

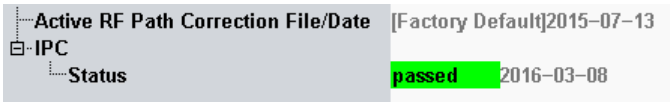
To perform an internal path correction, proceed as follows:

1. In the setup dialog, select the section "Maintenance > Internal Path Corr.".



2. Right-click the softkey "Internal Path Correction". Select "ON" to start the procedure.

   The procedure takes a few minutes. The displayed status changes from "---" via "in progress" to "passed".



The internal path correction data is stored on the system drive of the computer.

The correction data becomes invalid if the radio test head is recalibrated by Rohde & Schwarz. In that case, the status changes to "Off".

If you connect a different radio test head to your computer, for which no correction data is available on the computer, the status changes to "---".

For the status "failed" and the status "Off", available invalid internal path correction data are ignored.

The GUI elements of the "Internal Path Corr." section are described in the following.

Active RF Path Correction File/Date
This information refers to the calibration, not to the internal path correction.

It indicates the name and the creation date of the currently active RF path correction file.

IPC
The line "Status" displays the status of the internal path correction and the date of the correction.

An additional line "Result" is displayed if there are problems. It tells you how to proceed for problem solving, for example, repeat the correction procedure or contact the R&S service.

Possible status values:

"---"	No correction data is available / no path correction has been performed yet with this computer for the connected radio test head.
"OFF"	After the last correction procedure with this computer, the radio test head has been calibrated by Rohde & Schwarz. The correction data are no longer valid and are ignored.
"in progress"	A correction procedure is running.
"passed"	The last correction procedure was completed successfully. The resulting correction data is used.
"failed"	The correction data resulting from the last correction procedure cannot be used. They are ignored.

Remote command:
`FETCh:BASE:IPC?`
`FETCh:BASE:IPC:RESult?`

Initiating a correction
Right-click the softkey "Internal Path Correction" and select "ON" to start the procedure.

Remote command:
`INITiate:BASE:IPC`

6.5.7 IF Equalizer Correction

The instrument provides a fully automatic compensation of the frequency response of its TX and RX modules. Before the instrument leaves the Rohde & Schwarz factory, the frequency response is measured and a correction filter is calculated and stored.

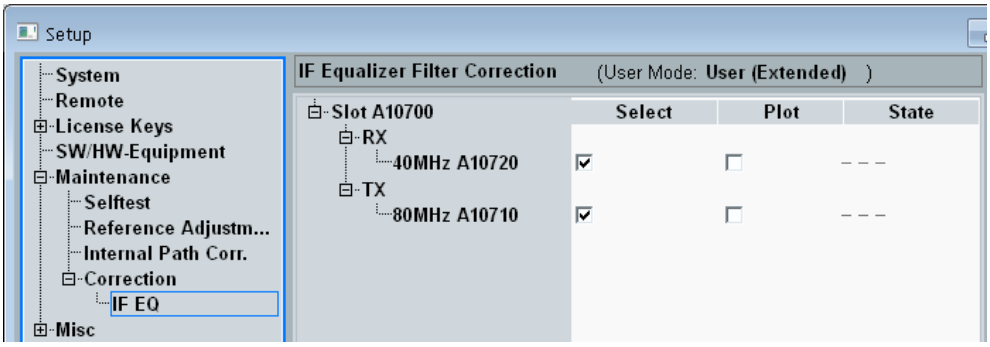
The frequency response of the modules is fairly stable over time. But if necessary, you can update the correction filter. No external equipment is required for this procedure. Proceed as follows:

1. In the setup dialog, press the button "User Mode Login".

   A dialog box opens.

2. Select the user mode "User (Extended)". Close the dialog box.

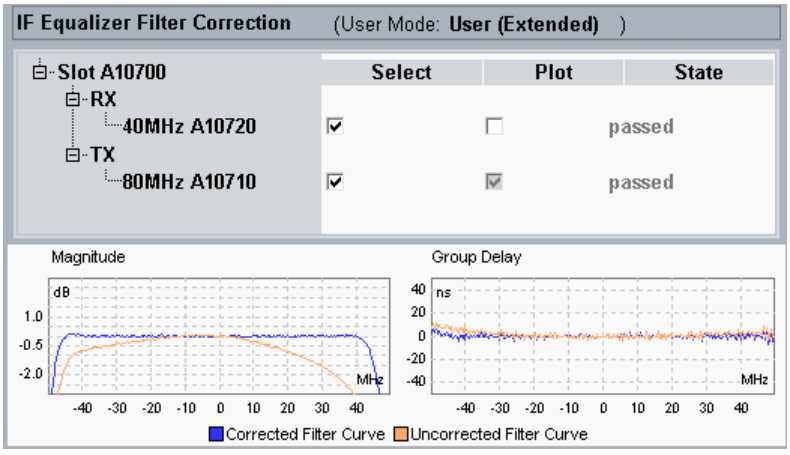
3. Select the section "Maintenance > Correction > IF EQ".



   Do not modify the default settings in column "Select". Keep both checkboxes enabled.

4. Right-click the "IF Equalizer Filter Corr" softkey. Select "ON" to start the procedure.

   First the RX module is measured, then the TX module. The state for each module changes from "---" via "in progress" to "passed".



   The diagrams show the frequency response and the group delay of the module selected in column "Plot". Each diagram shows two curves, with and without applying the calculated correction filter.

5. Close and restart the CMW application.

   The new correction filter is used from now on.

6.5.8 Sync Settings (K06)

The instrument variant K06 can be synchronized to its internal reference frequency source or to an external reference signal. The used frequency source is selected via the setup dialog.

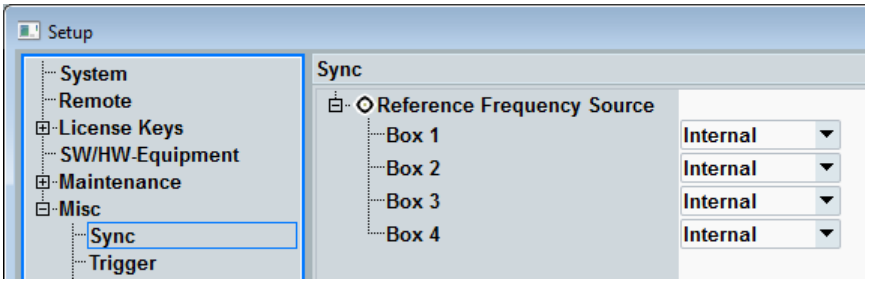


Figure 6-14: Sync settings section

Frequency Source
Selects the reference frequency source to be used. There is one setting per connected radio test head ("Box 1" to "Box 4").

The 10-MHz reference signal used by a radio test head is also available at the reference output port of that radio test head, for synchronization of other instruments.

"Internal" The radio test head uses its internal oven quartz (OCXO).

"External" The radio test head uses an external 10-MHz reference signal fed to its reference input port. The signal of the OCXO is ignored.
The external reference signal must meet the specifications of the data sheet.

Remote command:
`SYSTem:BASE:REFerence:FREQuency:SOURce`
`SYSTem:BASE:REFerence:FREQuency<n>:ADVanced:SOURce`

6.5.9 Trigger (K06)

This dialog configures the trigger output port on the rear panel of the instrument variant K06 and allows you to initiate the generation of a "User Initiated Trigger" signal.

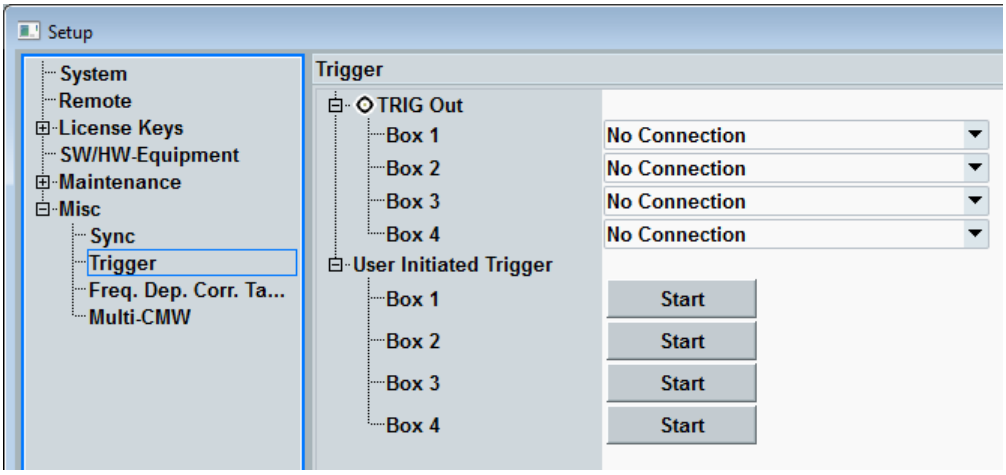


Figure 6-15: Trigger section

TRIG Out
Selects the source for the output trigger signal. There is one setting per connected radio test head ("Box 1" to "Box 4").

"No Connection" means that no output trigger signal is fed to the connector.

Most trigger sources depend on the installed options. For example, the ARB generator provides several output trigger signals to synchronize external devices to a processed waveform file.

Remote command:
`TRIGger:BASE:EOUT<n>:CATalog:SOURce?`
`TRIGger:BASE:EOUT<n>:SOURce`

User Initiated Trigger
There is one user-initiated trigger per connected radio test head ("Box 1" to "Box 4"). The "Start" button initiates the trigger event.

A user-initiated trigger signal is useful if you want to trigger several applications simultaneously and the absolute trigger timing is irrelevant.

Use case example: You want to synchronize the ARB files in the GPRF generators of several radio test heads to generate a MIMO signal. Route the user-initiated trigger of one radio test head to its trigger output port. Connect the trigger output port to the trigger input ports of all involved radio test heads. In the GPRF generator instances, select the external trigger as trigger source.

There is also a continuous trigger signal with a trigger pulse every 10 ms ("Cont.10ms Trigger"). It can also be used for synchronization of several GPRF generators.

Remote command:
`TRIGger:BASE:UINitiated<n>:EXECute`

6.5.10 Frequency-Dependent Correction Tables

This section administrates correction tables for frequency-dependent attenuation/gain. For additional information concerning the usage of the tables, refer to RF Path Settings (Generators) and Connection Control (Measurements).

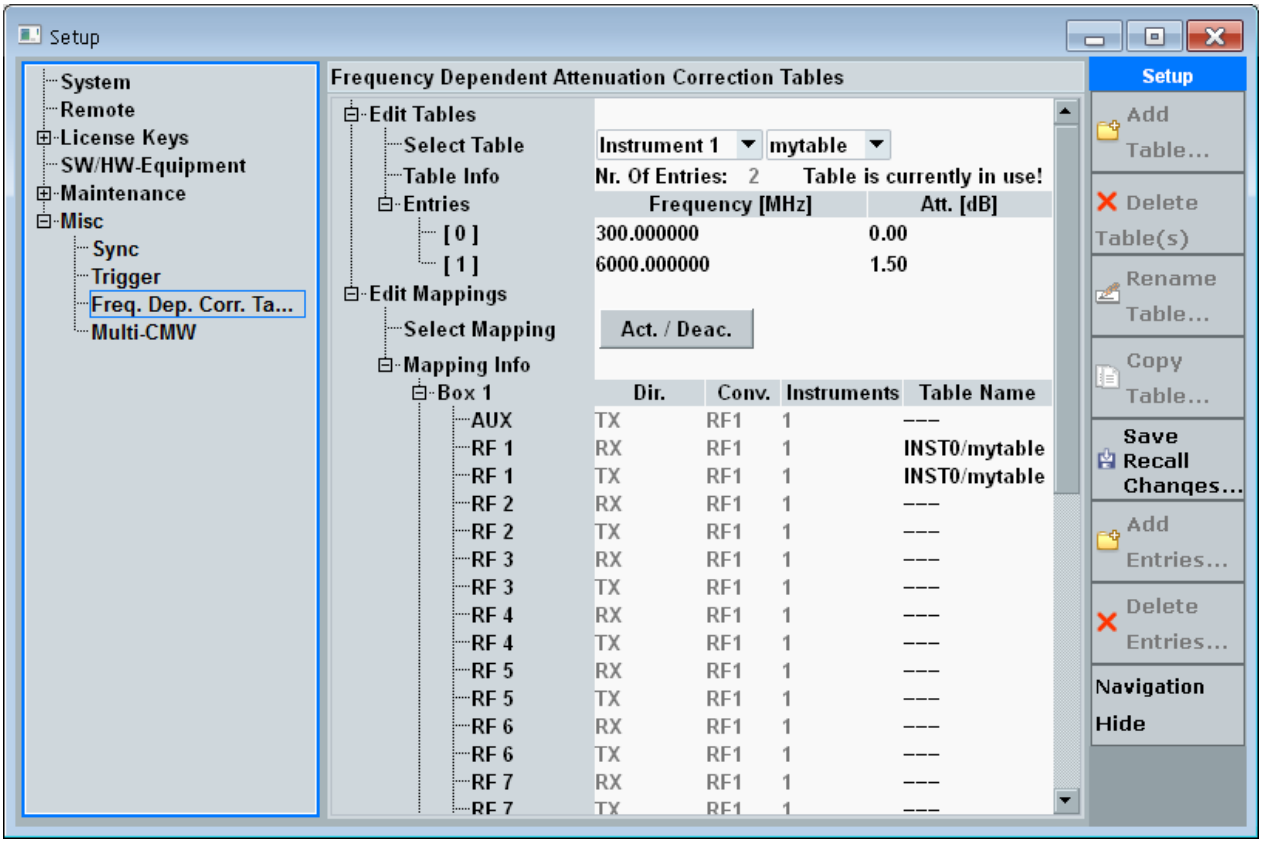


Figure 6-16: Frequency-Dependent Correction Tables section

Edit Tables
This section selects a correction table and displays its entries.

You can edit existing table entries directly. To add or delete entries, use the corresponding softkeys.

To rename, copy or delete a table, use the softkeys.

All correction tables are defined subinstrument specific. Within each subinstrument, the table names are unique. You can create two tables with the same name in different subinstruments, but not in the same subinstrument.

You can use all tables for all subinstruments, independent of for which subinstrument you have defined the table.

Select Table ← Edit Tables
Selects the correction table to be edited, copied or renamed.

To select a table, select first the subinstrument ("Instrument n"), then select the table name.

Remote command:
`CONFigure:BASE:FDCorrection:CTABle:CATalog?`
`CONFigure:BASE:FDCorrection:CTABle:COUNt?`

Table Info ← Edit Tables
Displays the number of entries in the selected table.

If "Table is currently in use!" is displayed, this message means that the table is assigned to at least one RF path in section "Edit Mappings".

Remote command:
`CONFigure:BASE:FDCorrection:CTABle:LENGth?`

Entries ← Edit Tables
Displays the entries of the selected table. Each entry consists of a frequency and an attenuation value.

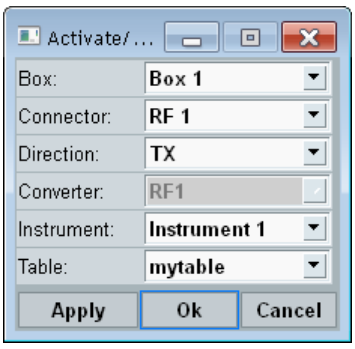
The table is sorted automatically with increasing frequencies. Each entry must have a different frequency.

Remote command:
`CONFigure:BASE:FDCorrection:CTABle:DETails?`

Edit Mappings
This section configures and displays the assignment of correction tables to RF signal paths.

A global reset or preset deletes all assignments. A reset or preset of a subinstrument or application does not affect the assignments.

Select Mapping ← Edit Mappings
The button opens a dialog box for assigning correction tables to RF signal paths.



The upper settings select a connector ("Box", "Connector", "Direction", "Converter").

The lower settings select the correction table assigned to the connector (subinstrument for which the table has been defined and name of the table).

Remote command:

```
CONFigure:FDCorrection:ACTivate
CONFigure:FDCorrection:DEACtivate
CONFigure:FDCorrection:DEACtivate:ALL
CONFigure:CMWS:FDCorrection:ACTivate:RX
CONFigure:CMWS:FDCorrection:DEACtivate:RX
CONFigure:CMWS:FDCorrection:DEACtivate:RX:ALL
CONFigure:CMWS:FDCorrection:DEACtivate:ALL
```

Mapping Info ← Edit Mappings
Provides an overview of the correction table assignments.

Each box number corresponds to a physical instrument. Each line shows information for a single input or output RF signal path.

The column "Instruments" shows to which subinstrument a connector belongs. The column "Table Name" shows assigned correction tables, including the subinstrument for which a table has been defined and the table name. "INST0/mytable", for example, refers to the table "mytable" defined for subinstrument number 1 (remote control `INST0`).

Remote command:

```
CONFigure:FDCorrection:USAGe?
CONFigure:CMWS:FDCorrection:USAGe?
```

Softkeys
The following softkeys are related to the correction tables.

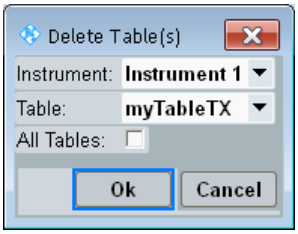
Most softkeys are only active if the section "Edit Tables" is selected.

Add Table ← Softkeys
Creates a correction table without any entries. A dialog box opens for selection of the subinstrument and specification of the table name.

Remote command:

```
CONFigure:BASE:FDCorrection:CTABle:CREate
```

Delete Table(s) ← Softkeys
Deletes a selected table or all tables of a subinstrument. A dialog box opens for selection of the table to be deleted.



If a deleted table was in use, the corresponding assignments in section "Edit Mappings" are removed.

Remote command:
`CONFigure:BASE:FDCorrection:CTABle:DELete`
`CONFigure:BASE:FDCorrection:CTABle:DELete:ALL`

Rename Table ← Softkeys
Renames the table selected in section "Edit Tables". A dialog box opens for specification of the new table name.

If the table is in use, section "Edit Mappings" is updated automatically.

Remote command:
n/a

Copy Table ← Softkeys
Creates a copy of the table selected in section "Edit Tables".

A dialog box opens for selection of the subinstrument for the new table and specification of the new table name.

Remote command:
n/a

Save / Recall Changes, Recall Tables ← Softkeys
When the application software is started, the existing correction tables are loaded from the system drive into the RAM.

Most actions performed in section "Edit Tables" affect only the tables in the RAM, but not the tables on the system drive. This rule applies, for example, to creating a table, copying a table or modifying table entries. Deleting or renaming a table affects both the RAM and the system drive.

When the application software is closed (e.g. by pressing the standby key), all correction tables in the RAM are stored to the system drive automatically.

A "Save Changes" operation triggers this action manually. The tables in the RAM are stored to the system drive.

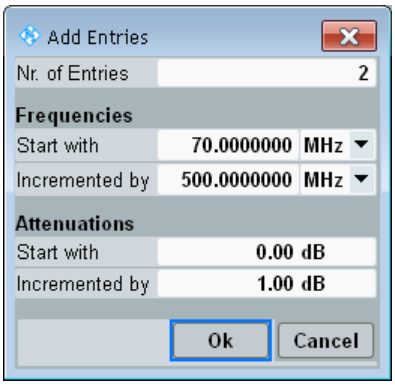
A "Recall Changes / Tables" operation loads all tables from the system drive into the RAM. Thus you can undo not yet saved changes. New tables in the RAM that do not yet exist on the system drive are not deleted from the RAM.

A dialog box opens where you can select whether you want to save or recall changes. If the softkey is labeled "Recall Tables", there are no unsaved changes.

Remote command:
CONFigure:BASE:FDCorrection:SAV

Add Entries ← Softkeys
Adds new entries to the table selected in section "Edit Tables".

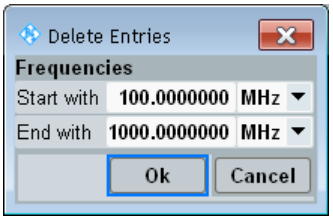
A dialog box opens for specification of the entries.



Remote command:
CONFigure:BASE:FDCorrection:CTABle:ADD

Delete Entries ← Softkeys
Deletes a range of entries from the table selected in section "Edit Tables".

A dialog box opens for specification of the frequency range to be deleted.



Remote command:
CONFigure:BASE:FDCorrection:CTABle:ERASe

6.5.11 Multi-CMW

If you connect several radio test heads to your control computer, the radio test heads are numbered automatically as CMW 1 to CMW n.

The same numbering is used by many other dialogs and remote commands to identify the radio test heads ("CMW 1" to "CMW n" or "Box 1" to "Box n").

You need the assignment, for example, to configure the RF connectors of the individual units.

The "Setup" dialog lists all connected radio test heads and allows you to check which number has been assigned to which radio test head.

You can also reorder the numbering, so that it matches the placement of your radio test heads.

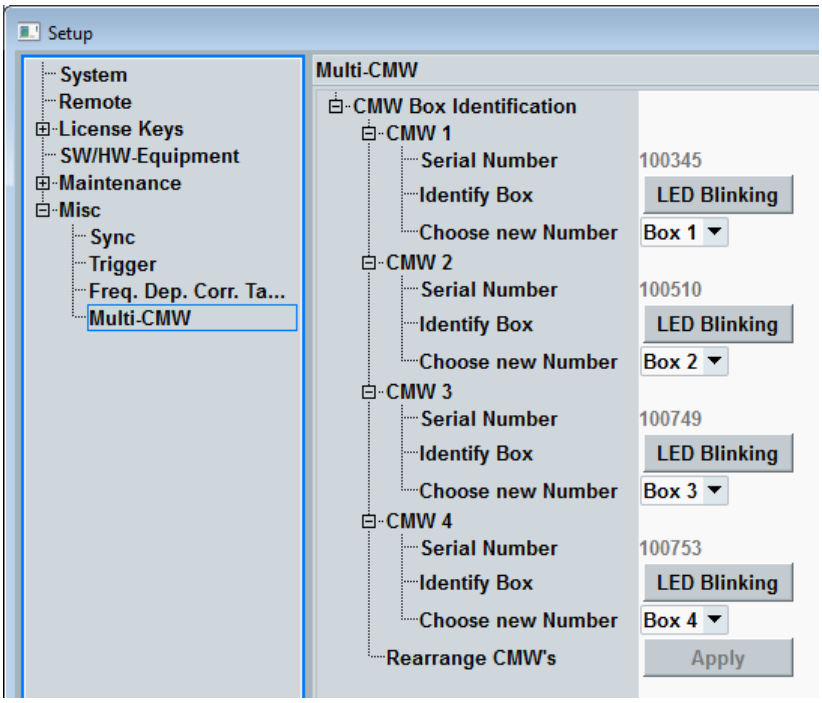


Figure 6-17: Setup - Multi-CMW

Serial Number
Displays the serial number of the radio test head.

Remote command:
`FETCh:BASE:MCMW:SNUMber?`

Identify Box
Lets all LEDs on the front panel of the radio test head blink for some seconds.

Remote command:
`STARt:BASE:MCMW:IDENtify`

Choose New Number, Rearrange CMWs
To change the assignment of box numbers to serial numbers, select the new box numbers via "Choose New Number". Then press "Apply".

The assignment must be unique. Conflicting box number assignments are shown by a red indicator. Green indicators tell you that the new assignment is ok.

Pressing "Apply" triggers the renumbering, including a restart.

Remote command:
`CONFigure:BASE:MCMW:REARrange`

6.6 Print Dialog

The "Print" dialog saves a screenshot to a file.

To open the dialog box, press the [PRINT] key on the (soft-) front panel, or the button.

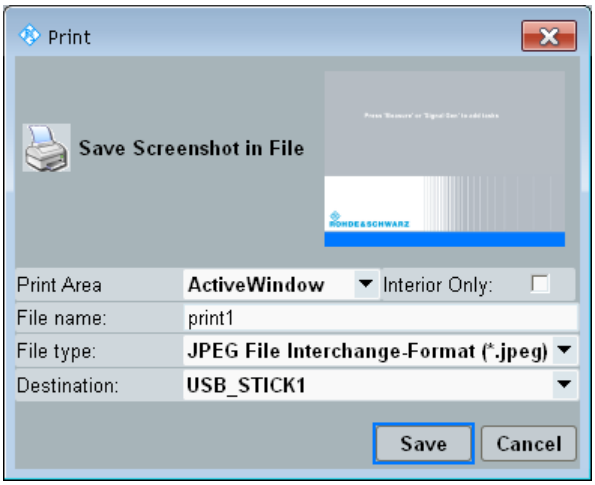


Figure 6-18: Print dialog

Print Area
Selects the part of the screen to be captured. A preview of the resulting screenshot is displayed at the top of the dialog box.

You can capture the currently active window or the main window or the complete screen.

Remote command:
`HCOPy:AREA`

Interior Only
This setting is available if you capture a window in single-window mode.

With disabled checkbox, the entire window is captured. With enabled checkbox, only the interior of the window is captured.

For the "CMW" window, the interior excludes the hotkey bar and the softkey bar.

File Name
Specifies the file name to be used.

A default file name is proposed and the suffix is incremented automatically. You can edit the field to specify a different file name.

File Type
Selects the file format, for example bmp, jpeg or png.

Remote command:
`HCOPy:DEVice:FORMat`

Destination
Selects the folder to which the file is saved.

You can save the file to a fixed predefined destination on the system drive or to a connected USB memory stick.

Save
Press the "Save" button to store the screenshot. The following commands also save a screenshot.

Remote command:
```
HCOPy:FILE
HCOPy:INTerior:FILE
HCOPy:DATA?
HCOPy:INTerior:DATA?
```

6.7 Info Dialog

To open the "Info" dialog box, press the [INFO] key on the (soft-) front panel, or the button.

The dialog box contains several tabs, described in the following. The last tab displays the open-source acknowledgment (OSA).

6.7.1 Current State Tab

The "Current State" tab provides information concerning the current state of all installed firmware applications. At the bottom, it displays the most important address settings for remote control and information related to calibration and RF path correction.

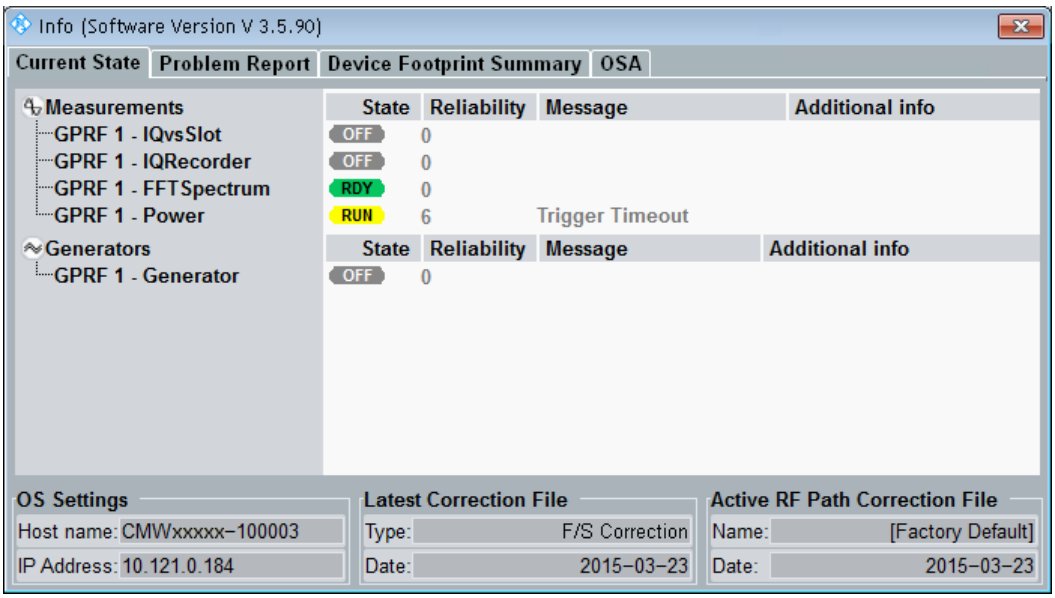


Figure 6-19: Info – Current State tab

State
State of the firmware application, see Chapter 5, "System Overview", on page 50.

Reliability, Message, Additional Info
The reliability indicator describes the validity of measurement results and the possible source of inaccuracies or errors.

The displayed reliability value indicates the most severe error that has been detected by an application since it has been started or switched on. A measurement returns this value also when measurement results are queried via remote control command. When an application is stopped or switched off, the available error information is kept. When it is restarted or switched on again, the "old" error information is deleted.

To display all errors detected by an application, not only the most severe one, select the application on the left and press the hotkey "Reliability List" at the bottom.

A zero in column "Reliability" indicates that no error has been detected. A detected error is indicated via a non-zero value in column "Reliability" and a text in column "Message". Additional information like a file name or an option can be indicated in the last column.

For a description of all reliability indicator values, refer to Chapter 7.4.3.1, "Reliability Indicator", on page 152.

OS Settings
Displays the most important address settings for remote control. For configuration, see Chapter 6.5.1, "System Settings", on page 87.

Latest Correction File
Displays the type and date of the last correction of the instrument. Remote commands allow you also to query information about previous corrections.

Possible types are:
- "F/S Correction": Correction performed in factory or service
- "Calibration": Verification in the factory
- "Outgoing Calibration": Verification by the service

Active RF Path Correction File
Displays the name and creation date/time of the currently active RF path correction file.

6.7.2 Problem Report Tab

The "Problem Report" tab allows you to prepare collected logging information for problem reporting to Rohde & Schwarz.

If you encounter problems with your instrument, proceed as follows to send the generated log files to Rohde & Schwarz:

1. On the (soft-) front panel, press [INFO] to open the "Info" dialog box.
2. Select the "Problem Report" tab.

   The tab shows a list of logging sessions.
3. If necessary, for example because of limitations in the electronic mail system, use the hotkey "Max. File Size" to adjust the maximum size of the compressed log files.
4. Select the directory of the session where the problem occurred. The date and time of the log session refer to the start of the instrument.
5. Press the hotkey "Prepare for sending".

   The selected log files are packed into one or more compressed files located in folder `Output`.
6. Send the files to Rohde & Schwarz for analysis.
   You can copy the files from folder `Output` to a USB memory stick, using the hotkeys "Copy" and "Paste".

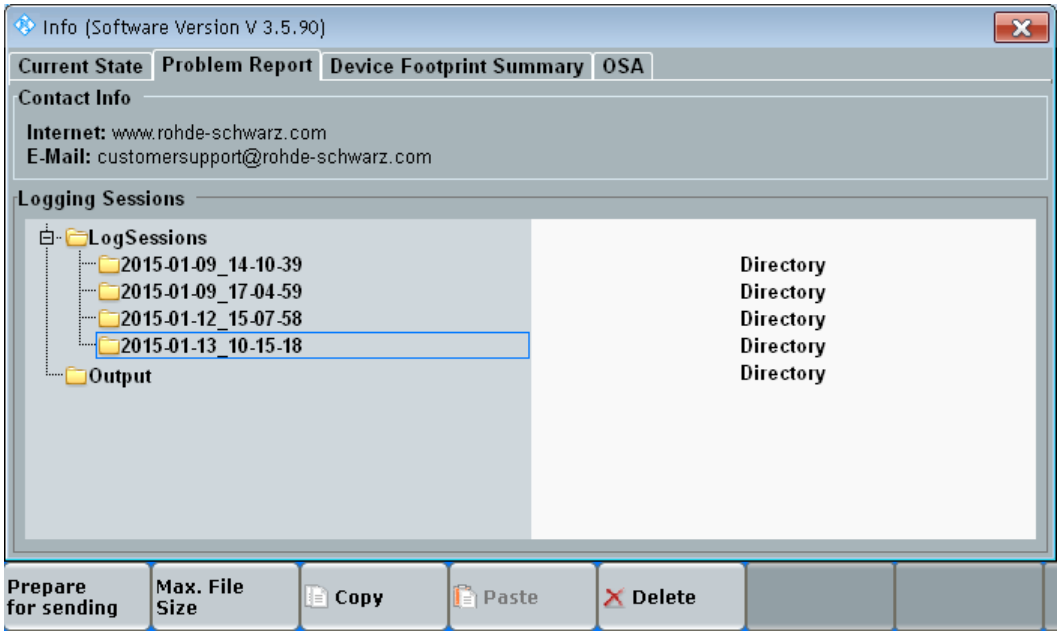


Figure 6-20: Info – Problem Report tab

The dialog provides the following hotkeys.

Prepare for sending
Stores the selected log file directory to one or more compressed files, located in the `Output` directory.

Max. File Size
Sets a maximum file size for the compressed files resulting from the "Prepare for sending" action.

Copy
Copies the selected folder or file to the clipboard. This hotkey can be used, for example, to copy a compressed file located in the output folder.

Paste
Pastes the folder or file located in the clipboard to the selected folder.

Delete
Deletes the selected folder or file.

6.7.3 Device Footprint Summary Tab

The "Device Footprint Summary" tab allows you to export information about the instrument to a PDF file. The PDF file contains information about the hardware, software and licenses of the instrument.

The information is retrieved from the log files of the current session. The date and time in the file name refer to the start date and time of the session, not to the current date and time.

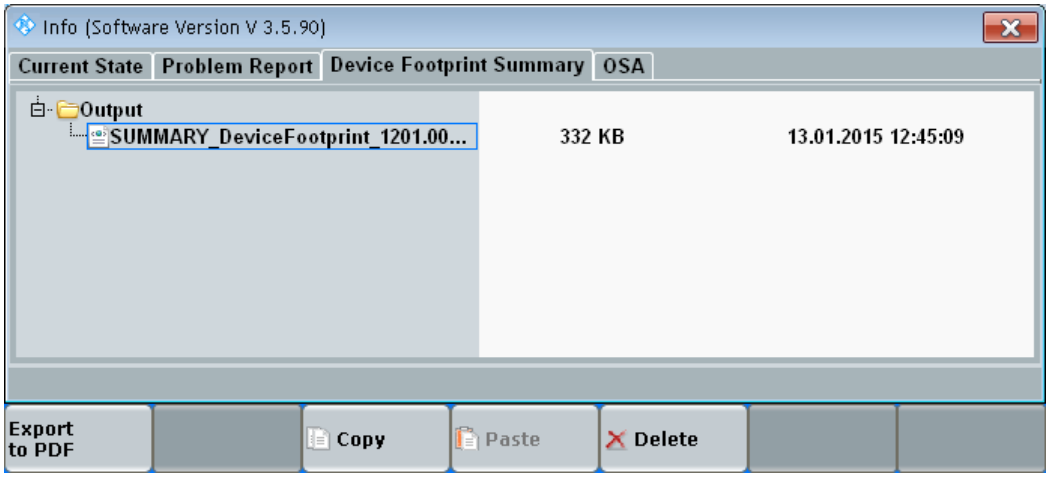


Figure 6-21: Info – Device Footprint Summary tab

To create a footprint PDF file, proceed as follows:

1. Press the [INFO] key on the (soft-) front panel to open the "Info" dialog box.
2. Select the "Device Footprint Summary" tab.
3. Press the hotkey "Export to PDF".

   The PDF file is created and stored in folder `Output`.
4. Copy the file from folder `Output` to a USB memory stick, using the hotkeys "Copy" and "Paste".

6.8 Instrument Setup Dialog

To open the "Instrument Setup" dialog box, press the [DEVICE] key on the (soft-) front panel.

In multiple-window mode, the key and the dialog box are not available. Instead, the settings can be accessed directly from the main window, see Chapter 6.2, "Main Window in Multiple-Window Mode", on page 77.

The settings are only relevant for instruments that can be split into subinstruments. Whether your instrument can be split, depends on the instrument model and the installed options.

A subinstrument can comprise one or two complete radio test heads. In smart channel mode, a subinstrument can comprise one, two or four RF connectors, see Chapter 5.3, "Smart Channel Mode", on page 72.

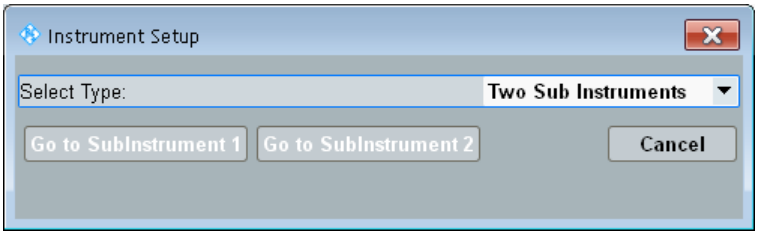


Figure 6-22: Instrument Setup dialog

Select Type
Selects the number of subinstruments. The offered values depend on the available hardware.

Changing this setting also affects the assignment of firmware applications to the subinstruments. The task bar is reset (cleared).

In the factory default configuration, all resources are grouped in a single subinstrument. Splitting the instrument can help to run tasks in parallel.

Remote command:
`SYSTem:BASE:DEVice:MSCont?`
`SYSTem:BASE:DEVice:MSCCount?`
`SYSTem:BASE:DEVice:COUNt`
`SYSTem:BASE:DEVice:RESet`
`SYSTem:BASE:DEVice:SUBinst?`

Go to Sub Instrument n
These buttons select the subinstrument number n for manual control. Remote control of the subinstruments is independent of this setting.

To facilitate the distinction between the subinstruments, the GUIs of the subinstruments use different colors for the background and the active softkeys and hotkeys.

Remote command:
n/a

6.9 Measurement Controller Dialog

The "Measurement Controller" dialog box lists the available measurement firmware applications and allows you to add measurement applications to the task bar.

To open the dialog box, press the [MEASURE] key on the (soft-) front panel.

In multiple-window mode, the key and the dialog box are not available. Instead, the settings can be accessed directly from the main window, see Chapter 6.2, "Main Window in Multiple-Window Mode", on page 77.

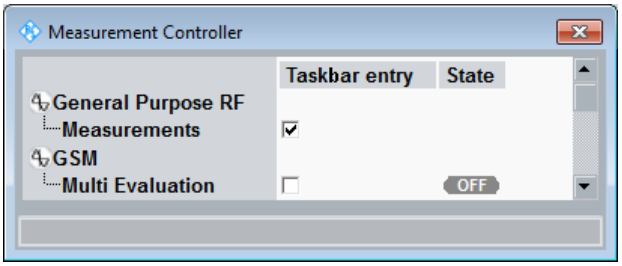


Figure 6-23: Measurement Controller dialog box

Use the checkboxes in the dialog to add or remove a task bar entry for a measurement.

For manual control of a firmware application, the application must be added to the task bar. Remote control of a firmware application is possible without a corresponding task bar entry.

6.10 Generator/Signaling Controller Dialog

The "Generator/Signaling Controller" dialog box lists the available generator and signaling firmware applications and allows you to add the applications to the task bar.

To open the dialog box, press the [SIGNAL GEN] key on the (soft-) front panel.

In multiple-window mode, the key and the dialog box are not available. Instead, the settings can be accessed directly from the main window, see Chapter 6.2, "Main Window in Multiple-Window Mode", on page 77.

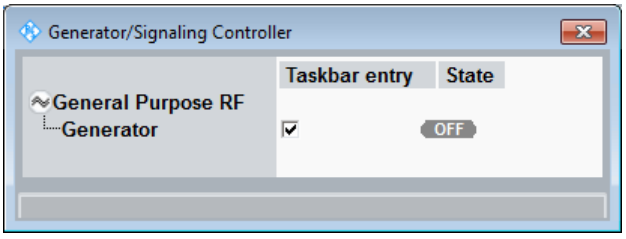


Figure 6-24: Generator/Signaling Controller dialog box

Use the checkboxes in the dialog to add or remove a task bar entry for an application.

For manual control of a firmware application, the application must be added to the task bar. Remote control of a firmware application is possible without a corresponding task bar entry.

6.11 Block View Dialog

The block view provides an overview of the configured signal routing settings.

To open the dialog box, press the [BLOCK VIEW] key on the (soft-) front panel, or the button.

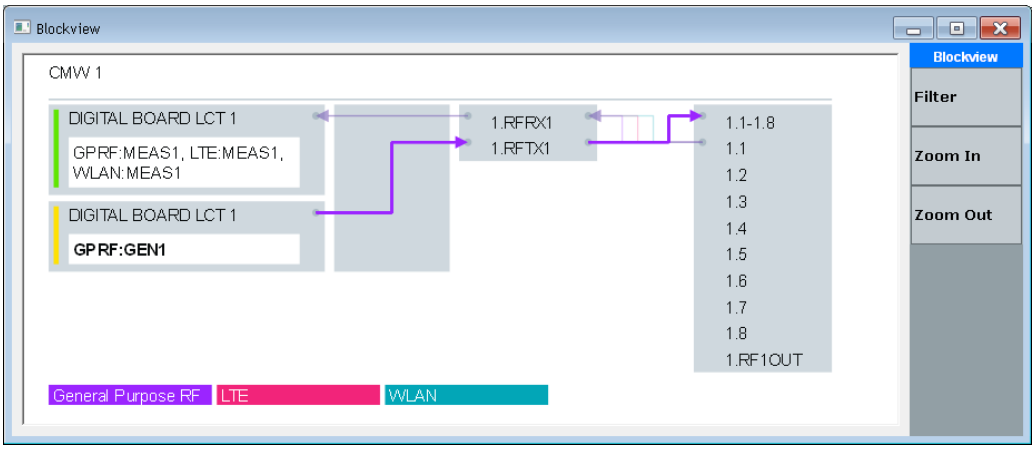


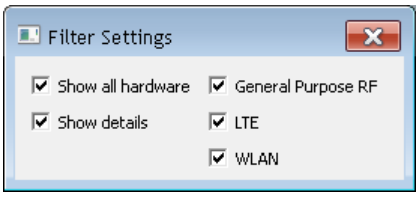
Figure 6-25: Block view

The block view shows the most important hardware blocks relevant for the signal paths. From left to right, the shown hardware blocks are: the units where the firmware applications are running, the TRX modules and the connectors.

Colored lines show the configured signal paths. Bold lines refer to currently used signal paths. The colored boxes at the bottom indicate which line color is used for which technology.

The hardware blocks on the left show the states of the loaded firmware applications. Currently running firmware applications are listed bold. A yellow vertical line means "RUN" or "ON", green means "RDY", red indicates that an error occurred and white means "OFF".

Filter softkey
Opens a dialog box that allows you to hide parts of the diagram.



- "Show All Hardware": Deselect the checkbox to hide unused hardware units (boards, modules and connectors).
- "Show Details": Deselect the checkbox to hide details like, for example, the signaling modules mounted to an SUU.
- Technologies on the right: Deselect a checkbox to hide the corresponding technology.

Zoom In/Out softkeys
Increases or redudes the size of the block view diagram.

6.12 System Messages

System messages give information about exceptional states of the instrument. If possible, the messages also describe necessary user interactions. The R&S CMW100 uses different types of messages, depending on the source and nature of the described situation.

Tooltips

Tooltips are short pieces of text that are displayed in a yellow, rectangular field.

Figure 6-26: Example of a tooltip

If used as a system message, a tooltip usually informs about unusual measurement conditions, e.g. due to a missing signal or inappropriate measurement settings. Check the preconditions for the measurement at the beginning of each measurement description.

System information

System information boxes describe the effects of an action that is about to be performed. The information box still allows you to cancel the action.

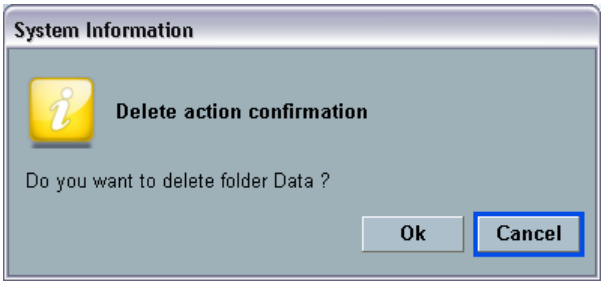


Figure 6-27: System information

System information boxes use different icons, depending on the consequences of the performed action.

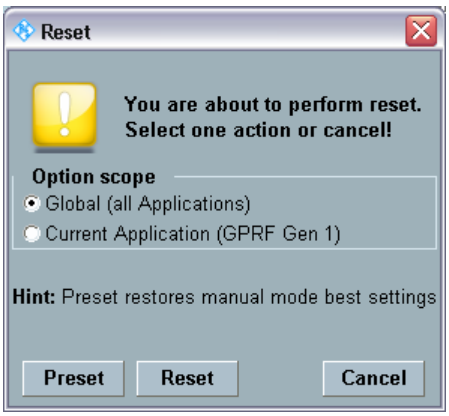


Figure 6-28: System warning

System error message

System error messages are displayed when the R&S CMW100 software needs to be restarted to continue operation. In general, the message box describes the cause of the exception and contains two buttons to shut down or restart the software.

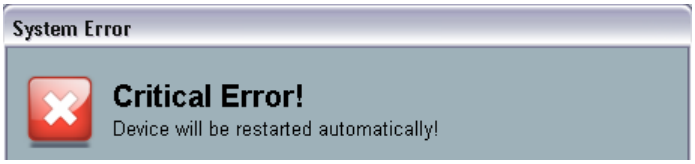


Figure 6-29: Error message

7 Remote Control

This chapter provides instructions on how to set up the tester for remote control. It gives a general introduction to remote control of programmable instruments and describes the tester's remote control concept.

For reference information about all remote control commands implemented by the instrument, refer to the command reference in the documentation of the individual firmware applications.

7.1 Remote Control Operation

7.1.1 Remote Control Interfaces and Protocols

The instrument supports the following interfaces for remote control:

- LAN interface, with HiSLIP protocol, VXI-11 protocol or raw socket mode
  See Chapter 7.1.1.1, "LAN Interface", on page 126

VISA libraries

VISA is a standardized software interface library providing input and output functions to communicate with instruments. High-level programming platforms use VISA as an intermediate abstraction layer. VISA encapsulates the low-level function calls and thus makes the transport interface transparent for the user.

The I/O channel is selected at initialization time via the channel-specific address string or a VISA alias (short name). A VISA installation is a prerequisite for remote control via the LAN interface. All VISA address resource strings are displayed in the "Setup" dialog, see Chapter 6.5.2, "Remote Settings", on page 92.

For more information about VISA, refer to the VISA user documentation.

SCPI compatibility

SCPI commands (standard commands for programmable instruments) are used for remote control. The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of instrument-specific commands, error handling and the status registers. The tutorial "Automatic Measurement Control - A tutorial on SCPI and IEEE 488.2" from John M. Pieper (R&S order number 0002.3536.00) offers detailed information on concepts and definitions of SCPI. The instrument supports the SCPI version 1999.

SCPI-confirmed commands are explicitly marked in the command reference chapters. Commands without SCPI label are instrument-specific. However, their syntax follows SCPI rules.

7.1.1.1 LAN Interface

If you want to control the R&S CMW100 via SCPI command scripts, you can execute the scripts on the computer where the test software is running. Alternatively, you can execute the scripts on a remote computer, connected to the test computer via a LAN interface.

In both cases, you use a LAN protocol - even if you execute the scripts directly on the computer where the test software is running and send the commands to "localhost". VISA must be installed on the computer where the SCPI scripts are executed.

The test software supports the following ways of LAN communication:

- HiSLIP protocol: recommended protocol for remote control via LAN, successor of VXI-11
- VXI-11 protocol: developed for test and measurement instruments
- Raw socket mode: synchronous, streaming oriented protocol without support of asynchronous events like service request (SRQ) or device clear (DCL)

HiSLIP protocol

The high-speed LAN instrument protocol (HiSLIP) was defined by the IVI Foundation and adopted by the LXI Consortium as the recommended LAN protocol.

The HiSLIP protocol uses two TCP sockets for a single connection - one for fast data transfer, the other for non-sequential control commands (e.g. `Device Clear` or `SRQ`).

HiSLIP has the following characteristics:

- High performance as with raw socket network connections
- Compatible IEEE 488.2 support for message exchange protocol, device clear, serial poll, remote/local, trigger, and service request
- Uses a single IANA registered port (4880), which simplifies the configuration of firewalls
- Supports simultaneous access of multiple users by providing versatile locking mechanisms
- Usable for IPv6 or IPv4 networks

Note that HiSLIP data is sent to the device using the "fire and forget" method with immediate return. A successful return of a VISA operation such as `viWrite()` does not guarantee that the instrument has finished or started the requested command. It only indicates that the command has been delivered to the TCP/IP buffers.

VXI-11 protocol

The VXI-11 standard is based on the open network computing remote procedure call (ONC RPC) protocol which in turn relies on TCP/IP at the network/transport layer. The TCP/IP network protocol and the associated network services are preconfigured.

TCP/IP ensures connection-oriented communication, where the order of the exchanged messages is adhered to and interrupted links are identified. With this protocol, messages cannot be lost.

Direct socket communication

With direct socket communication, the test application communicates directly with the TCP transport layer.

Service requests and polling are not supported in the raw socket mode. The additional socket modes "Agilent" and "IEEE1174" are available for compatibility reasons. The emulation codes for polling, service request and device clear messages differ for these modes, as listed in the following table.

Table 7-1: Emulation codes supported by the compatibility modes

Purpose	Direction (controller)	Agilent codes	IEEE1174 codes
Poll status byte	send receive	POL\n POL +stb\n	&POL\cr\n &stb\cr\n
Service request	receive	SRQ\n	&SRQ\cr\n
Device clear	send receive (DCL complete)	DCL\n DCL\n	&DCL\cr\n &DCL\cr\n
\n = newline, CHR$(10) \cr = carriage return, CHR$(13)			

For each socket, a data port and a control port are defined. The ports are used as follows:

- The "Raw" mode uses only the data port. This mode provides the best performance.
- The "IEEE1174" mode uses only the data port, even for transfer of emulation codes. The controller must listen for service requests and emulation code responses at the data port.
- The "Agilent" mode supports both the data port and the control port. Using the control port for emulation codes is optional and recommended. Alternatively, the data port can be used for all messages.

7.1.2 Drivers for Graphical Programming Interfaces

Many Rohde & Schwarz customers prefer graphical programming interfaces when writing applications for the R&S CMW100. Examples for such interfaces are LabVIEW and LabWindows/CVI from National Instruments or VEE from Agilent.

As a service, Rohde & Schwarz provides software device drivers free of charge for this purpose. The drivers are available for download from http://www.rohde-schwarz.com.

7.1.3 Establishing and Testing a LAN Connection

In the following example, a LAN connection for remote control of the instrument is set up. The connection is tested via a simple test script.

The steps in detail depend on the test environment in use. The present example is based on a test tool which requires an additional VISA installation.

1. Connect your computer with the test software to the remote controller or to the home/company network.
2. Start the test software.
3. Press the [SETUP] key.

   The "Setup" dialog box opens.
4. On the left, select section "Remote". Note the VISA address string displayed for HiSLIP. Close the dialog.
   In the following, we assume the following VISA address string:
   "TPCIP::10.121.11.192::hislip0::INSTR"
5. Start your test tool. Configure the connection to the instrument using the VISA address string and an alias, e.g. "CMW".
6. Write a test script using the alias. Run the script.

Hostname and IP address

The VISA address strings displayed in the "Setup" dialog contain the hostname instead of the IP address. Use the address string type that is most convenient for you.

The following test script queries the identification string and returns the contents of the error queue:

```
*IDN?
SYSTem:ERRor?
```

On test script execution, the test tool generates the following result log:

```
: Opening new VISA channel: TCPIP::10.121.11.192::hislip0::INSTR
: Connection to TCPIP::10.121.11.192::hislip0::INSTR established!
: Session handle: 0
: VISA Resource-Identifier: TCPIP::10.121.11.192::hislip0::INSTR
```

```
: send_Query(0, "*IDN?")
: [-->TCPIP::10.121.11.192::hislip0::INSTR] *IDN?
: read_Answer(0, ..., False)
: [<--TCPIP::10.121.11.192::hislip0::INSTR] Rohde&Schwarz,CMW,
1201.0002k50/624376,1.0.0.0
: send_Query(0, "SYSTem:ERRor?")
: [-->TCPIP::10.121.11.192::hislip0::INSTR] SYSTem:ERRor?
: read_Answer(0, ..., False)
: [<--TCPIP::10.121.11.192::hislip0::INSTR] 0,"No error"
```

7.1.4 Switching between Manual and Remote Control

On power-up, the instrument is always in the manual operating state. It can be operated via the graphical user interface (GUI) and the (soft-) front panel controls.

The instrument is switched to remote control when it receives a command from the controller.

While remote control is active, the instrument settings are optimized for maximum measurement speed. The normal display is "switched off", i.e. it shows a "Remote" dialog with hotkeys described in the following sections. All other operation via the (soft-) front panel is disabled.

7.1.4.1 Using the Display during Remote Control

It is possible to switch on the display during remote control. This feature allows you to observe the GUI, e.g. measurement results, while a remote control script is executed and the control elements on the (soft-) front panel are still disabled.

You can switch on the display manually or via remote control:

- Manually: Press the "Hide Remote Screen" hotkey

- Remote command: `SYSTem:DISPlay:UPDate ON`

Switch off the display again before closing the remote connection:

- Manually: Press the "Show Remote Screen" hotkey

- Remote command: `SYSTem:DISPlay:UPDate OFF`

Switching on the display is ideal for programming test purposes but tends to slow down the measurement. Therefore switch off the display in real measurement applications where a tested script is to be executed repeatedly.

7.1.4.2 Returning to Manual Operation

The R&S CMW100 returns to manual operation when the remote connection is closed.

The return to manual operation can also be initiated manually or via remote control:

- Manually: Press the "Go To Local" hotkey

  If an instrument is split into subinstruments, you can return to any of these subinstruments. The remote screen displays several hotkeys for this purpose.

- Via HiSLIP protocol: GTL interface message
- Via VXI-11 protocol: &GTL interface message

A "Go To Local" can be useful while a remote program pauses, e.g. to check the current instrument state in the GUI. When the program continues sending messages, the R&S CMW100 switches back to remote control immediately.

Local lockout

You can prevent the instrument from returning to manual control using a local lockout message:

- Via HiSLIP protocol: LLO interface message
- Via VXI-11 protocol: &LLO interface message
- Many instrument driver commands also contain a local lockout command, e.g. the NI commands `SetRWLS` (set remote with lockout state) or `SendLLO`.

In the local lockout state, an unintentional return to manual control is not possible. All "Go To Local" options listed above are blocked (hotkey and remote control). The "Go To Local" hotkey is replaced by a disabled "Local Lockout" hotkey.

The local lockout can be disabled as follows:

- Via HiSLIP protocol: deactivate the REN control line
- Via VXI-11 protocol: &NREN interface message

7.1.5 Monitoring the Remote Control Interface

You have several possibilities to monitor the control interface:

- Create a report file. To enable or disable the creation of report files, use the "Report File" hotkey displayed while remote control is active.

  Remote command:
  `TRACe:REMote:MODE:FILE<inst>:...`
  See Chapter 8.8, "Tracing the Remote Control Interface", on page 203

- Access the SCPI remote trace directly via the "Remote" dialog, displayed while remote control is active. To configure the display, use the "Report Display" hotkey.

The hotkey toggles between the following modes:

– OFF: Tracing is disabled.
– LIVE: Messages are traced and displayed.
– ANALYSE: Stop tracing to analyze already traced messages.

Remote command:
`TRACe:REMote:MODE:DISPlay:...`
See Chapter 8.8, "Tracing the Remote Control Interface", on page 203

7.1.6 Multiple Channels for Remote Access

Several remote channels can be used simultaneously. The R&S CMW100 supports up to four parallel channels. No restriction is placed upon the combination of channels: It is possible to combine several channels of the same type or channels of different types.

Example:

Suppose that you are executing a remote script to perform a relatively time-consuming measurement, and that you wish to query the instrument state while the measurement is running. If you open a second remote channel for status register handling and monitoring, you do not need to change your script.

The following remote control resources are channel-specific:

- Error queue
- Input and output buffer
- All programmable status register parts

Instrument settings and system resources

Instrument settings can affect several active remote channels. Be cautious in using scripts containing reset (*RST), save (*SAV) or recall (*RCL) commands, as a sudden change of the instrument settings can impair processes controlled through other channels.

Keep in mind that processes controlled through different remote channels can still share instrument resources.

7.1.6.1 Status Registers for Different Channels

Each status register consists of five different parts:

- The CONDition parts of the lowest-level registers are continuously updated by the instrument. These register parts are not programmable. They are an instrument resource which is shared by all remote channels.

- The PTRansition and NTRansiton parts, the EVENt parts, and the ENABle parts determine how the CONDition bits from the lower-level registers are passed on to higher registers. These status register parts can be programmed individually for each remote channel.

Examples for channel-dependent registers

The event status register (ESR) is similar to the EVENt part of a SCPI register. It indicates instrument events. This register is channel-dependent. It is cleared upon reading it.

The event status enable (ESE) register can be programmed individually for different remote channels. The contents of the status byte (STB) and the conditions for the R&S CMW100 to initiate a service request (SRQ) are also channel-specific.

7.2 Device Messages

The general structure of the device messages is defined by the SCPI standard. For specific features of the R&S CMW100 command set, refer to section R&S CMW Command Structure.

For device messages, the ASCII character set is used. A distinction is made according to the direction in which device messages are transferred:

- **Commands**
  Messages the controller sends to the instrument. They operate the device functions and request information.
- **Device responses**
  Messages the instrument sends to the controller after a query. They can contain measurement results, instrument settings and information on the instrument status.

Commands are subdivided according to two criteria:

- The effect they have on the instrument:
  - **Setting commands**
    Cause instrument settings such as a reset of the instrument or setting the output level to some value.
  - **Queries**
    Cause data to be provided for output, e.g. for identification of the device or polling the active input.
- Their definition in standard IEEE 488.2:
  - **Common commands**
    Have a function and syntax that is exactly defined in standard IEEE 488.2. Typical tasks are the management of the standardized status registers and reset.
  - **Instrument-control commands**
    Are functions that depend on the features of the instrument such as frequency settings. Most of these commands have also been standardized by the SCPI consortium.

The device messages have a characteristic structure and syntax, see SCPI Command Structure and Syntax. In the command reference chapters, all commands are listed and explained in detail.

7.2.1 SCPI Command Structure and Syntax

SCPI commands consist of a so-called header and, usually, one or more parameters. The header and the parameters are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). The headers consist of several mnemonics which are separated by colons. Queries are formed by appending a question mark to the header.

SCPI defines two command types with different syntax: common commands and instrument control commands.

7.2.1.1 Common Commands

Common commands are device-independent and consist of a header preceded by an asterisk "*" and possibly one or more parameters.

The following table shows some examples. For a comprehensive list, refer to Common Commands.

Command	Description
*RST	RESET, resets the subinstrument.
*ESE 253	EVENT STATUS ENABLE, sets the bits of the event status enable registers.
*IDN?	IDENTIFICATION QUERY, queries the instrument identification string.

7.2.1.2 Instrument-Control Commands

Instrument-control commands are based on a hierarchical structure and can be represented in a command tree. The command headers are built with one or several mnemonics (keywords). The first level (root level) mnemonic identifies a complete command system.

Example:

`SOURce...` This mnemonic identifies the `SOURce` command system which provides generator settings.

For commands of lower levels, the complete path must be specified, starting on the left with the highest level, the individual keywords being separated by a colon ":".

Example:

```
SOURce:GPRF:GENerator:STATe ON
```

This command is on the fourth level of the `SOURce` system. It turns on the RF generator signal.

In the following, some rules are described, simplifying / abbreviating the command syntax and helping you to read the command reference sections.

Repeated mnemonics with different meaning

The same mnemonic used on different command levels does not necessarily have the same meaning. The actual meaning of a mnemonic depends on its position in the command header.

Example:

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1GHZ
```

This command contains the mnemonic `SOURce` in the first command level ("define RF generator settings"). The command defines the frequency of the GPRF generator signal.

```
CONFigure:GPRF:MEASurement:POWer:TRIGger:SOURce
```

This command contains the mnemonic `SOURce` in the sixth command level. It selects the source of the trigger events for the GPRF power measurement.

Special characters

- **|**: A vertical stroke in the parameter list characterizes alternative parameter settings. Only one of the parameters separated by | must be selected.
  Example: The following command has three alternative settings:
  `SOURce:GPRF:GENerator:SETTing A | B | C`
- **[ ]**: Parts in square brackets can be omitted when composing the command header. The complete command must be recognized by the instrument for reasons of compatibility with the SCPI standard. Parameters in square brackets are optional as well. They can be entered in the command or omitted.
  Example: The following two commands have the same effect
  `SOURce:GPRF:GENerator[:STATe] ON`
  `SOURce:GPRF:GENerator ON`
- **{ }**: Braces or curly brackets enclose one or more parameters that can be included zero or more times.

Optional mnemonics with suffixes

Do not omit an optional mnemonic if its effect is further specified by a numeric suffix.

Long and short form

The key words have a long form and a short form. The short form consists of all upper-case characters, the long form of all upper case plus all lower case characters. The R&S CMW100 recognizes both the short form and the long form.

Example:

```
SOUR:GPRF:GEN:STAT ON
```

```
SOURce:GPRF:GENerator:STATe ON
```

Case insensitivity

The short form is marked by upper case letters, the long form corresponds to the complete word. Upper case and lower case notation only serves to distinguish the two forms in the manual, the instrument itself is case-insensitive.

Parameters

Many commands are supplemented by a parameter or a list of parameters.

In the command reference, parameters are described by a name (literal) written in angle brackets (<>). This literal merely serves as a parameters description. In an application program, it must be replaced by one of the possible settings reported in the detailed parameter description.

For a description of the parameter types, refer to section SCPI Parameters.

Example: `SOURce:GPRF:GENerator:STATe <Boolean>`

With <Boolean> = ON | OFF

Possible command syntax: `SOURce:GPRF:GENerator:STATe ON`

Numeric suffixes

Symbols in angular brackets (<ch>, <i>, <n>,...) denote numeric suffixes. Numeric suffixes are integer numbers distinguishing various items of the same type. The R&S CMW100 provides numeric suffixes for firmware application instances, signal sources, and so on. If unspecified, a numeric suffix is replaced by 1.

Example:

```
SOURce:GPRF:GENerator:DTONe:OFRequency2 1MHz
```

This command specifies an offset frequency for the second component of the dual-tone GPRF generator signal (`OFRequency2`). `GENerator` without suffix or with suffix 1 denotes the first instance of the GPRF generator.

Information in the command reference sections

All commands are described according to the same scheme. The following information is provided:

- Complete command syntax and parameter list
- Description of the command and its relationship with other commands
- List and description of the parameters with their numerical ranges, default values and default units
- Supported command types (setting command, query)
- Programming example (optional)

The commands are arranged according to the order of parameters in the corresponding dialogs of the GUI. Groups of commands with similar function (e.g. several `READ...?` and `FETCh...?` queries for a single measurement application) are listed with a common command description.

7.2.1.3 Structure of a Command Line

A command line can consist of one or several commands. It is terminated by a <New Line>, a <New Line> with EOI or an EOI together with the last data byte. Some programming languages automatically produce an EOI together with the last data byte.

Several commands in a command line must be separated by a semicolon ";". If the next command belongs to a different command system, the semicolon is followed by a colon.

Example:

```
ROUTe:GPRF:GENerator?; :SOURce:GPRF:GENerator:RFSettings:
FREQuency 1GHZ
```

This command line contains two commands. The first command belongs to the ROUTe system. The second command belongs to the SOURce system.

If the successive commands belong to the same system, having one or several levels in common, the command line can be abbreviated. To this end, the second command after the semicolon starts with the level that lies below the common levels. The colon following the semicolon must be omitted in this case.

Example:

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1GHz; :SOURce:GPRF:
GENerator:RFSettings:LEVel -80
```

This command line is written in its full length and contains two commands separated from each other by the semicolon. Both commands are part of the SOURce:GPRF:GENerator:RFSettings command subsystem, i.e. they have four levels in common.

When abbreviating the command line, the second command begins with the level below SOURce:GPRF:GENerator:RFSettings. The colon after the semicolon is omitted. The abbreviated form of the command line reads as follows:

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1GHz;LEVel -80
```

A new command line must always begin with the complete path.

Example:

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1GHz
```

```
SOURce:GPRF:GENerator:RFSettings:LEVel -80
```

7.2.1.4 Responses to Queries

A query is defined for each setting command unless explicitly specified otherwise. It is formed by adding a question mark to the associated setting command. The following rules apply to the responses:

- The requested parameter is transmitted without header.
  Example: SOURce:GPRF:GEN:RFSettings:EATTenuation?
  Response: 1.000000E+001

- Maximum values, minimum values and all further quantities which are requested via a special text parameter are returned as numerical values.
  Example: `SOURce:GPRF:GENerator:RFSettings:FREQuency? MAX`
  Response: `6.000000E+009`
- Numerical values are returned without their unit. The default unit for each command is reported in the command reference description.
  If you add a unit to the query, this unit is used instead of the default unit.
  Example: `SOURce:GPRF:GENerator:RFSettings:FREQuency?`
  Response: `3.000000E+009` for 3 GHz, default unit Hz used
  Example: `SOURce:GPRF:GENerator:RFSettings:FREQuency? GHz`
  Response: `3.000000E+000` for 3 GHz, specified unit GHz used
- Boolean values are returned as 0 (for OFF/FALSE) and 1 (for ON/TRUE). Possible exceptions to this rule are reported in the command reference description.
  Example: `SENSe:BASE:REFerence:FREQuency:LOCKed?`
  Response: `1`
- Text (character data) is returned in short form (see also next section).
  Example: `SOURce:GPRF:GENerator:BBMode?`
  Response: `DTON` (for `DTONe`, dual tone)

7.2.1.5 SCPI Parameters

Most commands require one or more parameters to specify their function. The parameters must be separated from the header by a "white space". Permissible parameters are numeric values, Boolean parameters, text, character strings and block data. The parameter types and the permissible ranges of values are specified in the command description.

Numeric Values

Numeric values can be entered in any form, i.e. with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The mantissa comprises up to 255 characters, the values must be in the value range -9.9E37 to 9.9E37. The exponent is introduced by an "E" or "e". Entry of the exponent alone is not allowed. In the case of physical quantities, the unit can be entered. Permissible unit prefixes are G (giga), MA (mega), MOHM and MHZ are also permissible), K (kilo), M (milli), U (micro) and N (nano). If the unit is omitted, the default unit is used.

Example:

The following two commands are equivalent:

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1.5GHz
```

```
SOURce:GPRF:GENerator:RFSettings:FREQuency 1.5E9
```

Special numeric values

The texts `MINimum`, `MAXimum`, `DEFault` and `KEEP` are interpreted as special numeric values. A query returns the associated numerical value.

Example:

`SOURce:GPRF:GENerator:RFSettings:FREQuency MINimum`

The query `SOURce:GPRF:GENerator:RFSettings:FREQuency?` returns `70000000`.

The following special values can be used on the R&S CMW100:

- MIN/MAX denote the minimum and maximum value of a range of numeric values.
- DEF denotes the reset value. This value is set by the *RST command.
- KEEP can be used within a list of values to "keep" a value unchanged. Example: To set the third value in a list of five parameters to 10 use KEEP,KEEP, 10,KEEP,KEEP.
- NAN: represents the value 9.91E37. Not a number (NAN) is only sent as device response. This value is not defined. Possible causes are division by zero, subtraction or addition of infinite values, or missing values.

Unless it is explicitly stated in the command description, you can use the special numeric parameters described above for all commands of the R&S CMW100. Other special parameters are not supported.

Boolean Parameters

Boolean parameters represent two states. The ON state (logically true) is represented by ON. The OFF state (logically false) is represented by OFF. Replacement of ON or OFF by 1 or 0 is not supported.

Example:

Setting command: `SOURce:GPRF:GENerator:STATe ON`

Query: `SOURce:GPRF:GENerator:STATe?` returns `ON`.

Some of the remote control commands in the `SYSTem...` and `STATus...` subsystems are not implemented as described above. These commands are not needed for measurements or to generate RF signals.

Text Parameters

Text parameters observe the syntax rules for keywords, i.e. they can be entered using a short or long form. Like any parameter, they have to be separated from the header by a white space. In the case of a query, the short form of the text is provided.

Example:

Setting command: `SOURce:GPRF:GENerator:BBMode DTONe`

Query: `SOURce:GPRF:GENerator:BBMode?` returns `DTON`.

Strings

Strings must be entered within single or double quotation marks (' or ").

Example: `MMEM:MDIR 'C:\test scripts'` or `MMEM:MDIR "C:\test scripts"`

Block Data Format

Block data is a transmission format which is suitable for the transmission of large amounts of data. A command using a block data parameter with definite length has the following structure:

Example: `HEADer:HEADer #45168xxxxxxxx`

The hash symbol # introduces the data block. The next number indicates how many of the following digits describe the length of the data block. In the example above, the four following digits indicate the length to be 5168 bytes. The data bytes follow. During the transmission of these data bytes all "End" or other control signs are ignored until all bytes are transmitted.

A #0 combination introduces a data block of indefinite length. The use of the indefinite format requires an "NL^END" message to terminate the data block. This format is useful when the length of the transmission is not known or if speed or other considerations prevent segmentation of the data into blocks of definite length.

Overview of Syntax Elements

Element	Usage
:	The colon separates the keywords of a command. In a command line, the separating semicolon marks the uppermost command level.
;	The semicolon separates two commands of a command line. It does not alter the path.
,	The comma separates several parameters of a command.
?	The question mark forms a query.
*	The asterisk marks a common command.
', "	Quotation marks introduce a string and terminate it.
#	The hash sign # introduces block data. Block: #21312
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates header and parameter.

7.2.1.6 Use of SCPI Subsystems

The structure of the instrument-control commands implemented by the R&S CMW100 is described in sections General Command Structure and Control of the Instrument. Due to this structure, some SCPI subsystems are used in a specific manner. The following list gives an overview.

Command	Usage
`ABORt...`	Stop measurement, release resources; see ABORt:<Application>:MEASurement<i>
`CONFigure...`	Specify measurement settings
`FETCh...`	Retrieve measurement results (running measurement; see FETCh...? Command), measurement or generator states

Command	Usage
INITiate...	Start measurement; see INITiate:<Application>:MEASurement<i>
READ...	Start new measurement and retrieve results; see READ...? Command
ROUTe...	Select output connectors and signal paths
SOURce...	Specify generator settings
STOP...	Stop measurement, do not release resources; see STOP:<Application>:MEASurement<i>

7.3 R&S CMW Command Structure

The syntax of the remote commands for the R&S CMW100 reflects the instrument's basic software modules. The header of each instrument-control command contains the logical software entity the command is assigned to, eliminating the need of addressing the entities (e.g. firmware applications) separately.

7.3.1 General Command Structure

The instrument-control command headers for the R&S CMW100 firmware applications consist of four parts.

SCPI subsystem (1 mnemonic)	Firmware application (2 mnemonics)	Instance (numeric suffix)	Setting/result (1 or several mnemonics)
SOURce	:GPRF:GENerator	<n>	[:STATe]

The purpose and format of the four command parts is as follows:

- **SCPI subsystem**
  One of the root mnemonics specified in the SCPI standard, indicating the SCPI subsystem. Commands within the same subsystem serve a similar purpose; see Use of SCPI Subsystems. Some root mnemonics are optional.
- **Firmware application (FWA)**
  Two non-optional mnemonics indicating a combination of a network standard (or GPRF for general purpose) and the type of the application, for example measurement or generator; see Firmware Applications.
- **Instance**
  Numeric suffix used to distinguish several FWAs of the same type (e.g. several GPRF generators). The supported value range depends on the instrument model, the FWA, the installed hardware and the installed software options. A suffix 1 can be omitted according to SCPI rules.
- **Setting/Result**
  One or several possibly optional mnemonics indicating the purpose of the command.

Extensions for measurement firmware applications

Many of the measurement firmware applications provide several Measurement Contexts and Views. They are identified by fourth- and fifth/sixth-level mnemonics.

Due to the general structure described above, most R&S CMW100 commands are not SCPI confirmed, however, they follow SCPI syntax rules (see also Remote Control Operation).

7.3.2 Firmware Applications

The R&S CMW100 supports several types of firmware applications (FWA):

- **GENerator**: Generator application, controls and configures RF generators. Example: The GPRF generator generates a flexible RF signal for test purposes.
- **MEASurement**: Measurement application, provides a set of measurements for a specific network standard or general-purpose tests. Example: The GPRF measurement application comprises different RF measurements, e.g. "Power", "External Power Sensor" etc.

In the remote control commands, an FWA is addressed by the second and third-level mnemonics; see General Command Structure. A possible numeric suffix <i> (short for <instance>) behind the FWA mnemonics distinguishes several FWAs of the same type.

7.3.3 Measurement Contexts and Views

Most measurement Firmware Applications are further subdivided into different measurement contexts. In manual control, a measurement context often consists of several views, providing different types of measurement results. In remote control, measurement contexts and views are identified by the fourth- and fifth/sixth-level mnemonics in the command headers, respectively.

Measurement contexts and views are addressed by different types of commands:

- Measurement control commands affect the entire measurement context. The same holds for most measurement configurations.
- Measurement results are assigned to a particular view. The commands used to define the measurement statistics and to retrieve results are also view-specific, which makes it possible to transfer a subset of needed results.

Example for context-specific commands

The following measurement control command affects the `POWer` measurement context in the `GPRF:MEASurement` firmware application:

```
INITiate:GPRF:MEASurement:POWer
```

The `POWer` measurement context appears as a fourth-level mnemonic. No view type is specified.

Example for view-specific commands

The following command retrieves the results in the `EVMagnitude:DSSS` view (EVM for DSSS signals) which is part of the `MEValuation` measurement context in the `WLAN:MEASurement<i>` firmware application:

```
FETCh:WLAN:MEASurement<i>:MEValuation:TRACe:EVMagnitude:DSSS:
CURRent?
```

The `EVMagnitude:DSSS` view is selected via the sixth- and seventh-level mnemonics.

READ and FETCh queries

The `READ` query is view-specific and calculates only the results needed for a particular view. Thus a `READ` query can be faster than the context-specific command sequence `INITiate...; FETCh...?`.

7.4 Control of the Instrument

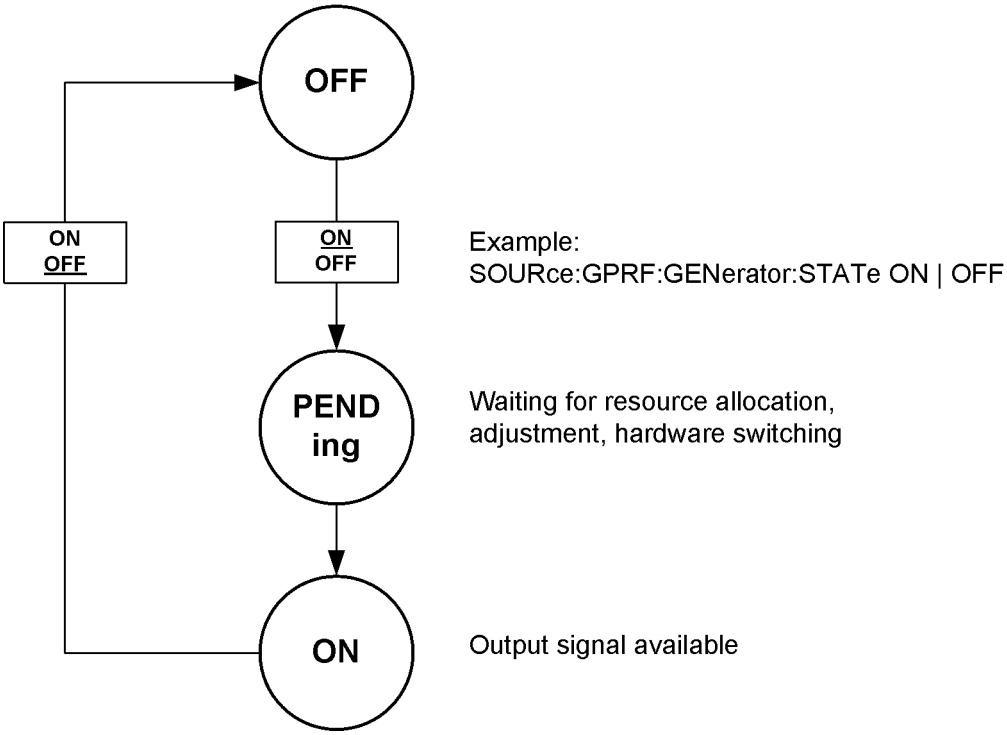
The following sections describe the control of firmware applications.

7.4.1 Generator Control

The commands used to control an RF signal generator are analogous to the commands explained in section Measurement Control. A generator is in one of the following generator states:

- **OFF**: Generator turned off, resources either released or (partially) reserved from the last time the generator was turned on.
- **PENDing**: Generator turned on, but still waiting for resource allocation, adjustment, hardware switching. No output signal is available at the selected output connector.
- **ON**: Generator turned on, with all necessary adjustments finished. An output signal is available at the selected output connector.
- **RDY**: Generator has completed ARB file processing and is turned off. Only relevant in smart channel mode and only if some prerequisites are fulfilled, see Chapter 5.3, "Smart Channel Mode", on page 72.

The OFF and PENDing/ON states correspond to the status indication "Off" and "On" in the generator softkeys. The relationship between generator commands and generator states is shown in the following diagram:



Generator control commands are of the following type (see also Firmware Applications):

SCPI subsystem	<Application>, e.g.	Generator instance	State
SOURce	:GPRF	:GENerator<i>	:STATe ON | OFF

Example: `SOURCe:GPRF:GENerator:STATe ON | OFF`

SOURce:<Application>:GENerator<i>:STATe ON

Starts the generator, reserves all necessary hardware and system resources and changes to the generator state "PENDing", then "ON". If the generator is already turned on, the command has no effect.

If the hardware and system resources are already assigned to another firmware application, this firmware application is released to start the generator; see Resource and Path Management.

If the generator cannot be started due to an unrecoverable resource conflict (e.g. a missing software option), it remains in the OFF state. The SCPI error "-213, Init ignored, ...", is generated. See also Causes for Task Conflicts.

Command synchronization

Before you use the generator signal, use the query `SOURce:<Application>:GENerator<i>:STATe?` to ensure that the generator has reached its `ON` state and that the generator signal is available.

SOURce:<Application>:GENerator<i>:STATe OFF

Switches off the generator, releases the hardware resources for other generators, and changes to the generator state "OFF". If the generator is already turned off, the command has no effect.

7.4.2 Measurement Control

The R&S CMW100 provides various measurements (also termed measurement contexts) for each of the supported network standards or general-purpose applications. All measurements are identified by a fourth level mnemonic and controlled in an analogous way. The benefit of this structure lies in the close analogy of all measurements. Commands for the different measurements have a similar structure and syntax.

The following topics describe the principles of measurement control.

Measurement contexts

Measurement control commands affect the entire measurement context, which can include several views (see Measurement Contexts and Views). The measurement states of all views within the same context are always equal. In contrast, the results of each view can be retrieved separately.

7.4.2.1 Measurement States and Measurement Control Commands

Measurement control commands are used to switch over between the following main measurement states:

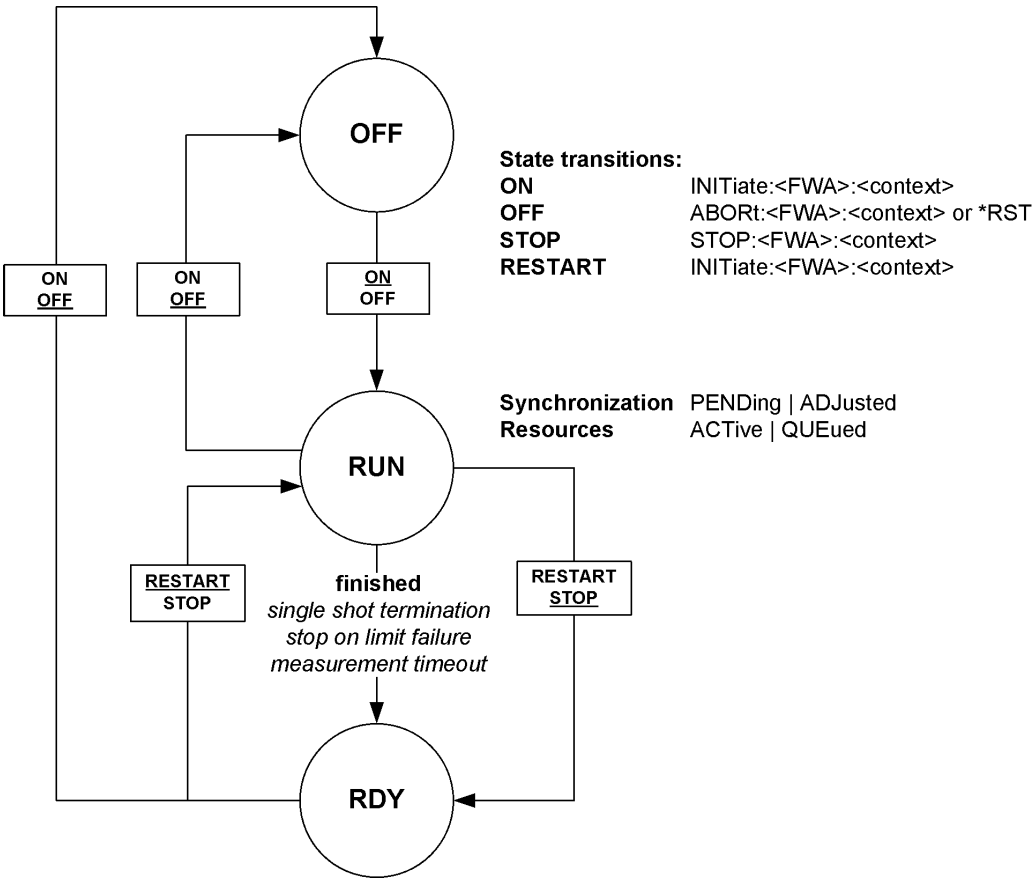
- **OFF**: Measurement switched off, no resources allocated, no results available. OFF corresponds to the SCPI trigger state IDLE.
- **RDY**: Measurement has been terminated. If no error occurred, valid results are available. RDY corresponds to the SCPI trigger state IDLE.
- **RUN**: Measurement running, synchronization pending or adjusted, resources active or queued (see Measurement Substates). RUN corresponds to the SCPI trigger state INITiated.

Querying the main measurement state

The main measurement state can be queried using `FETCh:<FWA>:<context>:STATe?` where `<FWA>` denotes the firmware application and `<context>` is to be replaced by the name of the measurement.

Example: `FETCh:GPRF:MEASurement:POWer:STATe?` (possible response: RDY).

The relationship between measurement states and measurement control commands is shown in the following diagram:



Measurement control commands are of the following type (see also Firmware Applications):

SCPI subsystem	<Application>, e.g.	Measurement instance	Context
INITiate ABORt STOP	:LTE :GPRF	:MEASurement<i>	:POWer :MEValuation ... (depending on <Application>)

Example: `INITiate:GPRF:MEASurement:POWer`

INITiate:<Application>:MEASurement<i>

Starts the measurement in the repetition mode set via `CONFigure:<Application>:MEASurement<i>:<Context>:REPetition` (single-shot or continuous, see Statistical Settings). The command resets the counters for the evaluation period and statistics cycle to zero and starts a timer. If the measurement has not completed the first measurement cycle when the timer expires, the measurement is stopped. The timeout value can be configured individually for each measurement.

`INITiate...` can be used irrespective of the current measurement state. An initiated measurement reserves all necessary system resources and enters the "RUN" state. If the hardware and system resources are already assigned to another firmware application, this firmware application is released to start the measurement; see Resource and Path Management.

If a previously initiated measurement is still running, the new measurement remains in the "PENDing" or "QUEUed" substate (see Queuing of Measurements). It is activated only after the previous measurement has completed the current measurement cycle. After the new measurement has entered the "RUN" state, the previous results are discarded.

If the measurement cannot be started due to an unrecoverable resource conflict (e.g. a missing software option), it remains in the OFF state. The SCPI error "-213, Init ignored, ...", is generated. See also Causes for Task Conflicts.

Conflicting settings cause restart of the measurement

Many measurement parameters (e.g. RF path settings, filter settings, etc.) have a direct impact on the measurement results. Changing these parameters while the measurement is running results in wrong results.

To avoid misleading results, a running measurement is restarted if a parameter with direct impact on the results is changed. All values acquired so far are discarded; the statistics counters are reset to zero.

READ...? command

`READ...?` can be used instead of `INITiate...` to start a single-shot measurement. `READ...?` also returns the results; see Retrieving Measurement Results.

ABORt:<Application>:MEASurement<i>

Aborts the current measurement immediately and causes the measurement to enter the OFF state. All measurement values are set to NAV; the hardware resources are released for other measurements.

STOP:<Application>:MEASurement<i>

Halts the measurement immediately. The measurement enters the RDY state; the R&S CMW100 retains all valid measurement results. Moreover, the hardware and system resources continue to be allocated to the measurement.

7.4.2.2 Measurement Substates

Each running measurement can be in one of the following substates:

Substate	Description
PENDing	Waiting for resource allocation, adjustment, hardware switching
ADJusted	All necessary adjustments finished, measurement running
INV	No substate (invalid)

Substate	Description
QUEued	Measurement without resources, no results available
ACTive	Resources allocated, acquisition of results in progress but not complete
INV	No substate (invalid)

Querying substates

The main measurement state and the substates can be queried using `FETCh:<FWA>:<context>:STATe:ALL?` where `<FWA>` denotes the firmware application and `<context>` is to be replaced by the name of the measurement.

Example: `FETCh:GPRF:MEAS<i>:POWer:STATe:ALL?` (possible response: RUN, ADJ, ACT).

7.4.2.3 Statistical Settings

Measurements generally cover a basic time interval and can be repeated periodically. The measurement interval depends on the measurement context.

The number of measurement intervals that the R&S CMW100 repeats to calculate statistical results is termed "statistic count" (multi-measurement count). After one statistic count, the instrument has terminated a basic measurement cycle (single-shot measurement). Measurement cycles can be repeated for an unlimited number of times, resulting in the continuous repetition mode.

Most measurement contexts provide sets of measurement results corresponding to the current measurement interval (single/scalar values and traces). And they provide the maximum, minimum, and average over several consecutive measurement intervals, see Chapter 5.2.4, "Statistical Results", on page 56. In remote control, these statistical results can be retrieved independently.

Statistic Count

Integer number of measurement intervals per measurement cycle (single-shot measurement). The statistic count can be set independently for any measurement context or view; see Measurement Contexts and Views.

`CONFigure:<FWA>:<Context>:SCOunt:<View> <Count>`

Example: `CONFigure:LTE:MEASurement:MEValuation:SCOunt:MODulation 10`

Sets the statistic count for LTE multi-evaluation modulation measurements.

Repetition Mode

Single-shot: The measurement is stopped after the number of measurement intervals defined by the "statistic count".

Continuous: The measurement is continued until it is stopped explicitly. Average results are calculated according to the rules given in section Statistical Results.

Manual and remote control

In contrast to other instrument settings, the repetition modes in manual and remote control are independent and do not overwrite each other. The default repetition mode in manual control is "Continuous" (observe results over an extended period of time). The default mode in remote control is "Single-Shot" (measure one cycle and retrieve results).

```
CONFigure:<FWA>:<Context>:REPetition <Count>
```

Example: `CONFigure:GPRF:MEASurement<i>:POWer:REPetition CONT`

Statistics Type

In general, the following types of statistical results are available for scalar results (single values) and traces:

Current: Result of the current measurement interval.

Minimum/Maximum: Minimum or maximum of all evaluation periods since the measurement was started.

Average: Average referenced to one single-shot measurement length.

```
FETCh:<FWA>:<Context>:<View>:CURRent...?
READ:<FWA>:<Context>:<View>:CURRent...?
FETCh:<FWA>:<Context>:<View>:MINimum...?
READ:<FWA>:<Context>:<View>:MINimum...?
FETCh:<FWA>:<Context>:<View>:MAXimum...?
READ:<FWA>:<Context>:<View>:MAXimum...?
FETCh:<FWA>:<Context>:<View>:AVERage...?
READ:<FWA>:<Context>:<View>:AVERage...?
```

Example: `FETCh:GPRF:MEAS<i>:POWer:CURRent:RMS?`

Detector

Some measurements provide different detector settings to calculate the returned results from the raw measurement data. An example is the "GPRF Power" measurement which evaluates the maximum, minimum, and RMS (average) power within the current measurement interval (see Detectors). The detector is identified by an additional mnemonic preceding the statistics type:

```
FETCh:<FWA>:<Context>:<View>:CURRent?
FETCh:<FWA>:<Context>:<View>:MINimum:CURRent?
FETCh:<FWA>:<Context>:<View>:MAXimum:CURRent?
READ:<FWA>:<Context>:<View>:CURRent?
READ:<FWA>:<Context>:<View>:MINimum:CURRent?
READ:<FWA>:<Context>:<View>:MAXimum:CURRent?
```

Simplified statistics

Some measurement contexts provide simplified statistical settings. Refer to the measurement and remote control description for details.

7.4.2.4 Retrieving Measurement Results

You can retrieve measurement results by `FETCh...?` or `READ...?` queries. To retrieve error indicators, for example limit check results, use `CALCulate...?` queries.

FETCh...? Command

A `FETCh` command waits until the end of the first measurement cycle before it returns the results. If the first measurement cycle is already complete when the `FETCh` command is sent, results are returned immediately. This behavior ensures that the results are statistically significant.

A `FETCh` command is similar to the corresponding `READ` command. However, it does not start a new measurement. Start the measurement via a `READ` or `INITiate` command, before you send a `FETCh` command.

Measurement states

A `FETCh` command can be used in the RUN state or in the RDY state. If no valid results are available, `NCAP`, `NAV` or `INV` are returned for each unavailable result, see also "Return Values NCAP, NAV and INV" on page 150.

The following table gives an overview of the behavior depending on the measurement state.

State	Valid results?	Effect of FETCh...?
OFF	No	Do not use `FETCh...?` in the OFF state as valid results are never available.
RUN	Yes/No	The R&S CMW100 waits until the first measurement cycle is complete and returns the results.
RDY	Yes/No	The R&S CMW100 returns the results immediately.

READ...? Command

A `READ` command aborts a running measurement and starts a new measurement in single-shot mode. At the end of the first measurement cycle, the measurement is stopped and the results are returned.

`READ` commands can be used in any measurement state.

Performance considerations, multi-evaluation measurements

The `READ` query is view-specific and calculates only the results needed for a particular view. Thus a `READ` query can be faster than the context-specific command sequence `INITiate...; FETCh...?`.

In contrast to `FETCh` queries, `READ` commands also provide valid results for disabled views in multi-evaluation measurements, see Retrieving results for disabled views.

CALCulate...? Command

A `CALCulate` command is used similar to a `FETCh` command. It also waits for the end of the first measurement cycle and does not start a measurement.

A `CALCulate` command analyzes the results that would be returned by the corresponding `FETCh` command and returns an error indicator for each `FETCh` result.

The following table lists all possible error indicator values.

Value	Meaning	Description
`OK`	OK	A measurement result is available. No error has been detected. The result is within the limits or no limit applies to this result.
`ULEU`	User limit exceeded upper	An upper limit is violated. The measurement result is above the limit.
`ULEL`	User limit exceeded lower	A lower limit is violated. The measurement result is below the limit.

Example

Assume that a `FETCh` command returns the reliability indicator, followed by four power values:

```
<Reliability>, <Power1>, <Power2>, <Power3>, <Power4>
```

In that case, the corresponding `CALCulate` command returns the reliability indicator, followed by four error indicators:

```
<Reliability>, <Ind. for Power1>, <Ind. for Power2>, <Ind. for Power3>, <Ind. for Power4>
```

Examples of return values:

- `0, OK, OK, OK, OK`
  No error, all four power results are within the limits
- `0, ULEU, ULEL, OK, OK`
  No error, `<Power1>` is above the upper limit, `<Power2>` is below the lower limit, `<Power3>` and `<Power4>` are within the limits

Return Values NCAP, NAV and INV

If no valid result is available, the value returned by `FETCh`, `READ` and `CALCulate` commands depends on the reason for the unavailability. Possible values are `NCAP` ("not

captured"), NAV ("not available") and INV ("invalid"). For each unavailable result value, one of these values is returned.

The following table provides some examples.

Situation	Returned value
Measurement state OFF	NAV returned by FETCh and CALCulate
Disabled view	NCAP returned by FETCh and CALCulate
Unsuitable settings for a certain part of the measurement. Example: The measurement of some results requires a specific trigger type.	NCAP returned
Object to be measured not available in signal (for example, certain physical channel missing)	NCAP returned
Underflow, overflow, sync error, trigger timeout	INV returned

7.4.2.5 Multi-Evaluation Measurements

In a multi-evaluation measurement, the R&S CMW100 acquires a wide range of measurement results at once.

Multi-evaluation measurements offer maximum speed and performance, even if only a subset of the measurement results is needed. In remote control, it is possible to control each multi-evaluation measurement as a whole but retrieve the different types of results separately.

Controlling multi-evaluation measurements

A multi-evaluation measurement is controlled like any other measurement; see Measurement Control. The following commands start, stop and abort a multi-evaluation measurement within a particular firmware application (<FWA>):

```
INITiate:<FWA>:MEValuation
```

```
STOP:<FWA>:MEValuation
```

```
ABORt:<FWA>:MEValuation
```

The following commands query the measurement state and the substates:

```
FETCh:<FWA>:MEValuation:STATe?
```

```
FETCh:<FWA>:MEValuation:STATe:ALL?
```

Example: `INITiate:LTE:MEASurement<i>:MEValuation`

Retrieving measurement results

The commands for retrieving the results of a multi-evaluation measurement follow general syntax rules (see Retrieving Measurement Results). In general, it is possible to specify the subset of results needed using an additional <View> mnemonic:

```
READ:<FWA>:MEValuation:<View>:<Statistics>...?
```

`FETCh:<FWA>:MEValuation:<View>:<Statistics>...?`

The result subsets are closely related to the different views that the multi-evaluation measurement provides in manual control.

Example: `READ:LTE:MEASurement<i>:MEValuation:EVMagnitude:AVERage?`

Retrieving results for disabled views

In some measurements, it is possible to disable individual views and results to gain measurement speed. The `READ...?` and `FETCh...?` queries act differently on disabled views:

- A `FETCh...?` command leaves the view in the unchanged (disabled) state and returns NCAP (not captured) results. The view must be enabled explicitly to obtain valid results.
- A `READ...?` command enables the view implicitly and returns valid results. After program execution, the view returns to the disabled state.

7.4.3 Reliability and Error Indicators

Reliability and error indicators describe the validity of measurement results and the possible source of inaccuracies or errors.

7.4.3.1 Reliability Indicator

The R&S CMW100 returns a numerical reliability indicator value for each measurement result query. This indicator value allows you to judge the reliability of the returned measurement results.

The value returned by a result query indicates the most severe error that has occurred during the measurement. Tooltips indicate the most severe current error. They are displayed by all types of applications. The "Info" dialog provides an overview of all applications and a list of all occurred errors per application. For some error types, the list provides also additional information on the cause of an error. See also Chapter 6.7, "Info Dialog", on page 115.

Reliability indicator values

The reliability indicator has one of the following values:

- **0 ("No Error")**:
  Measurement values available, no error detected.
- **1 ("Measurement Timeout")**:
  The measurement has been stopped after the configured measurement timeout. Measurement results can be available. However, at least a part of the measurement provides only `INValid` results or has not completed the full statistic count.
- **2 ("Capture Buffer Overflow")**:
  The measurement configuration results in a capture length that exceeds the available memory.

- **3 ("Input Overdriven") / 4 ("Input Underdriven")**:
  The accuracy of measurement results can be impaired because the input signal level was too high / too low.
- **6 ("Trigger Timeout")**:
  The measurement could not be started or continued because no trigger event was detected.
- **7 ("Acquisition Error")**:
  The R&S CMW100 could not properly decode the RF input signal.
- **8 ("Sync Error")**:
  The R&S CMW100 could not synchronize to the RF input signal.
- **9 ("Uncal")**:
  Due to an inappropriate configuration of resolution bandwidth, video bandwidth or sweep time, the measurement results are not within the specified data sheet limits.
- **15 ("Reference Frequency Error")**:
  The instrument has been configured to use an external reference signal. But the reference oscillator could not be phase-locked to the external signal (for example signal level too low, frequency out of range or reference signal not available at all).
- **16 ("RF Not Available")**:
  The measurement could not be started because the configured RF input path was not active. This problem can occur if a measurement is started in combined signal path mode and the master application has not yet activated the input path. The LEDs above the RF connectors indicate whether the input and output paths are active.
- **17 ("RF Level Not Settled") / 18 ("RF Frequency Not Settled")**:
  The measurement could not be started because the R&S CMW100 was not yet ready to deliver stable results after a change of the input signal power / the input signal frequency.
- **19 ("Call Not Established")**:
  For measurements: The measurement could not be started because no signaling connection to the DUT was established.
- **20 ("Call Type Not Usable")**:
  For measurements: The measurement could not be started because the established signaling connection had wrong properties.
- **21 ("Call Lost")**:
  For measurements: The measurement was interrupted because the signaling connection to the DUT was lost.
- **23 ("Missing Option")**:
  An action cannot be executed due to a missing option.
  For GPRF generator: The ARB file cannot be played by the GPRF generator due to a missing option.
- **24 ("Invalid RF Setting")**:
  The desired RF TX level or RF RX reference level could not be applied.
- **25 ("Level Overrange")**:
  The RF TX level is in overrange. The signal quality can be degraded.
- **26 ("Resource Conflict")**:

The application could not be started or has been stopped due to a conflicting hardware resource or software option that is allocated by another application.
Stop the application that has allocated the conflicting resources and try again.

- **28 ("Unexpected Parameter Change")**:
One or more measurement configuration parameters were changed while the measurement completed. The results were not obtained with these new parameter values. Repeat the measurement. This situation can only occur in remote single-shot mode.
- **29 ("Invalid RF Frequency Setting")**:
The desired RF TX frequency or RF RX frequency could not be applied.
- **30 ("File Not Found")**:
The specified file could not be found.
- **40 ("ARB File CRC Error")**:
The cyclic redundancy check of the ARB file failed. The ARB file is corrupt and not reliable.
- **42 ("ARB Header Tag Invalid")**:
The ARB file selected in the GPRF generator contains an invalid header tag.
- **43 ("ARB Segment Overflow")**:
The number of segments in the multi-segment ARB file is higher than the allowed maximum.
- **44 ("ARB File Not Found")**:
The selected ARB file could not be found.
- **45 ("ARB Memory Overflow")**:
The ARB file length is greater than the available memory.
- **46 ("ARB Sample Rate Out of Range")**:
The clock rate of the ARB file is either too high or too low.
- **47 ("ARB Cycles Out of Range")**:
The repetition mode equals "Single Shot" and the playback length is greater than 40 s. Reduce the playback length or set the repetition mode to "Continuous".
*<Length> = (<Cycles> * <Samples> + <Additional Samples>) / <Clock Rate>*
- **52 ("Connection Error")**
A connection setup failed or a connection was lost.
- **70 ("Wrong Standard")**
The standard of the measured signal does not match the configured standard.
- **71 ("Wrong Bandwidth")**
The bandwidth of the measured signal does not match the configured bandwidth.
- **72 ("Wrong Burst Type")**
The burst type of the measured signal does not match the configured burst type.
- **73 ("MIMO Signal Detected")**
The measurement expects a SISO signal and detected a MIMO signal. Use a MIMO receive mode to measure this signal.
- **74 ("More Streams than Antennas")**
The measured signal has more streams than expected due to the configured number of antennas. Increase the configured number of antennas to measure this signal.
- **75 ("Matrix Inversion Failed")**

The inversion of the channel matrix failed for a MIMO measurement. Check that the antennas are connected correctly to the instrument.

- **76 ("SIG CRC Failed")**
  The cyclic redundancy check of a SIGNAL field failed.
- **77 ("Parity Check Failed")**
  The parity check of a SIGNAL field failed.
- **78 ("Bursts Not Identical")**
  In training mode for composite MIMO measurements, at least some symbols of sequential bursts need to be identical to be used as training data. Setting a fix scrambler initialization can solve this problem.
- **79 ("Wrong Modulation")**
  The modulation type of the measured signal does not match the configured modulation type.
- **93 ("OCXO Oven Temperature Too Low")**:
  The accuracy of measurement results can be impaired because the oven-controlled crystal oscillator has a too low temperature. After switching-on the instrument, the OCXO requires a warm-up phase to reach its operating temperature.
- **101 ("Firmware Error")**:
  Indicates a firmware or software error. If you encounter this error for the first time, restart the instrument.

  If the error occurs again, consider the following hints:

  – Firmware errors can often be repaired by restoring the factory default settings. To restore these settings, restart your instrument and press the "Factory Default" softkey during startup.
  – If a software package (update) has not been properly installed, this failure is often indicated in the "Setup" dialog, section "SW/HW-Equipment > Installed Software".
  – Check for software updates correcting the error. Updates are provided in the CMW customer web on GLORIS (registration required): https://gloris.rohde-schwarz.com.

  If you get firmware errors even with the properly installed latest software version, send a problem report including log files to Rohde & Schwarz.
- **102 ("Unidentified Error")**:
  Indicates an error not covered by other reliability values. For troubleshooting, follow the steps described for "101 (firmware error)".
- **103 ("Parameter Error")**:
  Indicates that the measurement could not be performed due to internal conflicting parameter settings.
  A good approach to localize the conflicting settings is to start with a reset or preset or even restore the factory default settings. Then reconfigure the measurement step by step and check when the error occurs for the first time.
  If you need assistance to localize the conflicting parameter settings, contact Rohde & Schwarz (see http://www.service.rohde-schwarz.com).
- **104 ("Not Functional")**:
  The application could not be started with the configured parameter set.

7.4.3.2 Error Indicators for Single Results

Some applications provide error indicators for single results, in addition to the reliability indicator. Single-value error indicators are retrieved using a CALCulate...? Command.

Example: `CALCulate:GPRF:MEAS<i>:POWer:CURRent?`

Response: 3, OK, OK, OK, OFL, OFL, OK, OK, OK (global reliability: 3 (for "over-driven"), eight power results in eight power steps, two results overdriven)

The error indicator has one of the following values.

Returned value	Meaning	Description
OK	OK	A measurement result is available, no error detected
ULEU	User limit violation upper	An upper limit is violated. The measurement result is above the limit.
ULEL	User limit violation lower	A lower limit is violated. The measurement result is below the limit.
OFL	Overflow	A measurement result is available. The accuracy of the result is not reliable because the input signal level was too high.
UFL	Underflow	A measurement result is available. The accuracy of the result is not reliable because the input signal level was too low.
INV	Invalid	The measurement was performed but a single result is invalid, e.g. because it cannot be measured under the special conditions of the measurement.
NAV	Not available	No measurement result is available, e.g. because the measurement could not be started.

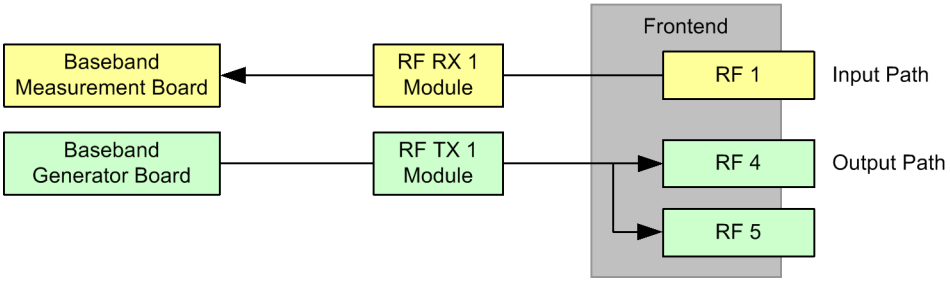
Some measurements provide only a subset of values.

Example: in the "GPRF Power" measurement, no user limits are defined and no underflow is indicated. The values ULEL, ULEU, and UFL are never returned.

7.4.4 Signal Path Settings

The signal path for input signals comprises an RF connector on the frontend, an RX module connected to the frontend and a baseband board.

Output signals use a reverse path with a TX module instead of an RX module. The signal can be routed to several output connectors.



To define a signal path via remote commands, the RF connectors and the RX/TX modules must be selected.

7.4.4.1 Values for RF Path Selection

This section lists all values available for RF path selection. Depending on how many radio test heads you have connected to your PC, only a subset is relevant for you.

There are physical and virtual connector names. Use virtual RF connector names if you want to send the same command sequence to different subinstruments. To query the mapping of physical and virtual connector names via remote command, see `SYSTem:CONNector:TRANslation?` on page 281.

Virtual connector names are only relevant for setting commands. Queries return the physical connector names.

Values for measurements

- RX connector:
  `R11 | R12 | R13 | R14 | R15 | R16 | R17 | R18`
  `R21 | R22 | R23 | R24 | R25 | R26 | R27 | R28`
  `R31 | R32 | R33 | R34 | R35 | R36 | R37 | R38`
  `R41 | R42 | R43 | R44 | R45 | R46 | R47 | R48`
  `RA1 | RA2 | RA3 | RA4 | RA5 | RA6 | RA7 | RA8`
  `RB1 | RB2 | RB3 | RB4 | RB5 | RB6 | RB7 | RB8`
  `RC1 | RC2 | RC3 | RC4 | RC5 | RC6 | RC7 | RC8`
  `RD1 | RD2 | RD3 | RD4 | RD5 | RD6 | RD7 | RD8`
  R<a><n> selects radio test head <a>, connector RF <n>.
  RA<n> is a virtual name for the first radio test head of the subinstrument and the <n>th connector of the subinstrument.
  RB<n> / RC<n> / RD<n> are virtual names for the second / third / fourth radio test head of the subinstrument, connector RF <n>
- RX module:
  `RX11 | RX21 | RX31 | RX41`
  RX<a>1 selects radio test head <a>, RX 1.
  Each radio test head is equipped with only one RX module.

Values for generators

You can route the signal to a single connector or to several connectors. The selection is done via two commands:

- The first command selects between the connector TX AUX and the connector bench RF 1 to RF 8.
- If you have selected the bench, a second command individually activates/deactivates the RF <n> connectors of the bench (connector usage).

RF path selection values:

- TX connector (bench):
  `R118 | R218 | R318 | R418 | RA18 | RB18 | RC18 | RD18`
  `R11O | R21O | R31O | R41O`
  R<a>18 selects radio test head <a>, connector RF 1 to RF 8 (connector bench).

RA18 / RB18 / RC18 / RD18 are virtual names for the first / second / third / fourth radio test head of the subinstrument, connector RF 1 to RF 8
R<a>1O selects radio test head <a>, connector TX AUX.

- Connector usage for a bench:
  ```
  R11 | R12 | R13 | R14 | R15 | R16 | R17 | R18
  R21 | R22 | R23 | R24 | R25 | R26 | R27 | R28
  R31 | R32 | R33 | R34 | R35 | R36 | R37 | R38
  R41 | R42 | R43 | R44 | R45 | R46 | R47 | R48
  RA1 | RA2 | RA3 | RA4 | RA5 | RA6 | RA7 | RA8
  RB1 | RB2 | RB3 | RB4 | RB5 | RB6 | RB7 | RB8
  RC1 | RC2 | RC3 | RC4 | RC5 | RC6 | RC7 | RC8
  RD1 | RD2 | RD3 | RD4 | RD5 | RD6 | RD7 | RD8
  ```
  R<a><n> selects radio test head <a>, connector RF <n>.
  RA<n> is a virtual name for the first radio test head of the subinstrument and the <n>th connector of the subinstrument.
  RB<n> / RC<n> / RD<n> are virtual names for the second / third / fourth radio test head of the subinstrument, connector RF <n>
- TX module:
  ```
  TX11 | TX21 | TX31 | TX41
  ```
  TX<a>1 selects radio test head <a>, TX 1.
  Each radio test head is equipped with only one TX module.

7.4.5 Resource and Path Management

The R&S CMW100 is a modular platform supporting a wide range of applications. For the remainder of this section, all these applications are termed "tasks".

In general, the instrument can run several tasks in parallel. For example, the GPRF generator can be used to generate a test signal, while the GPRF power measurement analyzes the input signal at different powers and frequencies. Conflicts between different tasks can occur if they rely upon the same system resources.

The resource and path management (RPM) of the R&S CMW100 represents a control mechanism for conflicting tasks, deciding whether and for how long a running task persists.

Remote and manual control

The RPM principles described in this section are valid in remote control and in manual control. In remote control, running different tasks in parallel enhances the speed and performance of the tester. In manual control, it allows you to monitor different measurements simultaneously and compare the results.

7.4.5.1 Basic RPM Principles

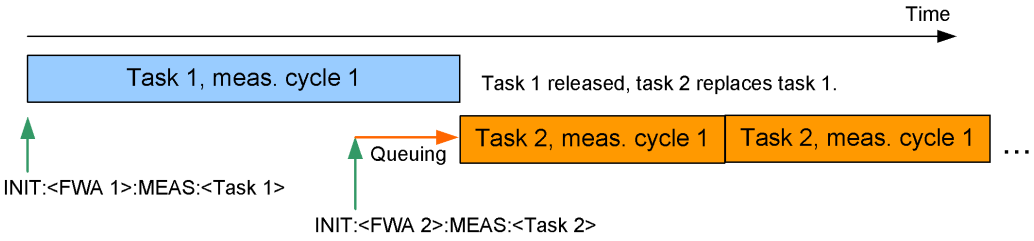
The principles of R&S CMW100 resource and path management can be summarized briefly:

- Non-conflicting tasks can run in parallel without restriction.

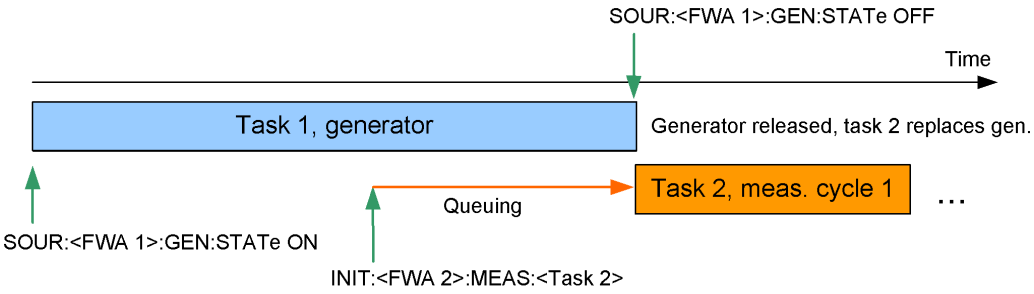
- A new task that is in conflict with a running task replaces the running task.
  Exceptions:
  – In smart channel mode, a new generator task that is in conflict with a running task is queued.
    See also Chapter 5.3, "Smart Channel Mode", on page 72.
  – Measurements can only replace measurements, but not tasks of another type. Instead, the new measurement is queued until the running task is ended by other means.
    See also Chapter 7.4.5.2, "Queuing of Measurements", on page 160.
- Measurements are released after the current measurement cycle. Other applications are released immediately.

The RPM principles can be visualized as follows:

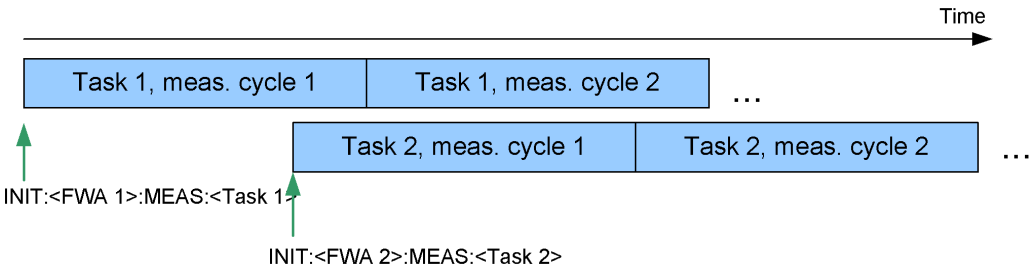
Two measurements in conflict, second measurement queued



Generator and measurement in conflict



Two non-conflicting measurements, executed in parallel



RPM acts on subinstrument level

Except in smart channel mode, each subinstrument is equipped with independent hardware and software resources to run the tasks assigned to it. So the RPM principles apply to each subinstrument separately.

This separation does not apply to software license keys. **Example:** To run two instances of a measurement in parallel, requires two license keys. If only one license key is active, the RPM mechanisms apply even if the two instances of the measurement are run on different subinstruments.

7.4.5.2 Queuing of Measurements

The R&S CMW100 queues conflicting measurements to ensure that no results are lost when a new measurement is started. Adding a measurement to the queue switches the preceding measurement into single-shot mode (if it is a continuous measurement). As a result, the instrument can acquire a valid set of measurement results for every measurement in the queue.

Example for queued conflicting measurements

Command	Comment
`*RST` `INITiate:<FWA>:<MeasA>`	Start measurement A in single-shot mode (default setting for remote control)
`INITiate:<FWA>:<MeasB>`	Start a new measurement B. The new measurement is queued.
`FETCh:<FWA>:<MeasA>:STATe:ALL?`	Query the measurement substates of measurement A. The response (`RUN, ADJ, ACT`) indicates that the measurement is running with all necessary adjustments finished, that the necessary resources are allocated and that results are being acquired.
`FETCh:<FWA>:<MeasB>:STATe:ALL?`	Query the measurement substates of measurement B. The response (`RUN, PEND, QUE`) indicates that the measurement is running but still queued, waiting for resource allocation.
`FETCh:<FWA>:<MeasA>...CURRent?` (or similar command syntax for reading measurement results)	Wait until the end of the statistics cycle of measurement A. Retrieve the results.
`FETCh:<FWA>:<MeasB>:STATe:ALL?`	Query the measurement substates of measurement B. The response (`RUN, ADJ, ACT`) indicates that measurement B is now active.

Extension: Queued continuous measurements

Command	Comment
`CONFigure:<FWA>:<MeasA>:REPetition CONT` `Initiate:<FWA>:<MeasA>`	Start measurement A in continuous mode
`CONFigure:<FWA>:<MeasB>:REPetition CONT` `Initiate:<FWA>:<MeasB>`	Start a new measurement B in continuous mode. The new measurement is queued. The old measurement is set to single-shot repetition mode.
`FETCh:<FWA>:<MeasA>:STATe:ALL?`	Query the measurement substates of measurement A. The response (`RUN, ADJ, ACT`) indicates that the measurement is running with all necessary adjustments finished, that the necessary resources are allocated and that results are being acquired.
`FETCh:<FWA>:<MeasB>:STATe:ALL?`	Query the measurement substates of measurement B. The response (`RUN, PEND, QUE`) indicates that the measurement is running but still queued, waiting for resource allocation.
`FETCh:<FWA>:<MeasA>...CURRent?` (or similar command syntax for reading measurement results)	Wait until the end of the statistics cycle of measurement A. Retrieve the results.
`FETCh:<FWA>:<MeasB>:STATe:ALL?`	Query the measurement substates of measurement B. The response (`RUN, ADJ, ACT`) indicates that measurement B is now active.
`CONFigure:<FWA>:<MeasA>: REPetition?` `CONFigure:<FWA>:<MeasA>: REPetition?`	Query the repetition mode of measurements A and B. Measurement A is in single-shot mode (`SING`), measurement B is still in continuous mode (`CONT`).

7.4.5.3 Causes for Task Conflicts

Resource conflicts can be categorized as follows.

Hardware resource conflicts

A generator requires an independent output path (digital and analog RF transmitter stages).

Mobile transmitter measurements (TX measurements) generally require their own input path (analog and digital RF receiver stages).

The number of tasks that the instrument can service in parallel depends on the number of installed independent RF paths.

System resource conflicts

In addition to the hardware resources required for TX and RX paths, other system resources cause limitations, for example the memory size and possible connections between modules.

Software license key conflicts

Many R&S CMW100 features must be enabled using a software license key. To be run in parallel, several measurements of the same type usually require several active software license keys. A resource conflict arises whenever the required number of software options is not available.

7.5 Status Reporting System

The status reporting system stores all information on the present operating state of the instrument, and on errors which have occurred. This information is stored in the status registers and in the error queue. Both can be queried via remote control commands, see Chapter 8.19, "Status Reporting System", on page 250.

Hierarchy of status registers

As shown in section Overview of Status Registers, the status information is of hierarchical structure.

- STB, SRE:
  The status byte (STB) register and its associated mask register service request enable (SRE) are at the highest level of the status reporting system. The STB provides a rough overview of the instrument status, collecting the information of the lower-level registers.
- The STB receives its information from:
  The event status register (ESR) with the associated mask register standard event status enable (ESE).
  The STATus:OPERation and STATus:QUEStionable registers which are defined by SCPI and contain detailed information on the instrument.
- IST, PPE:
  The IST flag ("individual status"), like the SRQ, combines the entire instrument status in a single bit. The PPE is associated to the IST flag. It fulfills an analogous function for the IST flag as the SRE does for the service request.
- Output buffer:
  Contains the messages the instrument returns to the controller. It is not part of the status reporting system but determines the value of the MAV bit in the STB.

All status registers have the same internal structure, see Structure of a SCPI Status Register.

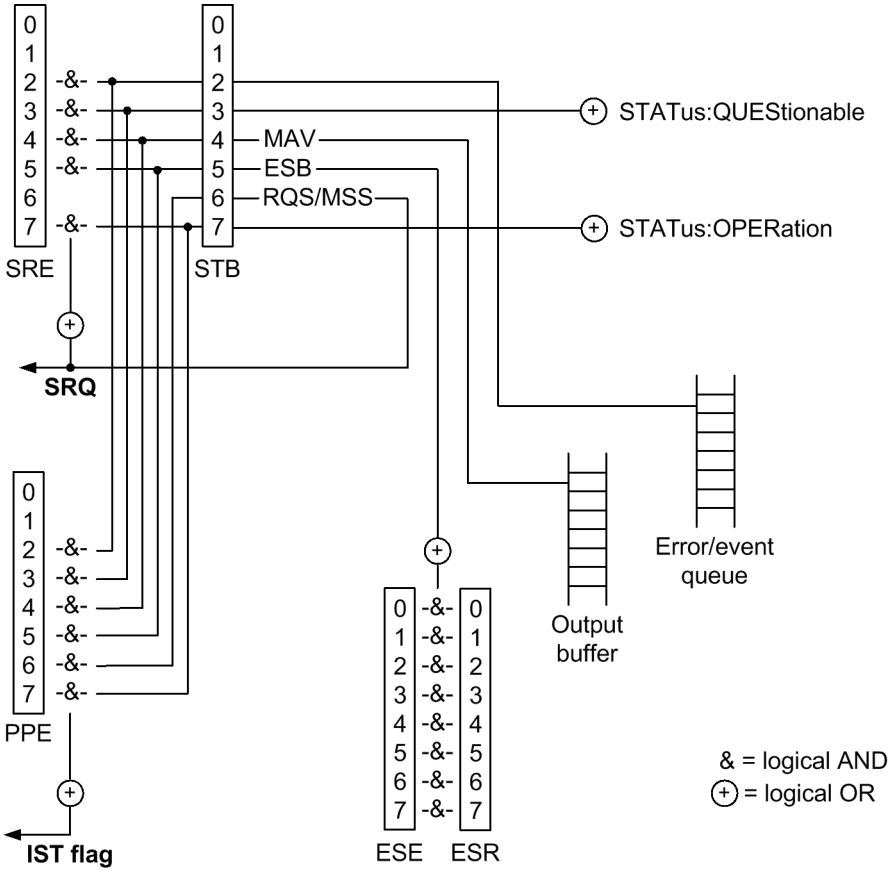
For more information on the individual status registers, see Contents of the Status Registers.

SRE register

The service request enable register SRE can be used as ENABle part of the STB if the STB is structured according to SCPI. By analogy, the ESE can be used as the ENABle part of the ESR.

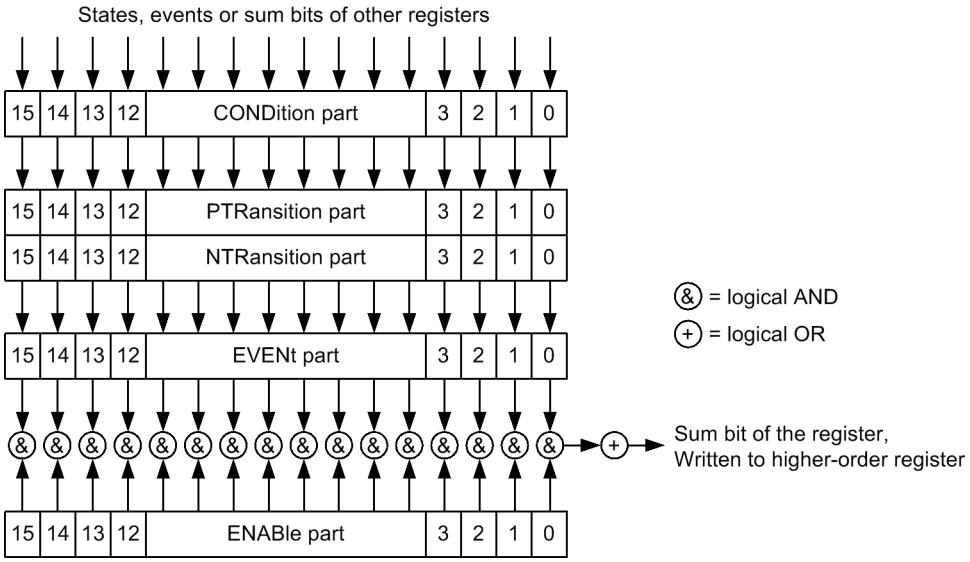
7.5.1 Overview of Status Registers

The status registers of the R&S CMW100 are implemented as shown below.



7.5.2 Structure of a SCPI Status Register

Each standard SCPI register consists of 5 parts which each have a width of 16 bits and have different functions. The individual bits are independent of each other, i.e. each hardware status is assigned a bit number which is valid for all five parts. Bit 15 (the most significant bit) is set to zero for all parts. Thus the contents of the register parts can be processed by the controller as positive integer.



The sum bit is obtained from the EVENt and ENABle part for each register. The result is then entered into a bit of the CONDition part of the higher-order register.

The instrument automatically generates the sum bit for each register. Thus an event can lead to a Service Request throughout all levels of the hierarchy.

The five parts of a SCPI register have different properties and function as described below.

CONDition

The CONDition part is permanently overwritten by the hardware or the sum bit of the next lower register. Its contents always reflect the current instrument state.

This register part can only be read, but not overwritten or cleared. Reading the CONDition register is nondestructive.

PTRansition

The two transition register parts define which state transition of the condition part (none, 0 to 1, 1 to 0 or both) is stored in the EVENt part.

The positive transition part acts as a transition filter. When a bit of the CONDition part is changed from 0 to 1, the associated PTR bit decides whether the EVENt bit is set to 1:

- PTR bit = 1: The EVENt bit is set.
- PTR bit = 0: The EVENt bit is not set.

This status register part can be overwritten and read at will. Reading the PTRansition register is nondestructive.

NTRansition

The negative transition part also acts as a transition filter. When a bit of the CONDition part is changed from 1 to 0, the associated NTR bit decides whether the EVENt bit is set to 1.

- NTR bit = 1: the EVENt bit is set.
- NTR bit = 0: the EVENt bit is not set.

This part can be overwritten and read at will. Reading the PTRansition register is non-destructive.

EVENt

The EVENt part indicates whether an event has occurred since the last reading. It is the "memory" of the condition part. It only indicates events passed on by the transition filters. It is permanently updated by the instrument. Reading the register clears it. This part is often equated with the entire register.

ENABle

The ENABle part determines whether the associated EVENt bit contributes to the sum bit (cf. below). Each bit of the EVENt part is ANDed with the associated ENABle bit (symbol '&'). The results of all logical operations of this part are passed on to the sum bit via an OR function (symbol '+').

- ENAB bit = 0: The associated EVENt bit does not contribute to the sum bit.
- ENAB bit = 1: If the associated EVENT bit is "1", the sum bit is set to "1" as well.

You can overwrite and read this part. Its contents are not affected by reading.

7.5.3 Contents of the Status Registers

The individual status registers are used to report different classes of instrument states or errors. The following status registers belong to the general model described in IEEE 488.2:

- The status byte (STB) gives a rough overview of the instrument status.
- The IST flag combines the entire status information into a single bit that can be queried in a Parallel Poll.
- The event status register (ESR) indicates general instrument states.

The status registers below belong to the device-dependent SCPI register model:

- The STATus:OPERation register contains conditions which are part of the instrument's normal operation.
- The STATus:QUEStionable register indicates whether the data currently being acquired is of questionable quality.

7.5.3.1 STB and SRE

The status byte (STB) provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. The STB represents the highest level within the SCPI hierarchy. A special feature is that bit six acts as the summary bit of the remaining bits of the status byte.

The status byte (STB) is linked to the service request enable (SRE) register on a bit-by-bit basis.

- The STB corresponds to the EVENt part of a SCPI register, it indicates general instrument events. This register is cleared when it is read.
- The SRE corresponds to the ENABle part of a SCPI register. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a Service Request (SRQ) is generated.

Bit 6 of the SRE is ignored, because it corresponds to the summary bit of the STB.

The bits in the STB are defined as follows:

Bit No.	Meaning
2	**Error queue not empty** This bit is set when an entry is made in the error queue.
3	**QUEStionable status summary bit** The bit indicates a questionable instrument status, which can be further pinned down by polling the QUEStionable register.
4	**MAV bit** (message available) This bit is set if a message is available and can be read from the output buffer. This bit can be used to transfer data automatically from the instrument to the controller.
5	**ESB bit** Sum bit of the event status register. It is set if one of the bits in the event status register is set and enabled in the event status enable register. Setting of this bit implies an error or an event which can be further pinned down by polling the event status register.
6	**MSS bit** (master status summary bit) This bit is set if the instrument triggers a service request. That means, if one of the other bits of the STB is set together with its mask bit in the SRE register.
7	**OPERation status register summary bit** This bit is set if an EVENt bit is set in the OPERation status register and the associated ENABle bit is set to 1.

Related common commands

The STB is read out using the command `*STB?` / `*XSTB?` or a Serial Poll.

The SRE can be set using command `*SRE` / `*XSRE` and read using `*SRE?` / `*XSRE?`.

7.5.3.2 IST Flag and PPE

In analogy to the Service Request (SRQ), the IST flag combines the entire status information in a single bit. It can be queried via a Parallel Poll.

The parallel poll enable (PPE) register determines which bits of the STB contribute to the IST flag. The bits of the STB are combined via AND with the corresponding bits of the PPE, with bit 6 being used as well in contrast to the SRE. The resulting bits are combined via OR to determine the IST flag.

Related common commands

The IST flag is queried using the command `*IST?`.

The PPE can be set using `*PRE` / `*XPRE` and read using command `*PRE?` / `*XPRE?`.

See also Common Commands

7.5.3.3 ESR and ESE

The event status register (ESR) indicates general instrument states. It is linked to the event status enable (ESE) register on a bit-by-bit basis.

- The ESR corresponds to the CONDition part of a SCPI register indicating the current instrument state (although reading is destructive).
- The ESE corresponds to the ENABle part of a SCPI register. If a bit is set in the ESE and the associated bit in the ESR changes from 0 to 1, the ESB bit in the status byte is set.

The bits in the ESR are defined as follows:

Bit No.	Meaning
0	**Operation complete** This bit is set on receipt of the command *OPC after all previous commands have been executed.
1	**Request control** This bit is set if the instrument requests the controller function. Example: The instrument sends a hardcopy to a printer or a plotter via the IEC-bus.
2	**Query error** This bit is set if the controller wants to read data from the instrument without having sent a query. It is also set if the controller does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query which is faulty and hence cannot be executed.
3	**Device-dependent error** This bit is set if a device-dependent error occurs. An error message with a number between -300 and -399 or a positive error number, which describes the error in greater detail, is entered into the error queue.
4	**Execution error** This bit is set if a received command is syntactically correct, but cannot be performed for other reasons. An error message with a number between -200 and -300, which describes the error in greater detail, is entered into the error queue.
5	**Command error** This bit is set if a command which is undefined or syntactically incorrect is received. An error message with a number between -100 and -200, which describes the error in greater detail, is entered into the error queue.

Bit No.	Meaning
6	**User request** This bit is set on pressing the *LOCAL* key, i. e. when the instrument is switched over to manual control.
7	**Power on** (supply voltage on) This bit is set when the instrument is switched on.

Related common commands

The event status register (ESR) can be queried using `*ESR?` / `*XESR?`.

The event status enable (ESE) register can be set using the command `*ESE` / `*XESE` and read using `*ESE?` / `*XESE?`.

See also Common Commands

7.5.3.4 STATus:OPERation

The STATus:OPERation register provides an overview of state transitions of the tasks (e.g. GPRF:MEASurement1:POWer) by collecting the information of lower registers. The STATus:OPERation register hierarchy is shown below. The paths can also be queried via remote command, see `SYSTem:HELP:STATus[:REGister]?` on page 253 and `SYSTem:HELP:STATus:BITS?` on page 253.

The figure and the remainder of this section describe the complete hierarchy supported by the software. Depending on the instrument model, not all applications are relevant.

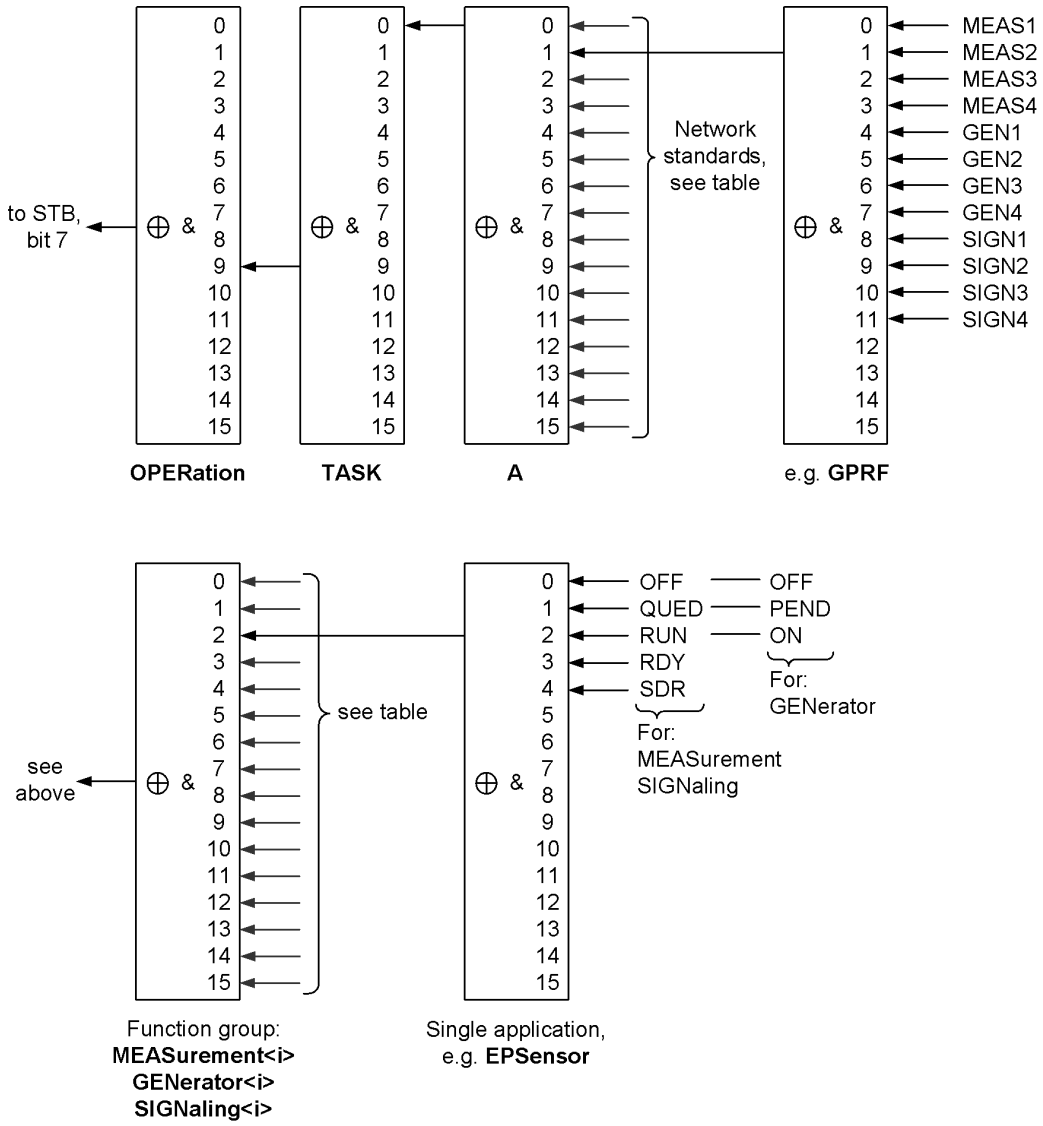


Figure 7-1: STATus:OPERation register hierarchy

Each single status register in the figure above consists of five parts as described in section Structure of a SCPI Status Register. The following simplified presentation is used:

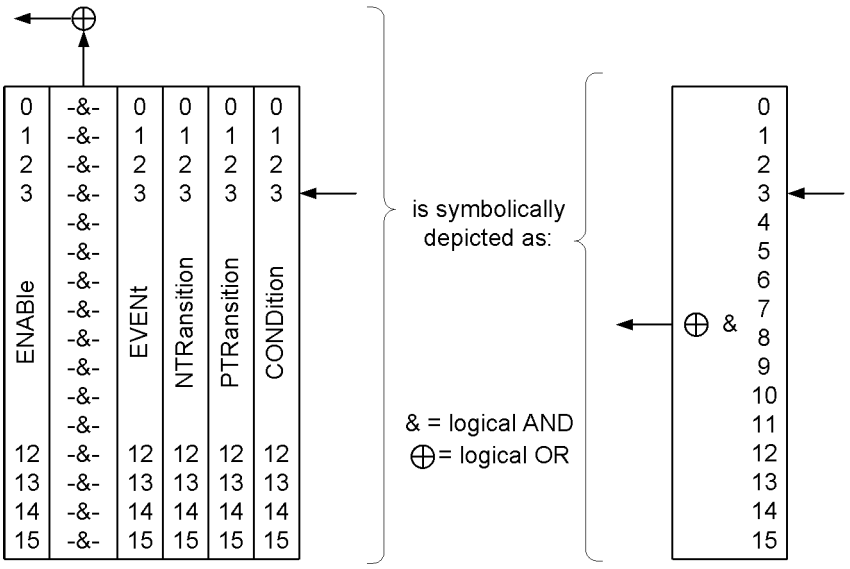


Figure 7-2: Legend for figure above

The following table lists the bits of the register STATus:OPERation:TASK:A. The network standard mnemonics are identical to the mnemonics used in remote commands. Depending on the instrument model and the installed applications, only a subset is relevant for you.

Table 7-2: Bit definition for register STATus:OPERation:TASK:A

Bit No.	Network standard
1	GPRF
2	GSM
3	WCDMa
4	WLAN
5	BLUetooth
7	EVDO
8	CDMA
9	TDSCdma
10	LTE

The following table lists the assignment of the individual applications to the bits of a function group register (STATus:OPERation:TASK:A:<network standard>:MEASurement<i> / ...:SIGNaling<i> / ...:GENerator<i>).

The mnemonics are as far as possible identical to the mnemonics used in remote commands. UNIVersal is used if the application is not explicitly identified in remote commands (e.g. STATus:OPERation:TASK:A:GPRF:GENerator1:UNIVersal).

Depending on the instrument model and the installed applications, only a subset of the mnemonics is relevant for you.

Table 7-3: Bit definition for function group registers

Bit no.	Measurement	Signaling	Generator
0	POWer, MEValuation, PING, ANALog	BERCswitched, BER, PER, TDATa, EBLer	UNIVersal
1	IQVSlot, PRACh, IPERf, TPC, OLTR, DIGital	THRoughput, BERPswitched, HACK	-
2	EPSensor, SRS, THRoughput	BLER, EHICh	-
3	IQRecorder	THRoughput	-
4	FFTSanalyzer	ULLogging	ANALog
5	SPECtrum	-	DIGital

The status registers at the lowest level indicate state transitions to the following states:

Table 7-4: Bit definition for application register

Bit no.	Measurement / signaling state	Generator state
0	**OFF**	**OFF**
1	**QUED** (queued)	**PEND** (pending)
2	**RUN** (running)	**ON**
3	**RDY** (ready)	not used
4	**SDR** (statistical depth reached)	not used

The reporting structures of the register hierarchy are administrable using the following commands:

- `STATus:OPERation:TASK:A:ENABle` etc.
- `STATus:OPERation:TASK:A:ESRQ` etc.
- `STATus:OPERation:TASK:A:NTRansition` etc.
- `STATus:OPERation:TASK:A:PTRansition` etc.

The states and state transitions of a specific task can be queried using the following commands:

- `STATus:OPERation:TASK:A:GPRF:GENerator1:UNIVersal:CONDition?` etc.
- `STATus:OPERation:TASK:A:GPRF:GENerator2:UNIVersal:EVENt?` etc.

Waiting until a certain state is reached or a certain state transition occurs, is possible using the following commands:

- `STATus:OPERation:TASK:A:GPRF:GENerator1:UNIVersal:WCONdition?` etc.
- `STATus:OPERation:TASK:A:GPRF:GENerator2:UNIVersal:WEVent?` etc.

These commands are also available as extended versions, providing more comfort:

- `...:XENable`, `...:XESRq`, `...:XNTRansition`, `...:XPTRansition`
- `...:XCONdition`, `...:XEVent`

- `...:XWCondition`, `...:XWEVent`

More sophisticated evaluations of the hierarchy are possible using e.g.:

- `STATus:CONDition:BITS:ALL?`
- `STATus:EVENt:BITS:ALL?`
- `STATus:MEASurement:CONDition:RDY?` etc.

For command descriptions refer to:

- Chapter 8.19.2, "STATus:OPERation (Elementary Commands)", on page 251
- Chapter 8.19.3, "STATus:OPERation (Extended Commands)", on page 259
- Chapter 8.19.4, "STATus:OPERation (Overall Evaluation)", on page 266

7.5.3.5 STATus:QUEStionable

The STATus:QUEStionable register indicates whether the data currently being acquired is of questionable quality.

The R&S CMW100 does not use the STATus:QUEStionable register.

7.5.4 Application of the Status Reporting System

The purpose of the status reporting system is to monitor the status of one or several devices in a measuring system. The controller must receive and evaluate the information of all devices. The following standard methods described in the following sections are used:

- Service request (SRQ) initiated by the measuring device
- Serial poll of all devices in the bus system, initiated by the controller to find out who sent an SRQ and why
- Parallel poll of all devices
- Query of a specific instrument status via commands
- Query of the error queue

7.5.4.1 Service Request

The measuring device can send a service request (SRQ) to the controller. Usually this service request causes an interrupt, to which the control program can react appropriately. It is also possible to wait until an SRQ is generated, see `*SRQ?` in Chapter 8.2, "Common Commands", on page 184).

Initiating an SRQ

As shown in Chapter 7.5.1, "Overview of Status Registers", on page 163, an SRQ is initiated if one or several of bits 2, 3, 4, 5 or 7 of the status byte are set and enabled in the SRE. Each of these bits summarizes the information of a further register, the error queue or the output buffer.

The ENABle parts of the status registers can be set such that arbitrary bits in an arbitrary status register initiate an SRQ. To use the possibilities of the service request effectively, all bits in the enable registers SRE and ESE must be set to "1".

Example: Use *OPC to generate an SRQ

1. Set bit 0 in the ESE (operation complete).
2. Set bit 5 in the SRE (ESB).
3. Insert `*OPC` in the command sequence.

When all commands preceding `*OPC` have been completed, the instrument generates an SRQ.

Example: Indicate measurement state via SRQ

In this example, the reporting path is enabled for a specific measurement state, so that an SRQ is generated when this state is reached. Then the instrument is commanded to wait until an SRQ is generated.

1. Enable reporting for the state RDY of the GPRF power measurement:
   `STATus:OPERation:TASK:A:GPRF:MEASurement:POWer:XESRq (RDY)`
2. Start the GPRF measurement:
   `INITiate:GPRF:MEASurement:POWer`
3. Wait until the measurement reaches the state RDY. If there is still no SRQ after 1 minute (60000 ms), continue nevertheless:
   `*SRQ? 60000`
   Evaluate the returned value: 0 indicates that a timeout occurred. 1 indicates an SRQ.

The following steps show an alternative way to achieve the same result without SRQ generation.

1. Start the GPRF measurement:
   `INITiate:GPRF:MEASurement:POWer`
2. Wait until the measurement reaches the state RDY, with a timeout of 1 minute (60000 ms):
   `STAT:OPER:TASK:A:GPRF:MEAS:POW:XWEVent? (RDY),60000`
   Evaluate the returned value: () indicates that a timeout occurred. (RDY) indicates the reached RDY state.

The SRQ is the only possibility for the instrument to become active on its own. Each controller program must set the instrument such that a service request is initiated in the case of malfunction. The program must react appropriately to the service request.

7.5.4.2 Serial Poll

In a serial poll, the controller queries the status bytes of the devices in the bus system one after another. The query is made via interface messages, so it is faster than a poll via `*STB?`.

The serial poll method is defined in IEEE 488.1 and used to be the only standard possibility for different instruments to poll the status byte. The method also works for instruments which do not adhere to SCPI or IEEE 488.2.

The serial poll is used to obtain a fast overview of the state of several instruments connected to the controller.

7.5.4.3 Parallel Poll

In a parallel poll, the controller requests up to eight instruments to set the data line allocated to each instrument to a logical "0" or "1".

In addition to the SRE register, which determines the conditions under which an SRQ is generated, there is a parallel poll enable register (PPE). This register is combined via AND with the STB bit by bit, considering bit 6 as well. The results are combined via OR. The result is possibly inverted and then sent as a response to the parallel poll of the controller. The result can also be queried without a parallel poll via the command `*IST?`.

The parallel poll method is used to find out quickly which one of the instruments connected to the controller has sent a service request. To this effect, SRE and PPE must be set to the same value.

7.5.4.4 Query of an Instrument Status

Each part of any status register can be read by queries. There are two types of commands:

- Common commands query the higher-level registers. Examples are `*ESR?`, `*ESE?`, `*IST?` and `*STB?`, see also Chapter 8.2, "Common Commands", on page 184.
- The commands of the STATus system query the SCPI registers (e.g. `STATus:OPERation...`).

All queries return a decimal number which represents the bit pattern of the status register. This number is evaluated by the controller program.

Queries are used after an SRQ to obtain more detailed information on the cause of the SRQ.

Decimal representation of a bit pattern

The STB and ESR registers contain 8 bits, the SCPI registers 16 bits. The contents of a status register are keyed and transferred as a single decimal number. Each bit is assigned a weighted value. The decimal number is calculated as the sum of the weighted values of all bits in the register that are set to 1.

Bits	0	1	2	3	4	5	6	7	...
Weight	1	2	4	8	16	32	64	128	...

Example: The decimal value 40 = 32 + 8 indicates that bits no. 3 and 5 in the status register are set.

7.5.4.5 Error Queue

Each error state in the instrument leads to an entry in the error queue. The entries of the error queue are detailed plain text error messages that can be queried via remote control using `SYSTem:ERRor[:NEXT]?` or `SYSTem:ERRor:ALL?`. Each call of `SYSTem:ERRor[:NEXT]?` returns one error queue entry. If no error messages are stored there any more, the instrument responds with `0, "No error"`.

Query the error queue after every SRQ . The entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program query the error queue regularly, since faulty commands from the controller to the instrument are recorded there as well.

7.5.5 Reset Values of the Status Reporting System

The table below indicates the effects of various commands upon the status reporting system of the R&S CMW100.

Event	**Switching on supply voltage power-on-status-clear**		**DCL, SDC (device clear, selected device clear)**	***RST or SYS-Tem:PRE-Set:ALL**	**STA-Tus:PRE-Set**	***CLS**
Effect	0	1				
Clear STB, ESR		yes				yes
Clear SRE, ESE		yes				
Clear PPE		yes				
Clear EVENt parts of the registers		yes				yes
Clear ENABle part of all OPERation registers and QUESTionable registers Fill ENABle parts of all other registers with "1"		yes			yes	
Fill PTRansition parts with "1" Clear NTRansition parts		yes			yes	
Clear error queue	yes	yes				yes
Clear output buffer	yes	yes	yes	1)	1)	1)

Event	Switching on supply voltage power-on-status-clear		DCL, SDC (device clear, selected device clear)	*RST or SYS-Tem:PRE-Set:ALL	STA-Tus:PRE-Set	*CLS
Clear command processing and input buffer	yes	yes	yes			
1) Every command being the first in a command line, i.e. immediately following a <PROGRAM MESSAGE TERMINATOR> clears the output buffer.						

7.5.6 Regular Expressions

Some remote control commands (for examples see STATus:OPERation (Overall Evaluation)) support the application of filters. A filter allows you to limit the results returned by the command (or the objects affected by the command) to a specific set of strings. This set can be defined via a regular expression without having to list all elements.

The most basic filter expression consists of a simple string containing no meta characters, e.g. 'CDM'. Strings match to this filter if they contain the defined filter string. Matching examples: 'CDMA2000', 'WCDMA', 'WCDMA network standard'.

More sophisticated filter expressions can be defined using the meta characters listed in the following table. A string matches to a filter example given in the table if it contains any of the given matching strings.

Meta character	Meaning
.	Matches any single character.
[]	Matches a single character given within the brackets. Example: '[ABC]' matches 'A', 'B' and 'C'. [-] specifies a range. Example: '[ABCX-Z]' matches 'A', 'B', 'C', 'X', 'Y' and 'Z', as does '[A-CX-Z]'.
[^]	Matches a single character not given within the brackets. Example: '[^ABC]' matches any character other than 'A', 'B' and 'C'.
^	If given at the beginning of a filter, matches the beginning of the string. Example: '^[ABC]' matches any string starting with 'A', 'B' or 'C'.
$	If given at the end of a filter, matches the end of the string. Example: '[ABC]$' matches any string ending with 'A', 'B' or 'C'.
?	Matches the preceding element zero or one time. Example: 'M[0-9][0-9]?' matches 'M1', 'M25', 'M0', 'M00', ...
+	Matches the preceding element one or more times. Example: 'TR[AB]+' matches 'TRA', 'TRB', 'TRABBBABA', ...
*	Matches the preceding element zero or more times. Example: 'A*B' matches 'B', 'AB', 'AAAAAAB', ...
()	Groups the content for other operators. Example: 'A(BCD)?E' matches 'AE' and 'ABCDE'.
\	Interpret the next character literally, not as meta character. Example: '[A\-C]' matches 'A', '-' and 'C', while '[A-C]' matches 'A', 'B' and 'C'.

Meta character	Meaning
|	Alternation operator, matching the expression before or after the operator. Example: 'NOT|R' matches 'NOT' and 'NOR'.
!	Negation operator. The expression following the exclamation mark does not match. Example: '1!01' matches '111', '121' etc. but not '101'.

Character sets can be represented by abbreviations indicated in the following table.

Abbreviation	Matches
\a	Any alphanumeric character ([a-zA-Z0-9])
\b	White space (blank)
\c	Any alphabetic character ([a-zA-Z])
\d	Any decimal digit ([0-9])
\h	Any hexadecimal digit ([a-fA-F0-9])
\w	A simple word ([a-zA-Z]+)
\z	An integer ([0-9]+)

7.6 Command Macros

A macro is a sequence of remote commands which can be referenced in a remote control program. Macros are kept in the RAM while a remote connection is active. Therefore, they are a means of saving transfer time and speed up the measurement. The gain is most noticeable if the macros contain many commands and if they are executed repeatedly. Macros are, for example, suited for the configuration sections of a program, e.g. to combine a group of settings which is repeated in a loop.

Macros must be defined at the beginning of a remote script. In general, they remain valid until the active remote control connection is closed. Macros are not directly transferable from one remote channel to another. It is possible though to save macros to files and to reuse them in other remote connections or sessions. It is also possible to use different macros with the same label for different connections.

Queries in macros, response buffers

A query in a macro sequence suspends program execution until the controller has retrieved the response. The R&S CMW100 provides response buffers to avoid delays; see Chapter 7.7, "Response Buffers", on page 180.

7.6.1 Macro Contents and Macro Commands

Macros can be defined in the RAM using the IEEE 488.2 common command `*DMC`.

It supports two formats for macro contents:

- Contents can be defined in IEEE 488.2 block data format with defined or indefinite length. "#0" introduces a data block of indefinite length. `*GMC?` returns the macro contents in block data format.
  See also "Block Data Format" on page 139.
- As an alternative, `*DMC` accepts strings with a maximum length of 1023 characters.

Macros present in the RAM can be saved to macro files using `MMEMory:STORe:MACRo`. Macro files can be loaded into the RAM using `MMEMory:LOAD:MACRo`.

Macro files can also be recorded. Start macro recording via `SYSTem:RECord:MACRo:FILE:STARt`. Then submit the commands to be stored into the macro file and stop recording via `SYSTem:RECord:MACRo:FILE:STOP`. Ensure that only one remote control interface of the instrument is used during recording of a macro file.

Macro files are identified via their path and filename on the instrument. Macros within the RAM are identified via a label.

Memory size

The block data size of any single macro (and the size of a macro file) must not exceed 1 MB. The R&S CMW100 can process macros with a combined size of up to 16 MB.

7.6.2 Macro Programming Examples

The following examples show you how to use macros, macro parameters, and macro files. For a command reference, see Chapter 8.3, "Macro Commands", on page 192.

Macro handling using common control commands

```
// ******************************************************************************
// Reset the instrument.
// Define a macro to set the GPRF generator frequency to 1 GHz.
// Query the contents of the macro (should be
// #247SOURce:GPRF:GENerator:RFSettings:FREQuency 1GHz).
// Query labels of existing macros.
// ******************************************************************************
*RST
*DMC 'SetFrequency', 'SOURce:GPRF:GENerator:RFSettings:FREQuency 1 GHz'
*GMC? 'SetFrequency'
*LMC?

// ******************************************************************************
// Query the GPRF generator frequency.
// Enable macro execution.
// Execute the macro.
// Check whether the generator frequency has actually been changed.
// ******************************************************************************
SOURce:GPRF:GENerator:RFSettings:FREQuency?
*EMC ON
```

```
SetFrequency
SOURce:GPRF:GENerator:RFSettings:FREQuency?

// ******************************************************************************
// Delete the macro.
// Delete all macros in the active connection.
// Check whether all macros have been deleted (response: "").
// ******************************************************************************
*RMC 'SetFrequency'
*PMC
*LMC?
```

Macros with parameters

```
// ******************************************************************************
// Reset the instrument.
// Define a macro to set the GPRF generator frequency and level to arbitrary
// values using macro parameters.
// Execute the macro and check the results.
// ******************************************************************************
*RST
*DMC 'RF_Settings', 'SOURce:GPRF:GENerator:RFSettings:FREQuency $1; LEVel $2'
*EMC ON
SOURce:GPRF:GENerator:RFSettings:FREQuency?; LEVel?
RF_Settings 1 GHz,-10 dBm
SOURce:GPRF:GENerator:RFSettings:FREQuency?; LEVel?
```

Using macro files and block data format

```
// ******************************************************************************
// Store the macro to a file.
// Delete the macro, check whether all macros have been deleted (response: "").
// ******************************************************************************
MMEMory:STORe:MACRo 'RF_Settings', '@USERDATA/macros/RF_Settings.txt'
*PMC
*LMC?

// ******************************************************************************
// Re-load the macro contents from the macro file, assigning the old label.
// Query the contents.
// ******************************************************************************
MMEMory:LOAD:MACRo 'RF_Settings', '@USERDATA/macros/RF_Settings.txt'
*GMC? 'RF_Settings'
*LMC?

// ******************************************************************************
// Re-define the macro using block data format. Check and store the macro.
// ******************************************************************************
*DMC 'RF_Settings', #0SOURce:GPRF:GENerator:RFSettings:FREQuency $1; LEVel $2
*GMC? 'RF_Settings'
MMEMory:STORe:MACRo 'RF_Settings', '@USERDATA/macros/RF_Settings.txt'
```

Recording macro files

```
// ******************************************************************************
// Start recording.
// Submit commands to be recorded.
// Stop recording.
// Load the recorded macro and check its contents.
// ******************************************************************************
SYSTem:RECord:MACRo:FILE:STARt '@USERDATA/macros/GPRF.txt'
SOURce:GPRF:GENerator:RFSettings:FREQuency 1 GHz
SOURce:GPRF:GENerator:RFSettings:EATTenuation 2
SOURce:GPRF:GENerator:RFSettings:LEVel -70
SYSTem:RECord:MACRo:FILE:STOP
MMEMory:LOAD:MACRo 'RF_Settings1', '@USERDATA/macros/GPRF.txt'
*GMC? 'RF_Settings1'
```

7.7 Response Buffers

A response buffer is a region of memory in the RAM, used to store device responses during queries. If no buffer is enabled, a query in a remote script suspends program execution until the controller has retrieved the response. The main purpose of response buffers is to avoid these delays. Results are stored while program execution continues and can be read at the end of the program.

Response buffers improve the efficiency of command macros containing queries. Similar to command macros, the buffers are created at the beginning of a remote script. Buffers remain valid until the active remote control connection is closed. They are not transferable from one remote channel to another. It is possible to assign buffers with the same name (but different contents) to different connections.

DCL, troubleshooting

A device clear (DCL) interface message does not deactivate the buffer. Errors in the response buffers can be monitored using the status reporting system and the error queue (`SYSTem:ERRor...?`).

7.7.1 Buffer Contents and Buffer Commands

Response buffers are defined and activated using a `STARt:BASE:BUFFer '<BufferLabel>'` command. The buffer label serves as a reference in all other buffer commands. It is possible to define several buffers. But only one buffer can be active at any time.

When a response buffer is active, the complete contents of the R&S CMW100's output buffer are copied to the active buffer instead of being transferred over the remote interface. Every program line in a command script containing queries generates a single new buffer line. The queries are not stored together with the results.

Buffers can be temporarily deactivated, e.g. to exclude a response to `SYSTem:ERRor?` from the buffer. It is also possible to clear the buffer contents or delete the buffer during program execution. The buffer contents can be read line by line. The syntax of buffer commands follows general R&S CMW100 syntax rules; see Chapter 8.4, "Buffer Commands", on page 196.

Memory size

The buffer size must not exceed 16 MB. When the maximum buffer size is reached, a "Buffer Deadlock" error message is created and no additional data is appended to the buffer. The previously stored data is retained.

7.7.2 Buffer Programming Example

The following example shows you how to work with response buffers. For a command reference, see Chapter 8.4, "Buffer Commands", on page 196.

```
// ******************************************************************************
// Define a macro containing a query and create a buffer to record the responses
// ******************************************************************************
*RST; *CLS
*DMC 'Query_macro', 'SOURce:GPRF:GENerator:RFSettings:FREQuency $1; *WAI;
   :SOURce:GPRF:GENerator:RFSettings:FREQuency?'
STARt:BASE:BUFFer 'Frequency_Buffer'

// ******************************************************************************
// Execute the macro repeatedly, stop recording in order to query the number of
// buffer lines (3) and the buffer line contents
// (the responses should be 3 GHz, 2 GHz, 1GHz)
// ******************************************************************************
Query_macro 1 GHz
Query_macro 2 GHz
Query_macro 3 GHz
STOP:BASE:BUFFer
FETCh:BASE:BUFFer:LINEcount? 'Frequency_Buffer'
FETCh:BASE:BUFFer? 'Frequency_Buffer',3
FETCh:BASE:BUFFer? 'Frequency_Buffer',2
FETCh:BASE:BUFFer? 'Frequency_Buffer',1

// ******************************************************************************
// With stopped buffer, no further lines are appended.
// The line count query still returns 3.
// ******************************************************************************
Query_macro 1 GHz
FETCh:BASE:BUFFer:LINEcount? 'Frequency_Buffer'

// ******************************************************************************
// Re-activate the buffer to appended further lines.
// The line count query returns 4.
```

```
// ******************************************************************************
CONTinue:BASE:BUFFer 'Frequency_Buffer'
Query_macro 4 GHz
STOP:BASE:BUFFer
FETCh:BASE:BUFFer:LINEcount? 'Frequency_Buffer'

// ******************************************************************************
// Clear the buffer contents: the line count query returns 0
// Delete the buffer: the line count query times out and the error queue
// contains an entry with code -273 "Illegal macro label"
// ******************************************************************************
CLEAR:BASE:BUFFer 'Frequency_Buffer'
FETCh:BASE:BUFFer:LINEcount? 'Frequency_Buffer'
DELete:BASE:BUFFer 'Frequency_Buffer'
FETCh:BASE:BUFFer:LINEcount? 'Frequency_Buffer'
SYSTem:ERRor?
```

8 Command Reference

This chapter lists all commands for the R&S CMW100 platform.

Application-specific commands (for example GPRF commands) are not listed here. Instead, refer to the documentation of the application.

8.1 Keywords

Selected keywords used in the command description are described in the following.

- **Command usage**
  If the usage is not explicitly stated, the command allows you to set parameters and query parameters. Otherwise, the command usage is stated as follows:
  - "Setting only": The command can only be used to set parameters.
  - "Query only": The command can only be used to query parameters.
  - "Event": The command initiates an event.
- **Parameter usage**
  The parameter usage is indicated by the keyword preceding the parameters:
  - "Parameters" are sent with a setting or query command and are returned as the result of a query.
  - "Setting parameters" are sent with a setting command.
  - "Query parameters" are sent with a query command, to refine the query.

– "Return values" are returned as the result of a query.

- **Firmware/Software:**
  Indicates the lowest software version supporting the command. Command enhancements in later software versions are also indicated.

8.2 Common Commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect on different instruments. The command headers consist of an asterisk "*" followed by three letters.

Many common commands are related to the status reporting system, see Chapter 7.5, "Status Reporting System", on page 162.

For commands related to macros, see Chapter 8.3, "Macro Commands", on page 192.

This section describes the supported common commands and additional R&S CMW100-specific commands that use similar syntax.

***CLS**

Clears the status byte (STB) register, the event status read (ESR) register and the `EVENt` part of the `QUEStionable` and the `OPERation` registers.

The command does not alter the mask and transition parts of the registers. It clears the output buffer.

Usage: Event

Firmware/Software: V1.0.0.4

***DEV?**

Queries the device number. It equals the "Assigned Instrument" number minus 1.

Return values:
<Value> Range: 0 to n

Example: `*DEV?`
Return value `0`, for example, means "Assigned Instrument" = 1.

Usage: Query only

Firmware/Software: V1.0.0.4

***ESE** <Value>

Sets the event status enable (ESE) register to the specified value.

Alternative command: `*XESE` on page 189

Parameters:
<Value> Range: 0 to 255

Example: `*ESE 5`
Sets bit 2 and bit 0 to 1. Sets the other bits to 0.

Firmware/Software: V1.0.0.4

***ESR?**

Returns the contents of the event status register (ESR) in decimal form and sets the register to zero.

Alternative command: `*XESR?` on page 190

Return values:
<Value> Range: 0 to 255

Example: `*ESR?`
Return value `5` for example means: bit 2 and bit 0 = 1, all other bits = 0.

Usage: Query only

Firmware/Software: V1.0.0.4

***IDN?**

Returns the instrument identification.

If you have connected several radio test heads and want to check the serial numbers, the following commands are more appropriate:

- `SYSTem:CMW:DEVice:ID?` on page 228
- `FETCh:BASE:MCMW:SNUMber?` on page 246

Return values:

<ID> Instrument identification as string: "Rohde&Schwarz,<instrument type>,<serial number>,<base software version>"

Usage: Query only

Firmware/Software: V1.0.0.4

***IST?**

Returns the IST flag in decimal form. The IST flag is the status bit which is sent during a parallel poll.

Return values:

<ISTflag> 0 | 1

Usage: Query only

Firmware/Software: V1.0.0.4

***OPC**

`*OPC` sets bit 0 in the event status register (ESR) when all preceding commands have been executed. This bit can be used to initiate a service request.

`*OPC?` waits until all preceding commands have been executed. Then it returns a "1". This is used for command synchronization.

Usage: Event

Firmware/Software: V1.0.0.4

***OPT?**

Returns a list of the installed options.

For an improved alternative command, see `SYSTem:BASE:OPTion:LIST?` on page 227.

Return values:

<Options> Comma-separated list of installed options

Usage: Query only

Firmware/Software: V1.0.0.4

***PRE** <Value>

Sets the parallel poll enable (PPE) register to the specified value.

Alternative command: `*XPRE` on page 190

Parameters:
<Value> Range: 0 to 255

Example: `*PRE 132`
Sets bit 2 and bit 7 to 1. Sets the other bits to 0.

Firmware/Software: V1.0.0.4

***PSC** <ClearRegisters>

Specifies whether the `ENABle` status registers are preserved or reset to zero when the instrument is switched on (power on status clear - PSC).

Parameters:
<ClearRegisters> ON | OFF | 0 | 1
OFF / 0: Preserve the register contents at the power-on.
ON / 1: Clear the register contents at the power-on.

Usage: Event

Firmware/Software: V1.0.0.4

***RCL** <Number>

Loads the instrument settings from an intermediate memory identified by the specified number. The instrument settings can be stored to this memory using the command `*SAV` with the associated number.

To load instrument settings from a file to the memory, see `MMEMory:LOAD:STATe` on page 216.

See also `MMEMory:RCL` on page 218.

Parameters:
<Number> Range: 0 to 99

Usage: Event

Firmware/Software: V1.0.0.4

***RST**

Resets all applications. The base settings are left unchanged.

The reset values are indicated in the command descriptions.

Usage: Event

Firmware/Software: V1.0.0.4

***SAV** <Number>

Stores the current instrument settings under the specified number in an intermediate memory. The settings can be restored, using the command `*RCL` with the associated number.

To save the stored instrument settings to a file, see `MMEMory:STORe:STATe` on page 220.

See also `MMEMory:SAV` on page 219.

Parameters:
<Number> Range: 0 to 99

Usage: Event

Firmware/Software: V1.0.0.4

***SRE** <Value>

Sets the service request enable (SRE) register to the specified value.

Alternative command: `*XSRE` on page 191

Parameters:
<Value> Range: 0 to 255

Example: `*SRE 132`
Sets bit 2 and bit 7 to 1. Sets the other bits to 0.

Firmware/Software: V1.0.0.4

***SRQ?** [<Timeout>]

Waits until a service request is generated or the timeout expires. Then returns a status information.

Query parameters:
<Timeout> Waiting for a service request is aborted after the timeout.
If no timeout is specified and no service request is generated, the command waits inifinitely. Other timers for example of the remote control interface may expire in that case.
Default unit: ms

Return values:
<Status> 0 | 1
0
The timer has expired. There was no service request.
1
A service request has been generated.

Usage: Query only

Firmware/Software: V2.0.10

***STB?**

Returns the contents of the status byte (STB) register in decimal form.

Alternative command: `*XSTB?` on page 192

Return values:

<Value> Range: 0 to 255

Example: `*STB?`
Return value `36` for example means: bit 2 and bit 5 = 1, all other bits = 0.

Usage: Query only

Firmware/Software: V1.0.0.4

***WAI**

Waits until all preceding commands have been executed and all signals have settled (see also `*OPC`).

Usage: Event

Firmware/Software: V1.0.0.4

***XESE** <Values>

Sets the event status enable (ESE) register.

The bits to be set to 1 are specified via mnemonics. The other bits are set to 0.

Parameters:

<Values> Comma-separated list of mnemonics, enclosed in brackets
Possible mnemonics are:

PON
Bit 7, power on

URQ
Bit 6, user request

CME
Bit 5, command error

EXE
Bit 4, execution error

DDE
Bit 3, device-dependent error

QYE
Bit 2, query error

RQC
Bit 1, request control

OPC
Bit 0, operation complete

Example: `*XESE (QYE,OPC)`
Sets bit 2 and bit 0 to 1. Sets the other bits to 0.

Firmware/Software: V2.0.10

***XESR?**

Returns the contents of the event status register (ESR) as a list of mnemonics and sets the register to zero.

Return values:

<Value> Comma-separated list of mnemonics, enclosed in brackets
Possible mnemonics are:

PON
Bit 7, power on

URQ
Bit 6, user request

CME
Bit 5, command error

EXE
Bit 4, execution error

DDE
Bit 3, device-dependent error

QYE
Bit 2, query error

RQC
Bit 1, request control

OPC
Bit 0, operation complete

Example: `*XESR?`
Return value `(QYE,OPC)` for example means: bit 2 and bit 0 = 1, all other bits = 0.

Usage: Query only

Firmware/Software: V2.0.10

***XPRE** <Values>

Sets the parallel poll enable (PPE) register.

The bits to be set to 1 are specified via mnemonics. The other bits are set to 0.

Parameters:

<Values> Comma-separated list of mnemonics, enclosed in brackets
Possible mnemonics are:

OPER
Bit 7, `OPERation` status register sum bit

RQS
Bit 6, service request triggered (MSS bit)

ESB
Bit 5, event status register sum bit

MAV
Bit 4, message available

QUES
Bit 3, `QUEStionable` status register sum bit

ERR
Bit 2, error queue not empty

DD2
Bit 1, not used

DD1
Bit 0, not used

Example: `*XPRE (OPER,ERR)`
Sets bit 7 and bit 2 to 1. Sets the other bits to 0.

Firmware/Software: V2.0.10

***XSRE** <Values>

Sets the service request enable (SRE) register.

The bits to be set to 1 are specified via mnemonics. The other bits are set to 0. Bit 6 is always 0.

Parameters:

<Values> Comma-separated list of mnemonics, enclosed in brackets
Possible mnemonics are:

OPER
Bit 7, `OPERation` status register sum bit

ESB
Bit 5, event status register sum bit

MAV
Bit 4, message available

QUES
Bit 3, `QUEStionable` status register sum bit

ERR
Bit 2, error queue not empty

DD2
Bit 1, not used

DD1
Bit 0, not used

Example: `*XPRE (OPER,ERR)`
Sets bit 7 and bit 2 to 1. Sets the other bits to 0.

Firmware/Software: V2.0.10

***XSTB?**

Returns the contents of the status byte (STB) register as a list of mnemonics.

Return values:

<Value> Comma-separated list of mnemonics, enclosed in brackets
Possible mnemonics are:

OPER
Bit 7, `OPERation` status register sum bit

RQS
Bit 6, service request triggered (MSS bit)

ESB
Bit 5, event status register sum bit

MAV
Bit 4, message available

QUES
Bit 3, `QUEStionable` status register sum bit

ERR
Bit 2, error queue not empty

Example: `*XSTB?`
Return value `(ESB,ERR)` for example means: bit 5 and bit 2 = 1, all other bits = 0.

Usage: Query only

Firmware/Software: V2.0.10

8.3 Macro Commands

The following remote control commands are related to macro files and macro execution. All macro commands are SCPI-confirmed.

R&S CMW100-specific command properties

The following macro command properties differ from SCPI stipulations:

- The `*RST` state of `*EMC` is `OFF`. Macro execution must be enabled deliberately for every remote connection.
- Normal remote control execution has priority over macro execution. Macros which have the same name as a remote command supported by the R&S CMW100 are ignored.

***DMC** <MacroLabel>, <MacroSequence>

Creates a macro. If the label exists already, the macro contents are overwritten.

Macros are deleted when a remote connection is closed but can be saved to a macro file for later reuse, see `MMEMory:STORe:MACRo` on page 195.

Avoid using labels which are identical with supported remote control commands. In contrast to SCPI stipulations, remote commands have priority over macros.

Parameters:

<MacroLabel> String parameter, specifies the label to be assigned to the macro.
The label is used to execute the macro and to reference it in other macro commands.

<MacroSequence> Block data element defining the macro contents, typically a sequence of remote control commands.
#0 introduces a data block of indefinite length.
Alternative data format: String with a maximum length of 1023 characters, intended for short command sequences.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

***EMC** <Boolean>

Enables or disables the execution of all macros that are defined for the active remote connection.

Note: In contrast to SCPI specifications, macro execution is disabled by default.

Parameters:

<Boolean> ON | OFF | 0 | 1

Boolean value to enable or disable macro execution. In the disabled state (OFF / 0), macros in a command sequence are not expanded. The R&S CMW100 issues an error message: `113, Undefined header;<MacroLabel>`.

*RST: 0

Example: See Macro Programming Examples

Firmware/Software: V2.1.10

***GMC?** <MacroLabel>

Returns the contents of a macro.

Query parameters:

<MacroLabel> String parameter, selects the macro to be queried.

Return values:

<MacroSequence> <dblock>
Block data element containing the macro contents.

Example: See Macro Programming Examples

Usage: Query only

Firmware/Software: V2.1.10

***LMC?**

Returns the labels of all macros of the active connection.

Return values:

<Label> Comma-separated list of string parameters.
An empty string indicates that no macros are defined for the active connection.

Example: See Macro Programming Examples

Usage: Query only

Firmware/Software: V2.1.10

***PMC**

Deletes all macros of the active remote connection.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

***RMC** <MacroLabel>

Deletes a macro.

Parameters:

<MacroLabel> String parameter, selects the macro to be deleted.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

MMEMory:LOAD:MACRo <MacroLabel>, <MacroFile>

Creates a macro, reading the macro contents from a file. If the label exists already, the macro contents are overwritten.

Avoid using labels which are identical with supported remote control commands. In contrast to SCPI stipulations, remote commands have priority over macros.

Parameters:

<MacroLabel> String parameter, specifies the label to be assigned to the macro.
The label is used to execute the macro and to reference it in other macro commands.

<MacroFile> String parameter, specifies the path and filename of the source file on the instrument.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

MMEMory:STORe:MACRo <MacroLabel>, <MacroFile>

Stores the contents of a macro to a file.

If the file exists, it is overwritten. If the file does not exist, it is created.

Parameters:

<MacroLabel> String parameter, selects the macro to be stored.

<MacroFile> String parameter, specifies the path and filename of the destination file on the instrument.

Example: See Macro Programming Examples

Usage: Event
SCPI confirmed

Firmware/Software: V2.1.10

SYSTem:RECord:MACRo:FILE:STARt <MacroFile>

Starts recording of submitted commands into a macro file.

If the file exists, it is overwritten. If the file does not exist, it is created.

Parameters:

<MacroFile> String parameter, specifies the path and filename of the destination file on the instrument.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

SYSTem:RECord:MACRo:FILE:STOP

Stops recording of commands into a macro file.

Example: See Macro Programming Examples

Usage: Event

Firmware/Software: V2.1.10

8.4 Buffer Commands

The following remote control commands are related to response buffers.

STARt:BASE:BUFFer <BufferLabel>

Creates and activates a buffer.

If the buffer exists already, it is cleared (equivalent to `CLEar:BASE:BUFFer`).

Parameters:

<BufferLabel> String parameter, defines the label of the buffer.
The buffer is identified via this label in all buffer commands.

Example: See Buffer Programming Example

Usage: Event

Firmware/Software: V2.1.10

STOP:BASE:BUFFer

Deactivates the active buffer. Only one buffer can be active at a time.

The buffer and its contents are maintained, but data recording is paused. Use `CONTinue:BASE:BUFFer` to reactivate a buffer.

Example: See Buffer Programming Example

Usage: Event

Firmware/Software: V2.1.10

CONTinue:BASE:BUFFer <BufferLabel>

Reactivates a buffer which was deactivated via `STOP:BASE:BUFFer`).

The R&S CMW100 continues writing data to the buffer.

Parameters:
<BufferLabel> String parameter, selects the buffer to be reactivated.

Example: See Buffer Programming Example

Usage: Event

Firmware/Software: V2.1.10

FETCh:BASE:BUFFer? <BufferLabel>, <LineNo>

Reads the contents of a buffer line.

Buffer contents are stored line by line. Every query generates a new buffer line. The queries are not stored together with the results.

Reading buffer contents is non-destructive. The lines can be read in arbitrary order.

Query parameters:
<BufferLabel> String parameter, selects the buffer to be read.

<LineNo> Line number, selects the line to be read.

Return values:
<LineContents> Returned line contents.

Example: See Buffer Programming Example

Usage: Query only

Firmware/Software: V2.1.10

FETCh:BASE:BUFFer:LINecount? <BufferLabel>

Returns the number of lines in a buffer.

Query parameters:
<BufferLabel> String parameter, selects the buffer to be queried.

Return values:
<Size> Number of lines in the buffer.

Example: See Buffer Programming Example

Usage: Query only

Firmware/Software: V2.1.10

CLEar:BASE:BUFFer <BufferLabel>

Clears the contents of a buffer. You get an empty buffer that you can fill with new commands.

Parameters:
<BufferLabel> String parameter, selects the buffer to be cleared.

Example: See Buffer Programming Example

Usage: Event

Firmware/Software: V2.1.10

DELete:BASE:BUFFer <BufferLabel>

Deletes a buffer.

Parameters:
<BufferLabel> String parameter, selects the buffer to be deleted.

Example: See Buffer Programming Example

Usage: Event

Firmware/Software: V2.1.10

8.5 High-Resolution Timer

After execution of a command, it can be necessary to wait a well-defined time before proceeding with command script processing. For example, because the DUT needs some time to stabilize after a modification. Waiting can be achieved using the following commands.

SYSTem:TIME:HRTimer:ABSolute <Timeout>

This command starts a timer. The timeout is specified relative to an already set timestamp, see `SYSTem:TIME:HRTimer:ABSolute:SET` on page 199.

When the timer expires, `"Operation Complete"` is indicated. This event can be evaluated by polling, via a `*OPC?` or via `*WAI`.

Setting parameters:
<Timeout> Range: 0 ms to 4294967295 ms
Default unit: ms

Example:
```
SYST:TIME:HRT:ABS:SET
SYST:TIME:HRT:ABS 1000
*OPC?
```
The three commands have the following effects:
Set a timestamp with the current system time.
Start a timer with a timeout value of 1 second relative to the timestamp.
Wait until the timer expires before processing the next command.

Usage: Event
Asynchronous command

Firmware/Software: V2.1.10

SYSTem:TIME:HRTimer:ABSolute:SET

This command sets a timestamp with the current system time. A timer can be started with a timeout relative to this timestamp, see `SYSTem:TIME:HRTimer:ABSolute` on page 198.

An existing timestamp is overwritten.

Return values:
<Timestamp> A query returns seven values, separated by commas:
Year, month, day, hour, minutes, seconds, milliseconds

Example:
```
SYSTem:TIME:HRTimer:ABSolute:SET
```
Set a timestamp with the current system time.
```
SYSTem:TIME:HRTimer:ABSolute:SET?
```
Query the timestamp. Possible result:
```
2011,3,20,12,5,30,999
```

Firmware/Software: V2.1.10

SYSTem:TIME:HRTimer:RELative <Timeout>

This command starts a timer. After the specified timeout, an OPC is generated.

When the timer expires, `"Operation Complete"` is indicated. This event can be evaluated by polling, via a `*OPC?` or via `*WAI`.

Setting parameters:
<Timeout> Range: 0 ms to 4294967295 ms
Default unit: ms

Example:
```
SYST:TIME:HRT:REL 1000
*OPC?
```
The commands have the following effects:
Start the timer with a timeout value of 1 second.
Wait until the timer expires before processing the next command.

Usage: Event
Asynchronous command

Firmware/Software: V2.1.10

8.6 Error Queue

The following commands query (and delete) the error queue.

SYSTem:ERRor:ALL?

Queries and deletes all entries in the error queue.

Each entry consists of an error number and a short error description. Positive error numbers are instrument-specific. Negative error numbers are reserved by the SCPI standard.

Example: `SYSTem:ERRor:ALL?`
Query all entries in the error queue. If the error queue is empty, `0, "No error"` is returned.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

SYSTem:ERRor:CODE:ALL?

Queries and deletes all entries in the error queue.

The command returns only the error numbers, not the error descriptions. Positive error numbers are instrument-specific. Negative error numbers are reserved by the SCPI standard.

Example: `SYSTem:ERRor:CODE:ALL?`
Query all entries in the error queue. If the error queue is empty, `"0"` is returned.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

SYSTem:ERRor:CODE[:NEXT]?

Queries and deletes the oldest entry in the error queue.

The command returns only the error number, not the error description. Positive error numbers are instrument-specific. Negative error numbers are reserved by the SCPI standard.

Example: `SYSTem:ERRor:CODE?`
Query the oldest entry in the error queue. If the error queue is empty, `"0"` is returned.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

SYSTem:ERRor:COUNt?

Queries the number of entries in the error queue.

Example: `SYSTem:ERRor:COUNt?`
If the queue is empty, `0` is returned.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

SYSTem:ERRor[:NEXT]?

Queries and deletes the oldest entry in the error queue.

Each entry consists of an error number and a short error description. Positive error numbers are instrument-specific. Negative error numbers are reserved by the SCPI standard.

The command has the same effect as `STATus:QUEue[:NEXT]?` on page 251.

Example: `SYSTem:ERRor?`
Query the oldest entry in the error queue. If the error queue is empty, `0, "No error"` is returned.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

8.7 Reset and Preset

The following commands reset or preset the instrument or specific software parts.

SYSTem:PRESet:ALL
SYSTem:RESet:ALL

A `PRESet` sets the parameters of all subinstruments and the base settings to default values suitable for local/manual interaction. A `RESet` sets them to default values suitable for remote operation.

Example: `SYSTem:PRESet:ALL`
Force the entire R&S CMW100 to a preset state optimized for manual operation.

Usage: Event

Firmware/Software: V1.0.5.3

Manual operation: See "Reset" on page 84

SYSTem:PRESet:BASE
SYSTem:RESet:BASE

A `PRESet` sets the base settings to default values suitable for local/manual interaction. A `RESet` sets them to default values suitable for remote operation.

Example: `SYSTem:PRESet:BASE`
Force the base settings to a preset state optimized for manual operation.

Usage: Event

Firmware/Software: V1.0.5.3

Manual operation: See "Reset" on page 84

SYSTem:PRESet [<Application>]
SYSTem:RESet [<Application>]

A `PRESet` sets the parameters of the subinstrument to default values suitable for local/manual interaction. A `RESet` sets them to default values suitable for remote operation.

Optionally, the preset/reset can be limited to a specific application instance.

Parameters:

<Application> String specifying an application and instance to be reset/preset.
Example: `'LTE Meas1'` for LTE UE measurements instance 1
Omitting the instance (e.g. `'LTE Meas'`) selects instance 1.
The supported strings are listed in the table below.

Example: `SYSTem:PRESet 'GPRF Meas1'`
Force the GPRF measurement instance 1 to a preset state optimized for manual operation.

Usage: Event

Firmware/Software: V1.0.5.3
V3.0.10: [<Application>] added

Manual operation: See "Reset" on page 84

Depending on your instrument model and the installed options, only a part of the following table is relevant for you.

Group	<Application> strings without instance
1xEV-DO	`'1xEV-DO Meas'`
Bluetooth	`'Bluetooth Meas'`
CDMA2000	`'CDMA2000 Meas'`
GPRF	`'GPRF Gen'` `'GPRF Meas'`
GSM	`'GSM BS Meas'` `'GSM Meas'` (for GSM MS measurements)
LR-WPAN	`'LR-WPAN Meas'`
LTE	`'LTE eNodeB Meas'` `'LTE Meas'` (for LTE UE measurements)
NB-IoT	`'NB-IoT eNodeB Meas'` `'NB-IoT Meas'`
New Radio	`'NR Sub-6 GHz'` (measurement)
TD-SCDMA	`'TD-SCDMA Meas'`
WCDMA	`'WCDMA Meas'` (for WCDMA UE measurements) `'WCDMA NodeB Meas'`
WLAN	`'WLAN Meas'`

8.8 Tracing the Remote Control Interface

The commands in this section configure and control tracing of the remote interface and events into a file and enable/disable the display of the SCPI remote trace.

Before you start tracing, configure all settings as desired. Modifying settings while tracing is active can result in loss of already traced data. Useful exception: Selecting a new target file while tracing is allowed. For start mode "EXPLicit", a restart of the instrument resets the settings to the documented default values.

If you want to start tracing already during start of the instrument, configure all settings (including start mode "AUTO"). Then restart your instrument. Tracing is started automatically during the restart, using the already configured settings.

If you use an XML file as trace file, ensure that tracing is stopped properly. If tracing is aborted instead of stopped, e.g. by shutting down the instrument for stop mode "EXPLicit", the XML file becomes invalid, because some tags are not closed.

If tracing is started with an existing trace file, a backup of the trace file is created and the file itself is reset and overwritten. The same action is performed if the maximum file size is reached (except for stop mode "BUFFerfull"). When the file is full for the second time or tracing is started the next time, the first backup file is lost because it is overwritten by the next backup. To prevent loss of data, set a sufficient file size, select an appropriate stop mode and archive/copy completed trace files if you want to keep them.

TRACe:REMote:MODE:FILE<inst>:ENABle <Enable>

Enable or disable tracing of the remote interface to a file for the specified subinstrument.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<Enable> Boolean value (ON | OFF | 1 | 0)
Default value: OFF

Example: `TRACe:REMote:MODE:FILE1:ENABle ON`
Enable tracing for subinstrument 1.

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:FILTer <Input>, <Output>, <Error>, <Trigger>, <DeviceClear>, <StatusRegister>, <Connection>, <RemoteLocal>

Specifies a filter for tracing of the specified subinstrument. The filter defines which message types and events are traced into a file.

The default setting is ON,ON,ON,OFF,OFF,OFF,OFF,OFF. All parameters support ON | OFF | 1 | 0.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<Input> Trace incoming commands.

<Output> Trace outgoing responses to queries.

<Error> Trace SCPI error messages.

<Trigger> Trace trigger events.

<DeviceClear> Trace device clear messages.

<StatusRegister> Trace status register changes.

<Connection> Trace remote connection changes.

<RemoteLocal> Trace transitions between local and remote operation mode.

Example: `TRACe:REMote:MODE:FILE1:FILTer ON,ON,ON,OFF,OFF,OFF,OFF,ON`
Trace incoming commands, outgoing responses to queries, error messages and remote-local transitions for subinstrument 1.

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:FORMat <Format>

Specifies the format of the target file for tracing of the remote interface for the specified subinstrument. The trace can be stored as ASCII file or as XML file.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<Format> ASCii | XML
Default value: XML

Example: `TRACe:REMote:MODE:FILE1:FORMat XML`
Select XML as file format for subinstrument 1.

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:NAME <FilePath>

Specify path and name of the target file for tracing of the remote interface. For different subinstruments, specify different files.

If you specify a new target file while tracing, the old target file is closed, the new file is created and tracing is continued with the new file.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<FilePath> String parameter specifying path and name of the file
Default values: "@LOG\RemoteTrace-Inst0.xml" and "@LOG\RemoteTrace-Inst1.xml"

Example: `TRACe:REMote:MODE:FILE1:NAME "@LOG\trace1.xml"`
Specifies the file `trace1.xml` in the log directory as target file for tracing (subinstrument 1).

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:SIZE <FileSize>

Specifies the maximum size of the trace file in bytes.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<FileSize> Recommended minimum value: 40000 bytes
Maximum value: 1000000000 bytes (1 GB)
Default value: 1000000000 bytes (1 GB)

Example: `TRACe:REMote:MODE:FILE1:SIZE 100000000`
Set 100 MB as maximum size for the trace file of subinstrument 1.

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:STARtmode <StartMode>

Specifies whether tracing is started automatically or manually.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<StartMode> AUTO | EXPLicit

AUTO: Start tracing automatically when the instrument is started.
EXPLicit: Start tracing via the command `TRACe:REMote:MODE:FILE<inst>:ENABle`.
Default value: EXPLicit

Example: `TRACe:REMote:MODE:FILE1:STARtmode AUTO`
Enable automatic start of tracing for the next start of the instrument.

Firmware/Software: V2.0.10

TRACe:REMote:MODE:FILE<inst>:STOPmode <StopMode>

Specifies how / when tracing is stopped and the trace file is closed.

Suffix:
<inst> 1..4
Selects the subinstrument

Parameters:
<StopMode> AUTO | EXPLicit | ERRor | BUFFerfull

AUTO: Stop tracing automatically when the instrument is shut down.
EXPLicit: Stop tracing via the command `TRACe:REMote:MODE:FILE<inst>:ENABle`.
ERRor: Stop tracing when a SCPI error occurs.
BUFFerfull: Stop tracing when the maximum file size is reached.
Default value: EXPLicit

Example: `TRACe:REMote:MODE:FILE1:STOPmode BUFFerfull`
Stop tracing when the maximum file size is reached (subinstrument 1).

Firmware/Software: V2.0.10

TRACe:REMote:MODE:DISPlay:ENABle <Enable>

Enables or disables the display of the SCPI remote trace. Two modes are available when the display is enabled: a live mode and an analysis mode.

Parameters:
<Enable> ANALysis | LIVE | OFF

ANALysis: Stop tracing to analyze already traced messages.
LIVE: Trace messages and display them.
OFF: Disable the report display.
Default value: OFF

Example: `TRACe:REMote:MODE:DISPlay:ENABle LIVE`
Activate the live mode.

Firmware/Software: V2.1.10

TRACe:REMote:MODE:DISPlay:CLEar

Clears the display of the SCPI remote trace in analysis mode.

Usage: Event

Firmware/Software: V3.0.10

8.9 Mass Memory Commands

The MMEMory subsystem provides mass storage capabilities for the R&S CMW100. For MMEMory commands related to command macros, see Chapter 8.3, "Macro Commands", on page 192.

The following rules apply to parameters specifying files or directories:

- Parameters for specification of file names and directory names are strings.
- You can specify complete absolute paths including the drive name and all subdirectories. For locations in a network, use the universal naming convention (UNC) format `'\\server\share'`.
  Examples: `'C:\TEMP\TRASH\test.txt'` specifies the file named `test.txt` in the directory `TEMP\TRASH` on volume `C:\` of the system drive.
  `'\\myserver\data\archive\test.txt'` specifies the file named `test.txt` in the subdirectory `archive` of the share `data` on the server `myserver`.
- If you specify a string containing an absolute path but no drive or server, the default storage unit is prefixed automatically (see `MMEMory:MSIS` on page 218).
- If you specify a string containing a relative path, the current directory is prefixed automatically (see `MMEMory:CDIRectory` on page 212).
- Alias strings facilitate the entry of paths. To display the predefined alias strings and the assigned paths, see `MMEMory:ALIases?` on page 209. The defined strings can be combined with normal text to specify file paths. Example:
  `'@SAVE\myfile.dfl'` specifies the file `myfile.dfl` in the save directory, assigned to the alias `@SAVE`.
- The use of wildcards `?` and `*` is only allowed if explicitly stated for a command. Wildcards are only allowed in the last component of the string.
  Example: `'archive\*test*'` allowed, `'archi*\test1.txt'` not allowed.
- A file name itself can contain the period as a separator for extensions.
- A single period represents the current directory.
- A double period represents the parent of the current directory.
- Example: Assume that the default storage unit equals `'X:'` and the current directory equals `'\user\data'`. Then the following commands yield the same result:

```
MMEM:CAT?
MMEM:CAT? '*'
MMEM:CAT? '*.*'
MMEM:CAT? '.\*.*'
MMEM:CAT? '..\data\*.*'
MMEM:CAT? '..\data'
MMEM:CAT? '\user\data\*.*'
MMEM:CAT? 'X:\user\data\*.*'
```

File and directory names can be chosen according to Windows™ conventions. The restrictions placed on file names known from DOS systems do not apply. All letters and numbers are allowed, also the special characters "_", "^", "$", "~", "!", "#", "%", "&", "-", "{", "}", "(", ")", "@" and "`". Reserved file names are
`CON, AUX, COM1, ..., COM4, LPT1, ..., LPT3, NUL` and `PRN`.

Mass memory commands are marked as "SCPI confirmed" if they provide at least the functionality defined by SCPI and can be used as specified by SCPI. Nevertheless, these commands can provide additional features not specified by SCPI, like wildcards. The functionality can also be enhanced via additional optional parameters.

List of commands:

MMEMory:ALIases?

Returns the defined alias entries and the assigned directories. These settings are predefined and cannot be configured.

Example: `MMEM:ALIases?`
Possible response:
```
"@SAVE",
"C:\ProgramData\Rohde-Schwarz\CMW\Data\Save",
"@PRINT",
"C:\ProgramData\Rohde-Schwarz\CMW\Data\Print",
"@USERDATA",
"C:\ProgramData\Rohde-Schwarz\CMW\Data\"
```

Example: When you specify a file destination, an alias can be used instead of the corresponding path. With the definition of the alias `@SAVE` listed above, the following two strings identify the same file:
```
"@SAVE\A\myfile.dfl"
"C:\ProgramData\Rohde-Schwarz\CMW\Data\Save\A\
myfile.dfl"
```

Usage: Query only

Firmware/Software: V2.0.10

MMEMory:ATTRibute <FileName>, <Action>
MMEMory:ATTRibute? [<FileName>]

Sets or removes file attributes for files and directories.

Parameters:

<Action> Attribute actions to be performed. The attributes are R (read-only), A (archive), S (system), H (hidden).
Set an attribute with a plus or clear an attribute with a minus as prefix. Separate several actions with a blank, see example.

Parameters for setting and query:

<FileName> String parameter to specify the name and/or path of the objects for which the attributes are modified or returned.
The wildcards * and ? are allowed.
If a directory is specified instead of a file, the directory itself and all files and subdirectories contained in that directory are modified/returned.
If the parameter is omitted completely for a query, the current directory is used as default value (see MMEMory:CDIRectory).

Return values:

<Information> Comma-separated list of strings.
Information strings are returned for the directories "." and "..", for files and for subdirectories.
Each string has the format '<ObjectName>,<Attributes>'.

Example: `MMEM:ATTR 'X:\USER\DATA\*.LOG', '-R +A -H'`
Clears attributes R and H and sets attribute A for all files with the extension LOG in directory X:\USER\DATA.

Example: `MMEM:ATTR? 'X:\USER\DATA'`
Returns attribute information for all files and subdirectories of directory X:\USER\DATA.
Possible response:
`".,",",..,","myfile.txt,RA","mysubdirectory,"`

Firmware/Software: V1.0.0.4

MMEMory:CATalog? [<DirectoryName>][,<Mode>]

Returns information on the contents of the current or of a specified directory.

Parameters:

<DirectoryName> String parameter to specify the directory. If this parameter is omitted, the command queries the contents of the current directory (see MMEMory:CDIRectory).
If the wildcards ? or * are used, only files and directories matching this pattern are returned.

<Mode> ALL | WTIme

ALL
Output enhanced with date, time and file attributes

WTIme
Output enhanced with date and time

Return values:

<UsedSize> Disk space in bytes used by the listed files, excluding subdirectories

<FreeDiskSpace> Available free disk space in bytes

<Information> Comma-separated list of strings.
Information strings are returned for the directories "." and "..", for files and for subdirectories.
Each string contains the following contents:
<Name>,<Type>,<Size>[,<DateTime>][,<Attributes>]

<Name>
Name of the object (file or directory)

<Type>
Type of the object: DIRectory | BINary | ASCii | STATe

<Size>
File size in bytes (0 for directories)

<DateTime>
Date and time of last modification
Only returned for <Mode> = ALL or WTIme

<Attributes>
Object attributes, only returned for <Mode> = ALL
Values: R = read-only, A = archive, S = system, H = hidden

Example:
```
MMEM:CAT?
```
Possible response:
```
156673,5195137024,
".,DIR,0","..,DIR,0",
"SaveFile001.xml,BIN,78335",
"SaveFile002.xml,BIN,78338"
```

Example:
```
MMEM:CAT? 'X:\User\Data\*.jpg'
```
Possible response:
```
300928,511337022,
"fig1.jpg,BIN,259395",
"screen.jpg,BIN,41533"
```
Considers all JPG files located in directory `X:\User\Data`.

Example:
```
MMEM:CAT? 'X:\User\Data\*.jpg',ALL
```
Possible response:
```
300928,511337022,
"fig1.jpg,BIN,259395,14-01-2011 10:54,A",
"screen.jpg,BIN,41533,10-01-2011 11:13,AR"
```
Displays all information for the JPG files located in directory `X:\User\Data`.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:CATalog:LENGth? [<DirectoryName>]

Returns the number of files and subdirectories of the current or of a specified directory. The number includes the directory strings "." and ".." so that it corresponds to the number of strings returned by the `MMEMory:CATalog?` command after the initial numeric parameters.

Parameters:

<DirectoryName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory (see `MMEMory:CDIRectory`).
If the wildcards `?` or `*` are used, the number of files and subdirectories matching this pattern are returned.

Example: `MMEM:CAT:LENG?`
Response: `4` (corresponding to the `MMEMory:CATalog?` example)

Example: `MMEM:CAT:LENG? 'X:\User\Data\session*'`
Returns the number of files and subdirectories starting with `session` which are located in directory `X:\User\Data`.

Usage: Query only

Firmware/Software: V1.0.5.3

MMEMory:CDIRectory [<DirectoryName>]

Changes the current directory for mass memory storage.

If <DirectoryName> is omitted, the current directory is set to `'\'`. If <DirectoryName> contains not only a directory, but also a drive letter or server name, the command `MMEM:MSIS` is also executed automatically.

Parameters:

<DirectoryName> String parameter to specify the directory. Wildcards are not allowed.

*RST: a *RST does not change the current directory

Example: `MMEM:CDIR 'X:\User\Data'`
Changes the current directory to `X:\User\Data`

Usage: SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:COPY <FileSource>[, <FileDestination>]

Copies an existing file. The target directory must exist.

Parameters:

<FileSource> String parameter to specify the name of the file to be copied. Wildcards `?` and `*` are allowed if <FileDestination> contains a path without filename.

<FileDestination> String parameter to specify the path and/or name of the new file. If no file destination is specified, the source file is written to the current directory (see `MMEMory:CDIRectory`). Wildcards are not allowed.

Example: `MMEM:COPY 'X:\USER\DATA\File1.pdf','X:\Archive'`
Copies `File1.pdf` in directory `X:\USER\DATA` to directory `X:\Archive`.

Example: `MMEM:COPY 'X:\USER\File1.pdf','X:\File2.pdf'`
Copies `File1.pdf` in directory `X:\USER` to `X:\` and renames the file to `File2.pdf`.

Usage: Event
SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:DATA <FileName>, <Data>
MMEMory:DATA? <FileName>

Stores the specified block data into the specified file.

Parameters:

<Data> <dblock>

Data in 488.2 block data format. The delimiter EOI must be selected to achieve correct data transfer. See also Block Data Format.

Parameters for setting and query:

<FileName> String parameter to specify the name of the file. Wildcards are not allowed.

Example: `MMEM:DATA 'C:\TEMP\TEST01.HCP', #219Content of the file`
Stores the data `Content of the file` to the indicated file. `#2` indicates that the next two characters (`19`) indicate the data length.

Example: `MMEM:DATA? 'C:\TEMP\TEST01.HCP'`
Returns the data contained in file `TEST01.HCP` in block data format

Usage: SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:DCATalog? [<DirectoryName>]

Returns the subdirectories of the current or of a specified directory.

Parameters:

<DirectoryName> String parameter to specify the directory. If this parameter is omitted, the command queries the contents of the current directory (see `MMEMory:CDIRectory`).
If the wildcards `?` or `*` are used, only the subdirectories matching this pattern are returned.

Return values:

<Directory> Comma-separated list of strings with subdirectory names, one string per name

Example: `MMEM:DCAT?`
Response: `".","..","temp","test","mydirectory"`

Example: `MMEM:DCAT? 'X:\User\Data\session*'`
Response: `"session1","session5","sessiontest"`
The response lists the subdirectories located in directory `X:\User\Data` which start with `session`.

Usage: Query only

Firmware/Software: V1.0.5.3

MMEMory:DCATalog:LENGth? [<DirectoryName>]

Returns the number of subdirectories of the current or of a specified directory. The number includes the directory strings "." and ".." so that it corresponds to the number of strings returned by the `MMEMory:DCATalog?` command.

Parameters:

<DirectoryName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory (see `MMEMory:CDIRectory`).
If the wildcards `?` or `*` are used, the number of subdirectories matching this pattern are returned.

Example: `MMEM:DCAT:LENG?`
Response: `5` (corresponding to the `MMEMory:DCATalog?` example)

Example: `MMEM:DCAT:LENG? 'X:\User\Data\session*'`
Returns the number of subdirectories starting with `session` which are located in directory `X:\User\Data`.

Usage: Query only

Firmware/Software: V1.0.5.3

MMEMory:DELete <FileName>

Deletes the specified files.

Parameters:

<FileName> String parameter specifying the file to be deleted. The wildcards `*` and `?` are allowed. Specifying a directory instead of a file is not allowed.

Example: `MMEM:DEL 'C:\TEMP\TEST01.HCP'`
Deletes file `TEST01.HCP` in directory `C:\TEMP`

Example: `MMEM:DEL 'C:\TEMP\*.*'`
Deletes all files in directory `C:\TEMP`

Usage: Event
SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:DRIVes?

Returns a list of the available drives.

Example: `MMEM:DRIV?`
Possible response: `"C:\","D:\"`

Usage: Query only

Firmware/Software: V1.0.5.3

MMEMory:LOAD:ITEM <ItemName>, <FileSource>

Executes a partial recall. That means, restores a selected part of a save file.

You can restore all settings of a specific application instance, for example of the GPRF generator instance 1.

Or you can restore the list mode settings of a specific measurement application instance, for example the list mode settings of the LTE multi-evaluation measurement instance 1.

Parameters:

<ItemName> String parameter identifying the part to be restored.
<ItemName> = <Application>[<i>][:MEV:LIST]
For <Application>, see table below.
<i> is the instance of the application. Omitting <i> restores instance 1.
Appending `:MEV:LIST` restores only the list mode settings.

<FileSource> String parameter specifying the path and filename of the source file. Wildcards are not allowed.

Example: `MMEM:LOAD:ITEM 'GPRF Meas', '@SAVE\mysavefile.dfl'`
Restores the settings of the GPRF measurement instance 1 from the file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias.

Example: `MMEM:LOAD:ITEM 'GSM Meas1:MEV:LIST', '@SAVE\mysavefile1.dfl'`
Restores the list mode settings of the GSM measurement instance 1 from the file `mysavefile1.dfl` located in the directory assigned to the `@SAVE` alias.

Usage: Event

Firmware/Software: V3.0.10

Manual operation: See "Recall / Partial Recall" on page 85

Depending on your instrument model and the installed options, only a part of the following table is relevant for you.

Group	<Application>
1xEV-DO	`1xEV-DO Meas`
Bluetooth	`Bluetooth Meas`
CDMA2000	`CDMA2000 Meas`
GPRF	`GPRF Gen` `GPRF Meas`
GSM	`GSM BS Meas` `GSM Meas` (for GSM MS measurements)
LR-WPAN	`LR-WPAN Meas`
LTE	`LTE eNodeB Meas` `LTE Meas` (for LTE UE measurements)
NB-IoT	`NB-IoT eNodeB Meas` `NB-IoT Meas`
New Radio	`NR Sub-6 GHz` (measurement)
TD-SCDMA	`TD-SCDMA Meas`
WCDMA	`WCDMA Meas` (for WCDMA UE measurements) `WCDMA NodeB Meas`
WLAN	`WLAN Meas`

MMEMory:LOAD:STATe <MemoryNumber>, <FileSource>

Loads the instrument settings from the specified file to the specified internal memory. After the file has been loaded, the settings must be activated using a `*RCL` command.

For more convenience, see `MMEMory:RCL` on page 218.

Parameters:

<MemoryNumber> Number of the internal memory to which the settings are loaded.

<FileSource> String parameter specifying the source file. Wildcards are not allowed.

Example: `MMEM:LOAD:STATe 4,'@SAVE\mysavefile.dfl'`
`*RCL 4`
Loads instrument settings from file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias to the internal memory number 4. Activates the settings in internal memory number 4.

Usage: Event
SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:MDIRectory <DirectoryName>

Creates a directory. If necessary, an entire path consisting of several subdirectories is created.

Parameters:

<DirectoryName> String parameter to specify the directory. Wildcards are not allowed.

Example: `MMEM:MDIR 'X:\User\Data\Images\Recent'`
Assuming that `X:\User\Data` exists, the subdirectories `Images` and `Recent` are created.

Usage: Event

Firmware/Software: V1.0.0.4

MMEMory:MOVE <FileSource>, <FileDestination>

Moves or renames an existing object (file or directory) to a new location.

Parameters:

<FileSource> String parameter to specify the name of the object to be moved or renamed.
Wildcards `?` and `*` are only allowed for moving files without renaming.

<FileDestination> String parameter to specify the new name and/or path of the object. Wildcards are not allowed.
If a new object name without path is specified, the object is renamed.
If a new path without object name is specified, the object is moved to this path.
If a new path and a new object name are specified, the object is moved to this path and renamed.

Example: `MMEM:MOVE 'X:\Temp\Setup.cfg','X:\Archive'`
Moves file `Setup.cfg` from `X:\Temp` to `X:\Archive`.

Example: `MMEM:MOVE 'X:\Temp\Setup.cfg','Test.cfg'`
Renames file `Setup.cfg` in directory `X:\Temp` to `Test.cfg`.

Example: `MMEM:MOVE 'X:\Setup.cfg','X:\Archive\Test.cfg'`
Moves file `Setup.cfg` from `X:\` to `X:\Archive` and renames it to `Test.cfg`.

Usage: Event
SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:MSIS <StorageUnit>

Changes the default storage unit (drive or server) for mass memory storage.

When the default storage unit is changed, it is checked whether the current directory (see `MMEMory:CDIRectory`) is also available on the new storage unit. If not, the current directory is automatically set to `'\'`.

Parameters:
<StorageUnit> String parameter to specify the default storage unit.
*RST: a *RST does not change the default storage unit

Example: `MMEM:MSIS 'E:'`
Sets the default storage unit to drive E.

Example: `MMEM:MSIS '\\Server\Share'`
Sets the default storage unit to the specified server.

Usage: SCPI confirmed

Firmware/Software: V1.0.0.4

MMEMory:RCL <FileSource>

Restores the instrument settings from the specified file.

This command has the same effect as the combination of `MMEMory:LOAD:STATe` and `*RCL`.

Parameters:
<FileSource> String parameter specifying the path and filename of the source file. Wildcards are not allowed.

Example: `MMEM:RCL '@SAVE\mysavefile.dfl'`
Loads and activates the instrument settings from the file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias.

Usage: Event

Firmware/Software: V2.0.10

Manual operation: See "Recall / Partial Recall" on page 85

MMEMory:RDIRectory <DirectoryName>

Removes an existing empty directory from the mass memory storage system.

Parameters:

<DirectoryName> String parameter to specify the directory. Wildcards are not allowed.

Example: `MMEM:RDIR 'X:\User\Data\myFolder'`
Removes the directory `myFolder`.

Usage: Event

Firmware/Software: V1.0.0.4

MMEMory:SAV <FileDestination>

Stores the current instrument settings to the specified file.

This command has the same effect as the combination of `*SAV` and `MMEMory:STORe:STATe`.

Parameters:

<FileDestination> String parameter specifying path and filename of the target file. Wildcards are not allowed.

Example: `MMEM:SAV '@SAVE\mysavefile.dfl'`
Saves the current instrument settings to the file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias.

Usage: Event

Firmware/Software: V2.0.10

Manual operation: See "Save / Partial Save" on page 85

MMEMory:STORe:ITEM <ItemName>, <FileDestination>

Executes a partial save, i.e. stores a part of the instrument settings to the specified file.

You can store all settings of a specific application instance, for example of the LTE signaling application instance 1.

Or you can store the list mode settings of a specific measurement application instance, for example the list mode settings of the LTE multi-evaluation measurement instance 1.

Parameters:

<ItemName> String parameter identifying the part to be saved.
<ItemName> = <Application>[<i>][:MEV:LIST]
For <Application>, see `MMEMory:LOAD:ITEM` on page 215.
<i> is the instance of the application. Omitting <i> stores instance 1.
Appending :MEV:LIST stores only the list mode settings.

<FileDestination> String parameter specifying the path and filename of the target file. Wildcards are not allowed.

Example: `MMEM:STOR:ITEM 'GPRF Meas', '@SAVE\mysavefile.dfl'`
Saves the settings of the GPRF measurement instance 1 to the file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias.

Example: `MMEM:STOR:ITEM 'GSM Meas1:MEV:LIST', '@SAVE\mysavefile1.dfl'`
Saves the list mode settings of the GSM measurement instance 1 to the file `mysavefile1.dfl` located in the directory assigned to the `@SAVE` alias.

Usage: Event

Firmware/Software: V3.0.10

Manual operation: See "Save / Partial Save" on page 85

MMEMory:STORe:STATe <MemoryNumber>, <FileDestination>

Stores the instrument settings from the specified internal memory to the specified file.

To store the current instrument settings to a file, use first `*SAV <MemoryNumber>` to store the settings to the memory. Then use this command to store the settings from the memory to a file.

For more convenience, see `MMEMory:SAV` on page 219.

Parameters:

<MemoryNumber> Number of the internal memory to which the settings have been stored using *SAV.

<FileDestination> String parameter specifying path and filename of the target file. Wildcards are not allowed.

Example:
```
*SAV 4
MMEM:STORe:STATe 4,'@SAVE\mysavefile.dfl'
```
Saves the current instrument settings to the internal memory number 4. Stores the settings from the internal memory number 4 to the file `mysavefile.dfl` located in the directory assigned to the `@SAVE` alias.

Usage: Event
SCPI confirmed

Firmware/Software: V1.0.0.4

8.10 LAN Settings

The following commands configure LAN connection settings.

SYSTem:COMMunicate:NET:HOSTname?

Queries the host name (computer name) of the R&S CMW100. The host name is part of the VISA address string for LAN-based connections.

Return values:
<Host> Host name as string

Usage: Query only

Firmware/Software: V1.0.0.4

Manual operation: See "Hostname" on page 89

SYSTem:COMMunicate:NET:ADAPter <Adapter>

Selects a LAN network adapter for configuration via other `SYSTem:COMMunicate:NET:...` commands.

Parameters:
<Adapter> String parameter identifying the network adapter
*RST: A *RST does not affect the LAN service settings.

Firmware/Software: V1.0.0.4

Manual operation: See "Network Adapter" on page 88

SYSTem:COMMunicate:NET:DHCP <Boolean>

Enables or disables the dynamic host configuration protocol (DHCP).

Parameters:
<Boolean> ON | OFF | 0 | 1
ON / 1: DHCP enabled, automatic TCP/IP address configuration.
OFF / 0: DHCP disabled, manual address configuration.
*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:DHCP 0`
Disable DHCP. Enable manual setting of the IP address information.

Firmware/Software: V1.0.0.4

Manual operation: See "DHCP" on page 89

SYSTem:COMMunicate:NET:IPADdress <Address>

Assigns one or more IPv4 addresses to the network adapter. This command is only relevant if DHCP is disabled.

A query returns the currently assigned addresses, irrespective of whether they have been assigned manually or via DHCP.

Parameters:

<Address> String parameter, IPv4 address consisting of four blocks (octets) separated by dots
Several strings separated by commas can be entered or several addresses separated by commas can be included in one string.

*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:IPADdress '10.113.10.38'`
Select a private IP address (characterized by 10 in the first octet).

Firmware/Software: V1.0.0.4, V3.0.10: support of more than one address

Manual operation: See "IP Addresses, Subnet Masks, Gateways" on page 89

SYSTem:COMMunicate:NET:SUBNet:MASK <Mask>

Defines the subnet masks to be used for the network adapter IPv4 addresses. This command is only relevant if DHCP is disabled.

A query returns the currently used subnet masks, irrespective of whether they have been assigned manually or via DHCP.

Parameters:

<Mask> String parameter, IPv4 subnet mask consisting of four blocks separated by dots
Several strings separated by commas can be entered or several masks separated by commas can be included in one string.

*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:SUBNet:MASK '255.255.0.0'`
Sets subnet mask 255.255.0.0.

Firmware/Software: V1.0.0.4, V3.0.10: support of more than one subnet mask

Manual operation: See "IP Addresses, Subnet Masks, Gateways" on page 89

SYSTem:COMMunicate:NET:GATeway <Address>

Defines IPv4 addresses of default gateways. This command is only relevant if DHCP is disabled.

A query returns the currently defined addresses, irrespective of whether they have been specified manually or via DHCP.

Parameters:

<Address> String parameter, gateway IPv4 address consisting of four blocks separated by dots
Several strings separated by commas can be entered or several addresses separated by commas can be included in one string.
*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:GATeway '10.113.0.1'`
Sets the default gateway address to 10.113.0.1.

Firmware/Software: V1.0.0.4, V3.0.10: support of more than one address

Manual operation: See "IP Addresses, Subnet Masks, Gateways" on page 89

SYSTem:COMMunicate:NET:DNS:ENABle <Boolean>

Enables or disables dynamic configuration of DNS server addresses.

Parameters:

<Boolean> ON | OFF | 0 | 1
ON / 1: Enabled, automatic configuration
OFF / 0: Disabled, manual configuration
*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:DNS:ENABle 0`
Disables dynamic/automatic DNS server address configuration.

Firmware/Software: V1.0.0.4

Manual operation: See "Obtain DNS Server Address Automatically" on page 89

SYSTem:COMMunicate:NET:DNS <Address>

Defines the DNS server IPv4 addresses to be used. The addresses are valid if dynamic configuration is disabled.

A query returns the defined DNS addresses, irrespective of whether they have been specified manually or via DHCP.

Parameters:

<Address> String parameters, DNS server IPv4 addresses consisting of four blocks separated by dots
Several strings separated by commas can be entered or several addresses separated by commas can be included in one string.
*RST: A *RST does not affect the LAN service settings.

Example: `SYSTem:COMMunicate:NET:DNS '10.0.2.166', '10.0.23.159'`
Use 10.0.2.166 for primary DNS server and 10.0.23.159 for secondary DNS server.

Firmware/Software: V1.0.0.4, V3.0.10: support of more than two addresses

Manual operation: See "DNS Servers" on page 89

8.11 Remote Control Settings

The following commands define the address information for the SCPI remote-control interfaces of the R&S CMW100.

SYSTem:COMMunicate:HISLip<i>:VRESource?

Queries the VISA resource string for the HiSLIP protocol.

Suffix:
<i> 1..32
Selects the remote channel.

Return values:
<String> VISA address string
*RST: *RST has no effect on the value.

Example:
```
SYSTem:COMMunicate:NET:HOSTname 'hh346999'
SYSTem:COMMunicate:HISLip2:VRESource?
```
Define a hostname and query the VISA resource string for the HiSLIP protocol (channel 2). The response is `'TCPIP::hh346999::hislip1::INSTR'`.

Usage: Query only

Firmware/Software: V3.0.10

Manual operation: See "Visa Resource (all protocol types)" on page 93

SYSTem:COMMunicate:SOCKet<i>:MODE <Mode>

Sets the protocol operation mode for direct socket communication.

Suffix:
<i> 1..32
Selects the remote channel.

Parameters:

<Mode> RAW | AGILent | IEEE1174

RAW: no support of control messages
AGILent: emulation codes via control connection (control port)
IEEE1174: emulation codes via data connection (data port)

*RST: A *RST has no effect on this parameter.

Example: `SYSTem:COMMunicate:SOCKet:MODE RAW`
Set the operation mode to raw.

Firmware/Software: V1.0.5.3

Manual operation: See "Protocol Mode (TCPIP only)" on page 94

SYSTem:COMMunicate:SOCKet<i>:PORT <Port>

Sets the data port number for direct socket communication.

Suffix:

<i> 1..32
Selects the remote channel.

Parameters:

<Port> Range: 0 to 32767
*RST: A *RST has no effect on the value.

Example: `SYSTem:COMMunicate:SOCKet:PORT 1025`
Set the port number to 1025 (channel 1).

Firmware/Software: V1.0.5.3

Manual operation: See "Data Port (TCPIP only)" on page 93

SYSTem:COMMunicate:SOCKet<i>:VRESource?

Queries the VISA resource string for direct socket communication.

Suffix:

<i> 1..32
Selects the remote channel.

Return values:

<String> VISA address string

*RST: *RST has no effect on the value.

Example:
```
SYSTem:COMMunicate:NET:HOSTname
'hh346999.domain.net'
SYSTem:COMMunicate:SOCKet2:PORT 1500
SYSTem:COMMunicate:SOCKet2:VRESource?
```
Define a host name, set a port and query the VISA resource string of the socket resource (channel 2). The response is `'TCPIP::hh346999.domain.net::1500::SOCKET'`.

Usage: Query only

Firmware/Software: V1.0.5.3

Manual operation: See "Visa Resource (all protocol types)" on page 93

SYSTem:COMMunicate:VXI<i>:GTR <State>

Enables or disables the VXI-11 interface.

Suffix:
<i> 1..32
Selects the remote channel.

Parameters:
<State> ON | OFF | 0 | 1

ON | 1: VXI-11 enabled
OFF | 0: VXI-11 disabled

*RST: A *RST has no effect on the value.

Example: `SYSTem:COMMunicate:VXI:GTR ON`
Enable VXI-11 interface for channel 1.

Firmware/Software: V1.0.0.4

SYSTem:COMMunicate:VXI<i>:VRESource?

Queries the VISA resource string for the VXI-11 protocol.

Suffix:
<i> 1..32
Selects the remote channel.

Return values:
<String> VISA address string

*RST: *RST has no effect on the value.

Example:
```
SYSTem:COMMunicate:NET:HOSTname 'hh346999'
SYSTem:COMMunicate:VXI2:VRESource?
```
Define a hostname and query the VISA resource string for the VXI-11 protocol (channel 2). The response is `'TCPIP::hh346999::inst1::INSTR'`.

Usage: Query only

Firmware/Software: V1.0.0.4

Manual operation: See "Visa Resource (all protocol types)" on page 93

8.12 Installed Software and Device ID

The following commands provide information about the installed software, hardware, licenses and the device ID.

SYSTem:BASE:OPTion:LIST? [<OptionType>[, <Validity>]]

Returns a list of installed software options (licenses), hardware options, software packages and firmware applications.

The list can be filtered using the described parameters. If filtering results in an empty list, a "0" is returned.

The meaning of the filter `<Validity>` depends on the `<OptionType>` as follows:

- A software option is valid if there is an active license key for it. The value "FUNCtional" is not relevant.
- A hardware option is functional if the corresponding hardware and all its components can be used (no defect detected). The value "VALid" is not relevant.
- A firmware application is functional if the required hardware, software and license keys are available and functional. The value "VALid" is not relevant.
- For software packages, the filter has no effect.

Parameters:

<OptionType> SWOPtion | HWOPtion | SWPackage | FWA | ALL

List only software options, hardware options, software packages or firmware applications. By default or if `ALL` is selected, all types are listed.

<Validity> FUNCtional | VALid | ALL

List only functional entries or only valid entries. By default or if `ALL` is selected, the list is not filtered according to the validity.

Example:

```
SYSTem:BASE:OPTion:LIST?
```
Returns an unfiltered list.
```
SYSTem:BASE:OPTion:LIST? SWOPtion, VALid
```
Returns all valid software options.
```
SYSTem:BASE:OPTion:LIST? FWA, FUNCtional
```
Returns all functional firmware applications.

Usage: Query only

Firmware/Software: V1.0.10.1
V2.0.10: Query parameters <OptionType> and <Validity> added

Manual operation: See "Related Commands" on page 98

SYSTem:BASE:OPTion:VERSion? [<Application>]

Returns version information for installed software packages.

You can either query a list of all installed packages and their versions or you can query the version of a single package specified via <Application>:

- <Application> specified: A string is returned, indicating the version of the <Application>. If the specified <Application> is unknown / not installed, "0" is returned.
- <Application> omitted: A string is returned, containing a list of all installed software packages and their version in the format "<PackageName1>,<Version1>;<PackageName2>,<Version2>;..."

Query parameters:

<Application> String selecting the software package for which the version is queried

Return values:

<SoftwareVersion> String containing a single version or a list of applications and versions

Example: `SYSTem:BASE:OPTion:VERSion?`
Returns a list of all packages, for example `"CMW BASE,V3.5.20;CMW GPRF Gen,V3.5.20;CMW GPRF Meas,V3.5.20"`.

Example: `SYSTem:BASE:OPTion:VERSion? "CMW GPRF Gen"`
Returns the version of the GPRF generator software, for example `"V3.5.20"`.

Example: `SYSTem:BASE:OPTion:VERSion? "nonsense"`
Returns `"0"`.

Usage: Query only

Firmware/Software: V2.1.25

Manual operation: See "Related Commands" on page 98

SYSTem:DEVice:ID?

Queries the device identification of the instrument. This ID is important for ordering licenses.

Return values:

<DeviceID> Device ID string

Usage: Query only

Firmware/Software: V1.0.10.1

SYSTem:CMW:DEVice:ID?

Queries the identification of all connected radio test heads, especially the serial number of each unit.

The returned string is structured as follows:

"Rohde&Schwarz,CMW,<order no.>,{<type>,<serial>}$_{unit\ 1}$, ..., {<type>,<serial>}$_{unit\ 4}$"

Return values:
<DeviceID> String with identification information

Example: Possible result for two connected radio test heads:
`"Rohde&Schwarz,CMW,1201.0002,K03,100403,K03,100417,NAV,NAV,NAV,NAV"`

Usage: Query only

Firmware/Software: V3.5.121

Manual operation: See "Related Commands" on page 98

8.13 Internal Path Correction

The following commands initiate an internal path correction and query the related status information.

INITiate:BASE:IPC

Starts an internal path correction procedure.

Usage: Event

Firmware/Software: V3.5.91

Manual operation: See "Initiating a correction" on page 104

FETCh:BASE:IPC?

Queries whether an internal path correction procedure is running or not.

Return values:
<MeasStatus> OFF | RUN

Usage: Query only

Firmware/Software: V3.5.91

Manual operation: See "IPC" on page 104

FETCh:BASE:IPC:RESult?

Queries information about the internal path correction status. For a successful correction, the returned values are `0,0,<Date>,NAV`.

Return values:
<Reliability> For reliability indicator values, see Chapter 7.4.3.1, "Reliability Indicator", on page 152

<ResultNumber> Number indicating the outcome of the correction, see Table 8-1

<Date> Date of the last correction as string

<ResultText> String with instructions for problem solving, for example, repeat the correction procedure or contact the R&S service

Example: Example of returned values for a successful correction:
`0, 0, "2017-06-08", NAV`

Usage: Query only

Firmware/Software: V3.7.11

Manual operation: See "IPC" on page 104

Table 8-1: <ResultNumber> values

Number	Meaning
0	If the <Reliability> also equals 0: The last correction procedure was completed successfully. The resulting correction data is used. If the <Reliability> is not 0, resolve the problem and repeat the correction procedure.
100	The feature IPC is not supported by your instrument. The feature will be supported after the next calibration by Rohde & Schwarz.
101	No path correction has been performed yet with this computer for the connected radio test head.
102	After the last correction procedure with this computer, the radio test head has been calibrated by Rohde & Schwarz. The correction data is ignored. Repeat the correction procedure.
103, 105	An error occurred during the last correction procedure.

8.14 Reference Frequency Settings

The following commands configure the reference frequency settings and adjust the OCXO.

The settings are also available in the "Misc > Sync" section or in the "Maintenance > Reference Adjustment" section of the "Setup" dialog.

SYSTem:BASE:REFerence:FREQuency<n>:ADVanced:SOURce
<FrequencySource>

Selects the reference frequency source for the connected radio test head number <n>.

Suffix:
<n> 1..4

Parameters:

<FrequencySource> INTernal | EXTernal

INTernal: Internal reference frequency
EXTernal: External reference frequency

Example: `SYST:BASE:REF:FREQ2:ADV:SOUR EXT`
Use an external reference frequency at radio test head number 2.

Firmware/Software: V3.7.14

Manual operation: See "Frequency Source" on page 106

SYSTem:BASE:REFerence:FREQuency:SOURce <FrequencySource>

Selects the reference frequency source to be used.

The setting applies to all connected radio test heads. If you have connected several radio test heads and you want to set different values or you want to query the settings, use `SYSTem:BASE:REFerence:FREQuency<n>:ADVanced:SOURce`.

Parameters:

<FrequencySource> INTernal | EXTernal

INTernal: Internal reference frequency
EXTernal: External reference frequency

Example: `SYST:BASE:REF:FREQ:SOUR INT`
Set all connected radio test heads to their internal reference frequency.

Firmware/Software: V1.0.5.3

Manual operation: See "Frequency Source" on page 106

SENSe:BASE:REFerence:FREQuency:LOCKed?

Queries whether the reference frequency is locked or not.

Return values:

<Locked> 1 | 0

1: The frequency is locked.
0: The frequency is not locked.

Usage: Query only

Firmware/Software: V1.0.5.3

8.15 Trigger Settings

The following commands configure the trigger output connector on the rear panel of the instrument. They also allow you to initiate the generation of a "User Initiated Trigger" signal.

The settings are also available in the "Misc > Trigger" section of the "Setup" dialog.

Contents of this chapter

The commands in this section belong to the R&S CMW100 base system. They are not related to specific firmware applications.

Most of the R&S CMW100 measurement firmware applications provide their own, specific trigger settings. For details, refer to the documentation of the `TRIGger...` subsystems in the firmware application documentation.

TRIGger:BASE:EOUT<n>:CATalog:SOURce?

Lists all trigger source values that can be set using `TRIGger:BASE:EOUT<n>:SOURce`. The returned values depend on the installed options.

Suffix:

<n> 1..4
Selects the radio test head for which the list is queried.

Return values:

<Sourcelist> Comma-separated list of all supported values. Each value is represented as a string.

Example: `TRIGger:BASE:EOUT1:CATalog:SOURce?`
Query the available output trigger signals for the first radio test head.

Usage: Query only

Firmware/Software: V3.7.14

Manual operation: See "TRIG Out" on page 107

TRIGger:BASE:EOUT<n>:SOURce <Source>

Selects the output trigger signal to be routed to the trigger output connector. The available values depend on the installed options. A complete list of all supported values can be retrieved using `TRIGger:BASE:EOUT<n>:CATalog:SOURce?`.

Suffix:

<n> 1..4
Selects the radio test head to be configured.

Parameters:

<Source> Trigger source as string

*RST: 'No Connection'

Example: `TRIGger:BASE:EOUT1:SOURce "GPRF Gen1: Waveform Marker 1"`
Select the waveform marker 1 of the GPRF generator as an output trigger signal for the first radio test head.

Firmware/Software: V3.7.14

Manual operation: See "TRIG Out" on page 107

TRIGger:BASE:UINitiated<n>:EXECute

Initiates the generation of a "User Initiated Trigger" signal.

Suffix:
<n> 1..4
Number of the user-initiated trigger

Example: `TRIGger:BASE:UINitiated2:EXECute`
Generates a trigger pulse for the user-initiated trigger number 2.

Usage: Event

Firmware/Software: V3.0.10

Manual operation: See "User Initiated Trigger" on page 107

8.16 Frequency-Dependent Correction

The commands described in this section administrate and activate/deactivate correction tables for frequency-dependent attenuation/gain. For additional information concerning the usage of the tables, refer to RF Path Settings (Generators) and Connection Control (Measurements).

Subinstruments

If the instrument is split into subinstruments, each correction table is assigned to one subinstrument and valid for this subinstrument only. Thus all commands described in this section are subinstrument specific.

There are several ways to select the subinstrument:

- Without further selection, the subinstrument is determined by the remote channel. Using the remote channel for a certain subinstrument configures the tables for that subinstrument.
- Some commands have the optional parameter <TablePath>. To select subinstrument <n>+1, set <TablePath> to "inst<n>".
- Some commands select a table via the parameter <TableName>. To select subinstrument <n>+1, add the prefix "inst<n>/" to the table name.
  Example: "inst2/mytable" means "mytable" for subinstrument number 3.

There is no copy mechanism for correction tables between subinstruments. However, you can use identical command sequences (even with identical table names but different attenuation values) for all subinstruments, addressed by different remote channels.

Subinstrument 1 uses the same correction table database as an instrument with only a single subinstrument. When an instrument is split into several subinstruments (e.g. via `SYSTem:BASE:DEVice:COUNt 2`), the correction tables created for the instrument are assigned to subinstrument 1, and vice versa when the split is canceled. The correction tables of the other subinstruments are maintained in the background when the split is canceled. So they are still available, when the instrument is split again. The connector and path assignment of correction tables is lost upon a change of the instrument setup.

Storage of correction tables

While the R&S CMW100 application software is active, all correction tables are stored in the RAM for fast access. When the application software is closed (e.g. by pressing the standby key), all correction tables in the RAM are stored to the system drive. When the application software is started, all correction tables on the system drive are loaded into the RAM. The following commands allow you to initiate the transfer of correction tables manually: `CONFigure:BASE:FDCorrection:SAV` and `CONFigure:BASE:FDCorrection:RCL`.

Maximum amount of correction table data

The number of entries in each correction table has an upper limit, depending on the firmware application. In most firmware applications, the upper limit is 1000.

Up to 100 correction tables can be assigned to each subinstrument.

Reset / preset of base settings

Note that a reset or preset of the base settings deactivates all correction tables. A reset or preset of a subinstrument or application does not affect the correction tables.

Commands deactivating correction tables:

- `SYSTem:RESet:BASE`, `SYSTem:PRESet:BASE`
- `SYSTem:RESet:ALL`, `SYSTem:PRESet:ALL`

Commands without impact on correction tables:

- `*RST`
- `SYSTem:RESet`, `SYSTem:PRESet`

List of commands:

CONFigure:BASE:FDCorrection:CTABle:CREate <TableName>{, <Frequency>, <Correction>}...

Creates a correction table for frequency-dependent attenuation and stores it in the RAM. If a table with the given name exists for the addressed subinstrument, it is overwritten.

The parameter pairs <Frequency>, <Correction> are used to fill the table. A command with an incomplete pair (e.g. <Frequency> without <Correction>) is ignored completely. To add entries to an existing table, see `CONFigure:BASE:FDCorrection:CTABle:ADD`.

You can enter parameter pairs in any order. The table entries (pairs) are automatically sorted from lowest to highest frequency.

The supported frequency range depends on the instrument model and the available options. The supported range can be smaller than stated here. See Chapter 1.3, "R&S CMW Models", on page 11.

Setting parameters:

<TableName> String parameter used to identify the table by other commands and to store the table on the system drive. The string must comply to Windows™ file name conventions, see Mass Memory Commands.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.
Example: "inst2/mytable" means "mytable" for subinstrument number 3

<Frequency> Range: 70E+6 Hz to 6E+9 Hz
Increment: 0.1 Hz
Default unit: Hz

<Correction> Range: -50 dB to 90 dB
Increment: 0.01 dB
Default unit: dB

Example: `CONFigure:BASE:FDCorrection:CTABle:CREate 'mytable', 1900000000, 0.5, 2000000000, 0.7`
Create the table `'mytable'` for the subinstrument addressed by the remote channel, with two entries: 0.5 dB at 1900 MHz and 0.7 dB at 2000 MHz

Usage: Event

Firmware/Software: V2.1.25

Manual operation: See "Add Table" on page 110

CONFigure:BASE:FDCorrection:CTABle:ADD <TableName>{, <Frequency>, <Correction>}...

Adds entries to an existing correction table. At least one parameter pair has to be specified. A command with an incomplete pair (e.g. <Frequency> without <Correction>) is ignored completely.

You can add parameter pairs in any order. The table entries (pairs) are automatically sorted from lowest to highest frequency.

The supported frequency range depends on the instrument model and the available options. The supported range can be smaller than stated here. See Chapter 1.3, "R&S CMW Models", on page 11.

Setting parameters:

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?`.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

<Frequency>
Range: 70E+6 Hz to 6E+9 Hz
Increment: 0.1 Hz
Default unit: Hz

<Correction>
Range: -50 dB to 90 dB
Increment: 0.01 dB
Default unit: dB

Example: `CONFigure:BASE:FDCorrection:CTABle:ADD 'mytable', 1925000000, 0.55, 1975000000, 0.65`
Add two entries to the table `'mytable'`: 0.55 dB at 1925 MHz and 0.65 dB at 1975 MHz

Usage: Event

Firmware/Software: V2.1.25

Manual operation: See "Add Entries" on page 112

CONFigure:BASE:FDCorrection:CTABle:ERASe <TableName>, <Frequency>...

Removes one or more selected entries from a correction table. Each table entry consists of a frequency value and a correction value. Entries to be removed are selected via their frequency values.

The supported frequency range depends on the instrument model and the available options. The supported range can be smaller than stated here. See Chapter 1.3, "R&S CMW Models", on page 11.

Setting parameters:

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?`.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

<Frequency> Selects the table entry to be removed. The value must match the frequency of an existing table entry. To remove several entries, specify a comma-separated list of frequencies.

Range: 70E+6 Hz to 6E+9 Hz
Increment: 0.1 Hz
Default unit: Hz

Example: `CONFigure:BASE:FDCorrection:CTABle:ERASe 'mytable', 1925000000, 1975000000`
The two entries with the frequencies 1925 MHz and 1975 MHz are removed from the table `'mytable'`.

Usage: Event

Firmware/Software: V2.1.27

Manual operation: See "Delete Entries" on page 112

CONFigure:BASE:FDCorrection:CTABle:EXISt? <TableName>

Queries whether a correction table with the specified name exists or not.

Query parameters:

<TableName> Table name as string parameter
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

Return values:

<Exists> 0 | 1

0: table does not exist
1: table exists

Example: `CONF:BASE:FDC:CTABle:EXISt? 'mytable'`
Check whether a correction table with the name `'mytable'` exists or not.

Usage: Query only

Firmware/Software: V3.2.20

CONFigure:BASE:FDCorrection:CTABle:DELete <TableName>

Deletes a correction table from the RAM and the system drive.

Setting parameters:

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?`.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

Example: `CONF:BASE:FDC:CTABle:DELete 'mytable'`

Usage: Event

Firmware/Software: V1.0.5.3

Manual operation: See "Delete Table(s)" on page 111

CONFigure:BASE:FDCorrection:CTABle:DELete:ALL [<TablePath>]

Deletes all correction tables for a selected subinstrument from the RAM and the system drive.

Setting parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Example: `CONF:BASE:FDC:CTABle:DELete:ALL`
Delete tables for the subinstrument addressed by the remote channel.

Example: `CONF:BASE:FDC:CTABle:DELete:ALL "inst2"`
Delete tables for subinstrument 3 (identified in VISA resource strings as inst2).

Usage: Event

Firmware/Software: V2.1.27, V3.7.14 added <TablePath>

Manual operation: See "Delete Table(s)" on page 111

CONFigure:BASE:FDCorrection:CTABle:COUNt? [<TablePath>]

Returns the number of correction tables currently stored on the system drive for a selected subinstrument.

Query parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Return values:
<TableCount> Number of tables

Example: `CONF:BASE:FDCorrection:CTABle:COUNt?`
Queries <TableCount> for the subinstrument addressed by the remote channel.

Example: `CONF:BASE:FDCorrection:CTABle:COUNt? "inst2"`
Queries <TableCount> for subinstrument 3 (identified in VISA resource strings as inst2).

Usage: Query only

Firmware/Software: V2.1.27, V3.7.14 added <TablePath>

Manual operation: See "Select Table" on page 109

CONFigure:BASE:FDCorrection:CTABle:CATalog? [<TablePath>]

Returns the names of the correction tables currently stored on the system drive for a selected subinstrument.

Query parameters:
<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Return values:
<TableName> Comma-separated list of table names as strings

Example: `CONF:BASE:FDCorrection:CTABle:CATalog?`
Returns the table names for the subinstrument addressed by the remote channel, for example:
`'mytable','setup 1','setup 3G'.`

Example: `CONF:BASE:FDCorrection:CTABle:CATalog? "inst2"`
Returns the table names for subinstrument 3 (identified in VISA resource strings as inst2).

Usage: Query only

Firmware/Software: V1.0.5.3, V3.7.14 added <TablePath>

Manual operation: See "Select Table" on page 109

CONFigure:BASE:FDCorrection:CTABle:DETails? <TableName>[, <StartIndex>[, <Count>]]

Returns the entries of a correction table.

Query parameters:

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?`.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

<StartIndex> Index number of the first entry to be listed. The first entry of a table has index number 0.
Default: 0

<Count> Maximum number of entries to be listed. By default, all entries from <StartIndex> to the end of the table are listed.

Return values:

<ValuePairs> Table entry pairs. Each pair consists of a frequency in Hz and an attenuation in dB.

Example: `CONF:BASE:FDC:CTABle:DETails? 'mytable', 3, 1`
Returns the entry with index number 3, e.g.: 2000000000,0.7

Usage: Query only

Firmware/Software: V1.0.5.3

Manual operation: See "Entries" on page 109

CONFigure:BASE:FDCorrection:CTABle:LENGth? <TableName>

Returns the number of entries (i.e. pairs of frequency and attenuation) of a correction table.

Query parameters:

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?` on page 239.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

Return values:

<TableLength> Number of table entries

Example: `CONF:BASE:FDC:CTABle:LENGth? 'mytable'`
For `'mytable'` containing four frequency/attenuation pairs, the result is `4`.

Usage: Query only

Firmware/Software: V1.0.5.3

Manual operation: See "Table Info" on page 109

CONFigure:BASE:FDCorrection:SAV [<TablePath>]

Saves the correction tables for a selected subinstrument from the RAM to the system drive.

This action is performed automatically when the R&S CMW100 application software is closed, for example, by pressing the standby key. However, you can use the command to save your work manually after creating or configuring correction tables.

Setting parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Example: `CONF:BASE:FDCorrection:SAV`
Save tables for the subinstrument addressed by the remote channel.

Example: `CONF:BASE:FDCorrection:SAV "inst2"`
Save tables for subinstrument 3 (identified in VISA resource strings as inst2).

Usage: Event

Firmware/Software: V2.1.25, V3.7.14 added <TablePath>

Manual operation: See "Save / Recall Changes, Recall Tables" on page 111

CONFigure:BASE:FDCorrection:RCL

Loads all correction tables for a selected subinstrument from the system drive into the RAM.

This action is performed automatically when the R&S CMW100 application software is started. However, you can use the command to retrieve the correction tables after the disk contents have been modified. Or you can use it to undo changes and fall back to the tables stored on the system drive.

Setting parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Example: `CONF:BASE:FDCorrection:RCL`
Load tables for the subinstrument addressed by the remote channel.

Example: `CONF:BASE:FDCorrection:RCL "inst2"`
Load tables for subinstrument 3 (identified in VISA resource strings as inst2).

Usage: Event

Firmware/Software: V2.1.25, V3.7.14 added <TablePath>

CONFigure:FDCorrection:ACTivate <Connector>, <TableName>[, <Direction>[, <RFConverter>]]

Activates a correction table for the TX AUX connector.

Setting parameters:

<Connector> R11O | R21O | R31O | R41O
Selects the RF connector

<TableName> String parameter identifying the table. To display a list of existing tables, use the command `CONFigure:BASE:FDCorrection:CTABle:CATalog?`.
You can add the prefix "inst<n>/" to address subinstrument number <n>+1.

<Direction> RXTX | TX
Can be omitted. Only for backward compatibility to existing scripts.

<RFConverter> RF1
Can be omitted. Only for backward compatibility to existing scripts.

Example: `CONFigure:FDCorrection:ACTivate R11O, 'mytable_out'`
Activates the table 'mytable_out' for the TX AUX connector.

Usage: Event

Firmware/Software: V3.5.20

Manual operation: See "Select Mapping" on page 109

CONFigure:FDCorrection:DEACtivate <Connector>[, <Direction>[, <RFConverter>]]

Deactivates any correction tables for the TX AUX connector.

Setting parameters:

<Connector> R11O | R21O | R31O | R41O
Selects the RF connector

<Direction> RXTX | TX
Can be omitted. Only for backward compatibility to existing scripts. Both values have the same effect.

<RFConverter> RF1
Can be omitted. Only for backward compatibility to existing scripts.

Example: `CONFigure:FDCorrection:DEACtivate R11O`
Deactivates all correction tables for the TX AUX connector.

Usage: Event

Firmware/Software: V3.5.20

Manual operation: See "Select Mapping" on page 109

CONFigure:FDCorrection:DEACtivate:ALL [<Direction>[, <TablePath>]]

Deactivates all correction tables for the TX AUX connectors of a selected subinstrument.

Setting parameters:

<Direction> TX

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Usage: Event

Firmware/Software: V3.7.14

Manual operation: See "Select Mapping" on page 109

CONFigure:FDCorrection:USAGe? <Connector>[, <RFConverter>]

Returns which correction table is assigned to the TX AUX connector.

Setting parameters:

<Connector> R11O | R21O | R31O | R41O
Selects the RF connector

<RFConverter> RF1
Can be omitted. Only for backward compatibility to existing scripts.

Return values:

<RXTableName> Empty string

<TXTableName> String identifying the table assigned to the TX direction. If no table is active, an empty string is returned.

Example: `CONFigure:FDCorrection:USAGe? R11O`
Result: `"","mytable_out"`

Usage: Query only

Firmware/Software: V3.5.20

Manual operation: See "Mapping Info" on page 110

CONFigure:CMWS:FDCorrection:ACTivate:TX <Connectors>, <Table1>[, <Table2>, <Table3>, <Table4>, <Table5>, <Table6>, <Table7>, <Table8>]
CONFigure:CMWS:FDCorrection:ACTivate:RX <Connectors>, <Table1>[, <Table2>, <Table3>, <Table4>, <Table5>, <Table6>, <Table7>, <Table8>]

Activates correction tables for the input paths (RX) or output paths (TX) of the connectors RF 1 to RF 8.

You can use the commands in two ways:

- To activate a correction table for a selected RF connector, specify a single connector and a single table name.
- To activate individual correction tables for all RF connectors, specify a table for each connector (8 connectors/tables).

You can add the prefix "inst<n>/" to table names to address subinstrument number <n> +1.

Setting parameters:

<Connectors>	Selects a single connector or all 8 connectors of a radio test head. For possible values, see Chapter 7.4.4.1, "Values for RF Path Selection", on page 157.
<Table1>	String parameter identifying the table for the selected single connector or for the first of the eight connectors (RF 1)
<Table2>	String parameter identifying the table for RF 2
<Table3>	String parameter identifying the table for RF 3
<Table4>	String parameter identifying the table for RF 4
<Table5>	String parameter identifying the table for RF 5
<Table6>	String parameter identifying the table for RF 6
<Table7>	String parameter identifying the table for RF 7
<Table8>	String parameter identifying the table for RF 8
Example:	`CONFigure:CMWS:FDC:ACTivate:RX R11, 'mytable'` Activates `'mytable'` for the input direction of connector RF 1.
Example:	`CONFigure:CMWS:FDCorrection:ACTivate:RX R118, 'table_a', 'tab_green', 'table_x', 'tabno1', 'mytable15', 'path4', 'mytable', 'cable5'` Activates different tables for the input direction of the connectors RF 1 to RF 8.
Usage:	Event
Firmware/Software:	V3.5.20
Manual operation:	See "Select Mapping" on page 109

CONFigure:CMWS:FDCorrection:DEACtivate:TX <Connectors>
CONFigure:CMWS:FDCorrection:DEACtivate:RX <Connectors>

Deactivate correction tables for the input paths (RX) or output paths (TX) of the connectors RF 1 to RF 8.

Setting parameters:

<Connectors> Selects a single connector or all 8 connectors of a radio test head.
For possible values, see Chapter 7.4.4.1, "Values for RF Path Selection", on page 157.

Example: `CONFigure:CMWS:FDCorrection:DEACtivate:RX R11`
Deactivates correction tables for the input direction of connector RF 1.

Example: `CONFigure:CMWS:FDCorrection:DEACtivate:TX R118`
Deactivates correction tables for the output direction of connector RF 1 to RF 8.

Usage: Event

Firmware/Software: V3.5.20

Manual operation: See "Select Mapping" on page 109

CONFigure:CMWS:FDCorrection:DEACtivate:TX:ALL [<TablePath>]
CONFigure:CMWS:FDCorrection:DEACtivate:RX:ALL [<TablePath>]

Deactivate all correction tables for all RF <n> connectors of a selected subinstrument, in input direction (RX) or output direction (TX).

Setting parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Usage: Event

Firmware/Software: V3.7.14

Manual operation: See "Select Mapping" on page 109

CONFigure:CMWS:FDCorrection:DEACtivate:ALL [<TablePath>]

Deactivate all correction tables for all RF <n> connectors of a selected subinstrument.

Setting parameters:

<TablePath> String selecting the subinstrument
If omitted: subinstrument addressed by the remote channel.
"inst<n>": subinstrument <n>+1

Usage: Event

Firmware/Software: V3.7.14

Manual operation: See "Select Mapping" on page 109

CONFigure:CMWS:FDCorrection:USAGe? <Connector>

Lists the correction tables assigned to one of the RF connectors RF 1 to RF 8.

Query parameters:

<Connector> Selects one of the connectors RF 1 to RF 8 of a radio test head. For possible values, see Chapter 7.4.4.1, "Values for RF Path Selection", on page 157.

Return values:

<CorrectionTableRX> String identifying the table assigned to the RX direction. If no table is active, an empty string is returned.

<CorrectionTableTX> String identifying the table assigned to the TX direction. If no table is active, an empty string is returned.

Example: `CONFigure:CMWS:FDCorrection:USAGe? R11`
Result: `"myRXtable",""`
For the RX direction, `"myRXtable"` is assigned. For the TX direction, no correction table is assigned.

Usage: Query only

Firmware/Software: V3.5.20

Manual operation: See "Mapping Info" on page 110

8.17 Multi-CMW Configuration

The following commands help you to identify the radio test heads connected to the control computer. You can identify the physical units and check which logical number is assigned to which unit.

FETCh:BASE:MCMW:SNUMber? <BoxNr>

Queries the serial number of a connected radio test head. The radio test head is selected via its assigned <BoxNr>.

Query parameters:

<BoxNr> BOX1 | BOX2 | BOX3 | BOX4 | BOX5 | BOX6 | BOX7 | BOX8
Selects for which radio test head the serial number is queried (box <n>, also called CMW <n>).
The command is prepared for up to 8 connected radio test heads.

Return values:

<Serialnumber> String containing the serial number of the radio test head with the <BoxNr>.

Example: `FETCh:BASE:MCMW:SNUMber? BOX2`
Returns for example `"100417"`.

Usage: Query only

Firmware/Software: V3.7.25

Manual operation: See "Serial Number" on page 113

STARt:BASE:MCMW:IDENtify <BoxNr>[, <BlinkingTime>]

Lets all LEDs on the front panel of a connected radio test head blink. The radio test head is selected via its assigned <BoxNr>.

Setting parameters:

<BoxNr> BOX1 | BOX2 | BOX3 | BOX4 | BOX5 | BOX6 | BOX7 | BOX8

Selects for which radio test head the blinking is triggered (box <n>, also called CMW <n>).
The command is prepared for up to 8 connected radio test heads.

<BlinkingTime> Duration of the blinking. If you omit this parameter, the default duration is used.

Range: 0 s to 300 s
Default unit: s

Example:
```
STARt:BASE:MCMW:IDENtify BOX2, 10
```
Lets the LEDs of CMW 2 / box 2 blink for 10 seconds.

Usage: Event

Firmware/Software: V3.7.25

Manual operation: See "Identify Box" on page 113

CONFigure:BASE:MCMW:REARrange <BoxNr>...

Rearranges the assignment of box numbers to serial numbers.

Parameters:

<BoxNr> BOX1 | BOX2 | BOX3 | BOX4 | BOX5 | BOX6 | BOX7 | BOX8

Comma-separated list of box numbers
Enter the new sequence of box numbers to be assigned to the old sequence of serial numbers.
The command is prepared for up to 8 connected radio test heads.

Example:
```
CONF:BASE:MCMW:REARrange BOX4, BOX1, BOX2, BOX3
```
Rearranges the numbering as listed in the following table.

Usage: Event

Firmware/Software: V3.7.80

Manual operation: See "Choose New Number, Rearrange CMWs" on page 113

Table 8-2: Example

CMW number	Old serial number	New serial number
1	100345	100753
2	100510	100345
3	100749	100510
4	100753	100749

8.18 Subinstruments

The following commands configure/cancel the split of the instrument into subinstruments.

SYSTem:BASE:DEVice:MSCont?

Returns the maximum number of subinstruments into which the instrument can be split.

This command returns the maximum number without smart channel mode. With smart channel mode, see `SYSTem:BASE:DEVice:MSCCount?` on page 248.

Return values:
<MaxSICount> Maximum number of subinstruments

Usage: Query only

Firmware/Software: V3.0.10

Manual operation: See "Select Type" on page 120

SYSTem:BASE:DEVice:MSCCount?

Returns the maximum number of subinstruments into which the instrument can be split in smart channel mode.

Without smart channel mode, see `SYSTem:BASE:DEVice:MSCont?` on page 248.

Return values:
<MaxSCcount> The value 0 indicates that the smart channel mode is not available or no split is possible.
Range: 0 to 32

Usage: Query only

Firmware/Software: V3.7.14

Manual operation: See "Select Type" on page 120

SYSTem:BASE:DEVice:COUNt <Count>

Splits the instrument into subinstruments or assigns all hardware resources to a single subinstrument.

Send this command to the subinstrument with the lowest number (device number 0 / assigned instrument 1 / subinstrument 1).

To assign/distribute the available hardware resources to the subinstruments, enter `SYSTem:BASE:DEVice:RESet` after you have changed the number of subinstruments.

Parameters:

<Count> Number of subinstruments
The allowed values depend on your instrument configuration.
*RST: n/a (factory default is 1)

Example:
```
SYSTem:BASE:DEVice:COUNt 2
SYSTem:BASE:DEVice:RESet
```
Split the instrument into two subinstruments and distribute the available resources to these subinstruments.

Firmware/Software: V1.0.5.3

Manual operation: See "Select Type" on page 120

SYSTem:BASE:DEVice:RESet

Assigns the available hardware resources to the subinstruments.

Send this command to the subinstrument with the lowest number (device number 0 / assigned instrument 1 / subinstrument 1).

After changing the number of subinstruments via `SYSTem:BASE:DEVice:COUNt`, always send this command.

Usage: Event

Firmware/Software: V1.0.5.3

Manual operation: See "Select Type" on page 120

SYSTem:BASE:DEVice:SUBinst?

Queries the number of the addressed subinstrument and the total number of subinstruments.

Return values:

<CurSubInst> Number of the addressed subinstrument, as indicated in a VISA resource string for VXI-11
Value n means instrument n+1.
Example: 0 means instrument 1.

<SubInstCount> Total number of subinstruments into which the instrument is split

Example: `SYSTem:BASE:DEVice:SUBinst?`
If there are four subinstruments, possible results are:
`0,4 | 1,4 | 2,4 | 3,4`

Usage: Query only

Firmware/Software: V2.0.10

Manual operation: See "Select Type" on page 120

8.19 Status Reporting System

The STATus subsystem controls the SCPI-defined status reporting structures. For a description of the status registers, see Chapter 7.5, "Status Reporting System", on page 162.

The command description is structured as follows:

8.19.1 General STATus Commands

The following commands preset the status registers and query the error queue.

STATus:PRESet

Configures the status reporting system such that device-dependent events are not reported at a higher level.

The command affects only the transition filter registers, the ENABle registers, and queue enabling:

- The ENABle parts of the STATus:OPERation and STATus:QUEStionable... registers are set to all 0's.
- The PTRansition parts are set all 1's, the NTRansition parts are set to all 0's, so that only positive transitions in the CONDition part are recognized.

The status reporting system is also affected by other commands, see Reset Values of the Status Reporting System.

Example: `STAT:PRES`
Preset the status registers.

Usage: Event

Firmware/Software: V1.0.0.4

STATus:QUEue[:NEXT]?

Queries and at the same time deletes the oldest entry in the error queue. Operation is identical to that of `SYSTem:ERRor[:NEXT]?`.

The entry consists of an error number and a short description of the error. Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.

Example: `STAT:QUE?`
Query the oldest entry in the error queue. 0, "No error" is returned if the error queue is empty.

Usage: Query only

Firmware/Software: V1.0.0.4

8.19.2 STATus:OPERation (Elementary Commands)

The STATus:OPERation subsystem controls the status reporting structures of the STATus:OPERation register, see Chapter 7.5.3.4, "STATus:OPERation", on page 168.

The commands listed below control all levels of the STATus:OPERation register hierarchy. They require the knowledge of the register hierarchy at bit level. For additional commands, refer to the following sections:

- To control the registers based on register names instead of bit numbers, see Chapter 8.19.3, "STATus:OPERation (Extended Commands)", on page 259.
- For comfortable overall evaluation of the registers, see Chapter 8.19.4, "STATus:OPERation (Overall Evaluation)", on page 266.

The lowest three levels of the STATus:OPERation register hierarchy depend on the installed firmware applications. The syntax description of the related commands uses the variables listed in the following table.

Supported variable values
The following table lists all variable values supported by the software. Depending on your instrument model and the installed software, not all values are relevant.

Table 8-3: Variables in STATus:OPERation commands

Variable	Description
<netw_std>	network standard: **AUDio | BLUetooth | CDMA | DATA | EVDO | GPRF| GSM | LTE | TDSCdma | WCDMa | WLAN**

<func_grp>	function group: **MEASurement<i> | SIGNaling<i> | GENerator<i>** with instance <i> = 1 | 2 | 3 | 4
<appl>	application: for GENerator: **UNIVersal | ANALog | DIGital** for MEASurement: **ANALog | DIGital | EPSensor | FFTSanalyzer | IPERf | IQRecorder | IQVSlot | MEValuation | OLTR | PING | POWer | PRACh | SPECtrum | SRS | THRoughput | TPC** for SIGNaling: **EBLer | BER | BERCswitched | BERPswitched | BLER | EHICh | HACK | PER | TDATa | THRoughput | ULLogging**

Example: For the GSM multi-evaluation measurement, the variable combination is `GSM:MEASurement:MEValuation`. Command example: `STATus:OPERation:TASK:A:GSM:MEASurement:MEValuation:ENABle?`.

All commands related to the highest level (STATus:OPERation) are SCPI-confirmed. Note that *RST does not influence the status registers (see also Chapter 7.5.5, "Reset Values of the Status Reporting System", on page 175).

SYSTem:HELP:STATus:BITS?

Returns a list of paths for the bits of the STATus:OPERation registers at the lowest level of the hierarchy.

Each path is represented by a string containing all registers from highest to lowest level separated by colons.

Example: `"STATus:OPERation:TASK:A:GPRF:MEASurement:POWer:OFF"`

Usage: Query only

Firmware/Software: V2.0.10

SYSTem:HELP:STATus[:REGister]?

Returns a list of paths for the STATus:OPERation registers.

Each path is represented by a string containing all registers from highest level down to the individual register, separated by colons.

For the GPRF power measurement for example the following paths are listed: `"STATus:OPERation", "STATus:OPERation:TASK", "STATus:OPERation:TASK:A", "STATus:OPERation:TASK:A:GPRF", "STATus:OPERation:TASK:A:GPRF:MEASurement", "STATus:OPERation:TASK:A:GPRF:MEASurement:POWer"`

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:CONDition?
STATus:OPERation:TASK:CONDition?
STATus:OPERation:TASK:A:CONDition?
STATus:OPERation:TASK:A:<netw_std>:CONDition?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:CONDition?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:CONDition?

Returns the contents of the CONDition part of the status register, see Structure of a SCPI Status Register. Reading the CONDition registers is nondestructive.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Return values:
<ConditionBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:TASK:A:GPRF:GEN:UNIV:COND?`
Query the CONDition part of the GPRF generator status register to check the current generator state.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:OPERation:ENABle <MaskBits>
STATus:OPERation:TASK:ENABle <MaskBits>
STATus:OPERation:TASK:A:ENABle <MaskBits>
STATus:OPERation:TASK:A:<netw_std>:ENABle <MaskBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:ENABle <MaskBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:ENABle <MaskBits>

Sets the enable mask which allows true conditions in the EVENt part of the status register to be reported to the next higher level in the summary bit. If a bit is 1 in the enable register and the associated event bit changes to true, a positive transition occurs in the summary bit. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<MaskBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:TASK:A:GPRF:MEAS:ENAB 1536`
Set bits no. 9 and 10 of the ENABle part of the status register for GPRF measurements (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.4.11

STATus:OPERation:ESRQ <MaskBits>
STATus:OPERation:TASK:ESRQ <MaskBits>
STATus:OPERation:TASK:A:ESRQ <MaskBits>

STATus:OPERation:TASK:A:<netw_std>:ESRQ <MaskBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:ESRQ <MaskBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:ESRQ <MaskBits>

Sets the enable mask for the register to the specified value. Also sets the relevant bit in the enable mask of all higher registers up to the STATus:OPERation register and the SRE. Thus the entire reporting path from the register up to the SRE is enabled so that an SRQ can be generated.

If the enable mask is set to 0, the higher registers are not modified.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<MaskBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:TASK:A:GPRF:MEAS:POW:ESRQ 4`
Sets bit no. 2 (decimal 4) and disables all other bits in the ENABle part of the following status register:
STAT:OPER:TASK:A:GPRF:MEAS:POW
Also sets the relevant bit in the ENABle part of the following status registers, without changing the other bits of these ENABle parts:
STAT:OPER:TASK:A:GPRF:MEAS: set bit no. 0
STAT:OPER:TASK:A:GPRF: set bit no. 0
STAT:OPER:TASK:A: set bit no. 1
STAT:OPER:TASK: set bit no. 0
STAT:OPER: set bit no. 9
SRE: set bit no. 7

Firmware/Software: V2.0.10

STATus:OPERation[:EVENt]?
STATus:OPERation:TASK[:EVENt]?
STATus:OPERation:TASK:A[:EVENt]?
STATus:OPERation:TASK:A:<netw_std>[:EVENt]?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>[:EVENt]?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>[:EVENt]?

Returns the contents of the EVENt part of the status register. Reading an EVENt part clears it. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Return values:
<EventBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:TASK:A:GPRF:MEAS:POW?`
Query the EVENt part of the status register for the GPRF power measurement to check whether an event has occurred since the last reading.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:OPERation:NTRansition <FilterBits>
STATus:OPERation:TASK:NTRansition <FilterBits>
STATus:OPERation:TASK:A:NTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:NTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:NTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:NTRansition <FilterBits>

Sets the negative transition filter. If a bit is set, a 1 to 0 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<FilterBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:TASK:A:GPRF:NTR 1536`
Set bits no. 9 and 10 of the NTRansition part of the GPRF status register (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.4.11

STATus:OPERation:PTRansition <FilterBits>
STATus:OPERation:TASK:PTRansition <FilterBits>
STATus:OPERation:TASK:A:PTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:PTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:PTRansition <FilterBits>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:PTRansition <FilterBits>

Sets the positive transition filter. If a bit is set, a 0 to 1 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<FilterBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:OPER:PTR 1536`
Set bits no. 9 and 10 of the PTRansition part for STATus:OPERation register (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.4.11

STATus:OPERation:WCONdition? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:WCONdition? <WaitBits>[, <Timeout>]

STATus:OPERation:TASK:A:WCONdition? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:WCONdition? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:WCONdition? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:WCONdition? <WaitBits>[, <Timeout>]

Waits until at least one bit is set to true in the CONDition part of the status register, that has also been set in the <WaitBits>. In other words, waits until an AND operation between the bit patterns yields a positive result.

An optional timeout can be defined. When the AND operation yields a positive result or the timeout is reached, the command returns the result of the AND operation.

The command can be used, for example, to wait until a measurement has been finished before querying the measurement results.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Query parameters:

<WaitBits> Range: 0 to 65535 (decimal representation)

<Timeout> Timeout in ms

Return values:

<Result> Result of the AND operation. A 0 indicates that a timeout occurred.
Range: 0 to 65535 (decimal representation)

Example:

```
INIT:GPRF:MEAS:POW
STAT:OPER:TASK:A:GPRF:MEAS:POW:WCON? 9,1000
```

Initiate a GPRF power measurement and wait until it has reached the state OFF or RDY. OFF is indicated by bit number 0 (decimal 1) and RDY by bit number 3 (decimal 8). The decimal sum equals 9. The timeout is set to 1000 ms.
The returned value is `0`, `1` or `8` (timeout occurred, state OFF reached or state RDY reached).

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:WEVent? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:WEVent? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:WEVent? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:WEVent? <WaitBits>[, <Timeout>]

STATus:OPERation:TASK:A:<netw_std>:<func_grp>:WEVent? <WaitBits>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:WEVent? <WaitBits>[, <Timeout>]

Waits until at least one bit is set in the EVENt part of the status register, that has also been set in the <WaitBits>. In other words, waits until an AND operation between the bit patterns yields a positive result.

An optional timeout can be defined. When the AND operation yields a positive result or the timeout is reached, the command returns the result of the AND operation. The bits corresponding to this result are cleared in the EVENt part.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Query parameters:

<WaitBits> Range: 0 to 65535 (decimal representation)

<Timeout> Timeout in ms

Return values:

<Result> Result of the AND operation. A 0 indicates that a timeout occurred.

Range: 0 to 65535 (decimal representation)

Example:

```
STAT:OPER:TASK:A:GPRF:MEAS:POW:PTR 0
STAT:OPER:TASK:A:GPRF:MEAS:POW:NTR 65535
```

Configure the transition registers so that the EVENt part of the status register reports only transitions from 1 to 0.

```
INIT:GPRF:MEAS:POW
STAT:OPER:TASK:A:GPRF:MEAS:POW:WEV? 4,1000
```

Initiate a GPRF power measurement and wait until it has left the state RUN (bit number 2, decimal 4). The timeout is set to 1000 ms.
The returned value is 0 or 4 (timeout occurred or state RUN left).

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:BIT<n>:CONDition?
STATus:OPERation:BIT<n>[:EVENt]?

Returns bit no. <n> of the CONDition or EVENt part of the STATus:OPERation register, see also Structure of a SCPI Status Register. To return the entire parts, see `STATus:OPERation:CONDition?` and `STATus:OPERation[:EVENt]?`.

Suffix:

<n> 8 to 12
Number of the bit

Return values:

<BitValue> 0 | 1

Example: `STAT:OPER:BIT9:COND?`
Query bit no. 9 of the CONDition part of the STATus:OPERation register.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:OPERation:BIT<n>:ENABle <BitValue>
STATus:OPERation:BIT<n>:NTRansition <BitValue>
STATus:OPERation:BIT<n>:PTRansition <BitValue>

Sets bit no. <n> of the ENABle, NTRansition or PTRansition part of the STATus:OPERation register, see also Structure of a SCPI Status Register. To set the entire parts, see `STATus:OPERation:ENABle`, `STATus:OPERation:NTRansition` and `STATus:OPERation:PTRansition`.

Suffix:
<n> 8 to 12
Number of the bit

Parameters:
<BitValue> 0 | 1

Example: `STAT:OPER:BIT9:ENAB 1`
Set bit no. 9 of the ENABle part of the STATus:OPERation register.

Firmware/Software: V1.0.4.11

8.19.3 STATus:OPERation (Extended Commands)

The commands listed in this section serve the same purpose as the elementary commands, see Chapter 8.19.2, "STATus:OPERation (Elementary Commands)", on page 251.

However, the extended commands provide more comfort by mapping bit values to the corresponding status register mnemonics whenever possible. Thus they allow you to control the STATus:OPERation register hierarchy without knowing it by heart at bit level.

For a complete overview of the mnemonics used in the register hierarchy, see STATus:OPERation and `SYSTem:HELP:STATus[:REGister]?`, `SYSTem:HELP:STATus:BITS?`.

Settings

Assume that you want to set the enable mask for the states OFF and RDY of the GPRF power measurement 1, up to the highest level of the hierarchy. You can study the hierarchy and set the correct bits using the following elementary commands:

```
STATus:OPERation:TASK:A:GPRF:MEAS1:POWer:ENABle 9
STATus:OPERation:TASK:A:GPRF:MEAS1:ENABle 1
```

```
STATus:OPERation:TASK:A:GPRF:ENABle 1
STATus:OPERation:TASK:A:ENABle 2
STATus:OPERation:TASK:ENABle 1
STATus:OPERation:ENABle 512
```

Or you use the following extended commands. Knowing the first command is sufficient to write down all subsequent commands correctly. The first (lowest level) command reflects the entire register path up to the highest level and contains all mnemonics:

```
STATus:OPERation:TASK:A:GPRF:MEAS1:POWer:XENable (OFF,RDY)
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (POWer)
STATus:OPERation:TASK:A:GPRF:XENable (MEAS1)
STATus:OPERation:TASK:A:XENable (GPRF)
STATus:OPERation:TASK:XENable (A)
STATus:OPERation:XENable (TASK)
```

Extended commands accept also decimal numbers in addition to mnemonics. If you use a mixture of numbers and mnemonics, both the decimal numbers and the mnemonics are internally translated into bits and the sum of all bits is set. Example: To set the enable mask for the GPRF measurements POWer (bit 0), IQRecorder (bit 3) and FFTSanalyzer (bit 4), you can use one the following commands. All commands yield the same result.

```
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (POWer,FFTSanalyzer,IQRecorder)
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (POWer,FFTSanalyzer,8)
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (POWer,FFTSanalyzer,24)
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (POWer,24)
STATus:OPERation:TASK:A:GPRF:MEAS1:XENable (25)
```

Queries

A query using an extended command returns a list of mnemonics, i.e. the returned bit pattern is translated into the corresponding mnemonics. Bits set to true that cannot be translated to a mnemonic are summed up and listed as an additional decimal number. If no bit at all is set, empty brackets are returned.

Example: Assume that the bits number 1, 2, 10 and 12 have been set for the ENABle part of register STATus:OPERation:TASK:A:GPRF:POW. Bit 1 and 2 can be translated into QUED and RUN, but bits 10 and 12 have no mnemonics. These two bits are presented as $2^{10}+2^{12}$ = 1024+4096 = 5120.

The query `STATus:OPERation:TASK:A:GPRF:MEAS:POW:XENable?` returns `(QUE,RUN,5120)`.

Example: Assume that no event has occurred since the last query. The event register equals 0. No bit is set to true.

The query `STATus:OPERation:TASK:A:GPRF:MEAS:POW:XEVent?` returns `()`.

STATus:OPERation:XCONdition?
STATus:OPERation:TASK:XCONdition?
STATus:OPERation:TASK:A:XCONdition?
STATus:OPERation:TASK:A:<netw_std>:XCONdition?

STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XCONdition?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XCONdition?

Returns the contents of the CONDition part of the status register, see Structure of a SCPI Status Register. Reading the CONDition registers is nondestructive.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Return values:

<List> Comma-separated list of mnemonics (and/or decimal numbers between 0 and 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:GEN:UNIV:XCON?`
Query the CONDition part of the status register for the GPRF generator. The result is `(OFF)` or `(PEND)` or `(ON)`.

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:XENable <List>
STATus:OPERation:TASK:XENable <List>
STATus:OPERation:TASK:A:XENable <List>
STATus:OPERation:TASK:A:<netw_std>:XENable <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XENable <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XENable <List>

Sets the enable mask which allows true conditions in the EVENt part of the status register to be reported to the next higher level in the summary bit. If a bit is 1 in the enable register and its associated event bit changes to true, a positive transition occurs in the summary bit. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:

<List> Comma-separated list of mnemonics (and/or decimal numbers between 0 and 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:MEAS:XEN (POW,IQR)`
Set the enable mask bits corresponding to the measurements POWer (bit 0) and IQRecorder (bit 3) to true.

Firmware/Software: V2.0.10

STATus:OPERation:XESRq <List>
STATus:OPERation:TASK:XESRq <List>
STATus:OPERation:TASK:A:XESRq <List>
STATus:OPERation:TASK:A:<netw_std>:XESRq <List>

STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XESRq <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XESRq <List>

Sets the enable mask for the register according to the specified list. Also sets the relevant bit in the enable mask of all higher registers up to the STATus:OPERation register and the SRE. Thus the entire reporting path from the register up to the SRE is enabled so that an SRQ can be generated.

If an empty list is specified (set enable mask to 0), the higher registers are not modified.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:

<List> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:MEAS:POW:XESRq (RUN)`
Sets the RUN bit (bit 2) and disables all other bits in the ENABle part of the following status register:
STAT:OPER:TASK:A:GPRF:MEAS:POW
Also sets the relevant bit in the ENABle part of the following status registers, without changing the other bits of these ENABle parts:
STAT:OPER:TASK:A:GPRF:MEAS: set POW bit
STAT:OPER:TASK:A:GPRF: set MEAS1 bit
STAT:OPER:TASK:A: set GPRF bit
STAT:OPER:TASK: set A bit
STAT:OPER: set TASK bit
SRE: set OPER bit

Firmware/Software: V2.0.10

STATus:OPERation:XEVent?
STATus:OPERation:TASK:XEVent?
STATus:OPERation:TASK:A:XEVent?
STATus:OPERation:TASK:A:<netw_std>:XEVent?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XEVent?
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XEVent?

Returns the contents of the EVENt part of the status register. Reading an EVENt part clears it. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Return values:

<List> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:MEAS:POW:XEV?`
Query the EVENt part of the status register for the GPRF power measurement to check whether an event has occurred since the last reading.

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:XNTRansition <List>
STATus:OPERation:TASK:XNTRansition <List>
STATus:OPERation:TASK:A:XNTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:XNTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XNTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XNTRansition <List>

Sets the negative transition filter. If a bit is set, a 1 to 0 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<List> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:MEAS1:POW:XNTR (OFF,RDY)`
Set the negative transition filter bits corresponding to the GPRF power measurement states OFF (bit 0) and RDY (bit 3) to 1.

Firmware/Software: V2.0.10

STATus:OPERation:XPTRansition <List>
STATus:OPERation:TASK:XPTRansition <List>
STATus:OPERation:TASK:A:XPTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:XPTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XPTRansition <List>
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XPTRansition <List>

Sets the positive transition filter. If a bit is set, a 0 to 1 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Parameters:
<List> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

Example: `STAT:OPER:TASK:A:GPRF:MEAS1:POW:XPTR (OFF,RDY)`
Set the positive transition filter bits corresponding to the GPRF power measurement states OFF (bit 0) and RDY (bit 3) to 1.

Firmware/Software: V2.0.10

STATus:OPERation:XWCondition? <Wait>[, <Timeout>]
STATus:OPERation:TASK:XWCondition? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:XWCondition? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:XWCondition? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XWCondition? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XWCondition? <Wait>[, <Timeout>]

Waits until at least one bit is set to true in the CONDition part of the status register and in the bit pattern corresponding to the <Wait> list. In other words, waits until an AND operation between the bit patterns yields a positive result.

An optional timeout can be defined. When the AND operation yields a positive result or the timeout is reached, the command returns the result of the AND operation.

The command can be used, for example, to wait until a measurement has been finished before querying the measurement results.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Query parameters:

<Wait> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

<Timeout> Timeout in ms

Return values:

<Result> Result of the AND operation, as a comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets. Empty brackets indicate that a timeout occurred.

Example:

```
INIT:GPRF:MEAS:POW
STAT:OPER:TASK:A:GPRF:MEAS:POW:XWC?(RDY),1000
```

Initiate a GPRF power measurement and wait until it has reached the state RDY. The timeout is set to 1000 ms.
The returned value is `()` or `RDY` (timeout occurred or state RDY reached).

Usage: Query only

Firmware/Software: V2.0.10

STATus:OPERation:XWEVent? <Wait>[, <Timeout>]
STATus:OPERation:TASK:XWEVent? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:XWEVent? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:XWEVent? <Wait>[, <Timeout>]

STATus:OPERation:TASK:A:<netw_std>:<func_grp>:XWEVent? <Wait>[, <Timeout>]
STATus:OPERation:TASK:A:<netw_std>:<func_grp>:<appl>:XWEVent? <Wait>[, <Timeout>]

Waits until at least one bit is set to true in the EVENt part of the status register and in the bit pattern corresponding to the <Wait> list. In other words, waits until an AND operation between the bit patterns yields a positive result.

An optional timeout can be defined. When the AND operation yields a positive result or the timeout is reached, the command returns the result of the AND operation. The bits corresponding to this result are cleared in the EVENt part.

For a description of the variables <netw_std>, <func_grp> and <appl>, refer to Table 8-3.

Query parameters:

<Wait> Comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets

<Timeout> Timeout in ms

Return values:

<Result> Result of the AND operation, as a comma-separated list of mnemonics (and/or decimal numbers in the range 0 to 65535) enclosed in brackets. Empty brackets indicate that a timeout occurred.

Example:

```
STAT:OPER:TASK:A:GPRF:MEAS:POW:XPTR ()
STAT:OPER:TASK:A:GPRF:MEAS:POW:XNTR (RUN)
```

Configure the transition registers so that the EVENt part of the status register reports only transitions from 1 to 0 for the bit corresponding to the measurement state RUN.

```
INIT:GPRF:MEAS:POW
STAT:OPER:TASK:A:GPRF:MEAS:POW:XWEV? (RUN),1000
```

Initiate a GPRF power measurement and wait until it has left the state RUN. The timeout is set to 1000 ms.
The returned value is `()` or `(RUN)` (timeout occurred or state RUN left).

Usage: Query only

Firmware/Software: V2.0.10

8.19.4 STATus:OPERation (Overall Evaluation)

Information about the current state of tasks and state transitions of tasks can be derived via an evaluation of the STATus:OPERation register hierarchy. But you have to query the registers one by one and interpret the decimal representation of bit values.

The commands listed below offer a much more comfortable way to evaluate the STATus:OPERation register hierarchy. They allow you to query the current states or state transitions of all tasks. You can even display all measurement or generator tasks being

in a certain state. And you can use the command `STATus:EVENt:BITS:NEXT?` within programs reacting on state transitions.

Most commands return a single string or a comma-separated list of strings. Each string is composed of the complete path of the status register plus the state. Example of a result list (with additional line breaks for better readability):

```
"STAT:OPER:TASK:A:GPRF:MEAS1:POW:RUN",
"STAT:OPER:TASK:A:GPRF:MEAS1:IQVS:QUED",
"STAT:OPER:TASK:A:GPRF:MEAS1:EPS:OFF",
"STAT:OPER:TASK:A:GPRF:MEAS1:IQR:OFF",
"STAT:OPER:TASK:A:GPRF:GEN1:UNIV:ON"
```

The available commands are listed below.

STATus:CONDition:BITS:ALL? [<RegExp>]

This command offers a comfortable way to get an overview of all task states, without querying each register individually.

It evaluates the CONDition parts of the lowest level OPERation status registers. The result consists of a comma-separated list of strings. Each string indicates the state of one task and is composed of the complete path of the status register plus the state. The command is nondestructive.

In most situations, the returned list shows all task states of the installed firmware applications. However it can happen that a task is not listed if currently no resources at all are assigned to that task (e.g. directly after installation). In that case, you could say that the state of the task is less than "OFF".

Query parameters:

<RegExp> String, optional regular expression filtering the returned results, see also Regular Expressions.

Example: `STATus:CONDition:BITS:ALL? '(POW)|(IQV)'`
List the current task states of the installed firmware applications. Limit the results to strings containing 'POW' (GPRF power measurement) or 'IQV' (GPRF IQ vs slot measurement).

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:CONDition:BITS:COUNt? [<RegExp>]

Returns the number of task states listed by `STATus:CONDition:BITS:ALL?`.

Query parameters:

<RegExp> String, optional regular expression filtering the task states before they are counted, see also Regular Expressions.

Example: `STATus:CONDition:BITS:COUNt?`
List the number of task states returned by `STATus:CONDition:BITS:ALL?`.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:CONDition:BITS:CATaloge? [<RegExp>]

Returns a list of all possible task states for the installed firmware applications. The current task states returned by `STATus:CONDition:BITS:ALL?` form a subset of the list returned by this command.

Query parameters:

<RegExp> String, optional regular expression filtering the returned results, see also Regular Expressions.

Example: `STATus:CONDition:BITS:CATaloge?`
List all possible task states of the installed firmware applications.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:EVENt:BITS:ALL? [<RegExp>]

Evaluates the EVENt parts of all lowest level OPERation status registers. The result consists of a comma-separated list of strings. Each string is composed of the complete path of the status register plus the state. The command is nondestructive.

This command offers a comfortable way to get an overview of the EVENt parts of all lowest level registers, without querying each register individually.

Query parameters:

<RegExp> String, optional regular expression filtering the returned results, see also Regular Expressions.

Example: `STATus:EVENt:BITS:ALL? '[RDY]$'`
List the EVENt parts of the lowest level status registers. Limit the results to strings ending with 'RDY'.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:EVENt:BITS:CLEar [<RegExp>]

Clears the EVENt parts of all status registers of the STATus:OPERation register hierarchy. If a regular expression is defined, the command is only applied to the registers matching the filter criteria.

Query parameters:

<RegExp> String, optional regular expression. The EVENt part is only cleared for registers matching the regular expression. See also Regular Expressions.

Example: `STAT:EVENt:BITS:CLEar '![IQR]'`
Clear all status registers except the IQ recorder registers (clear if string does not contain 'IQR').

Usage: Event

Firmware/Software: V1.0.4.11

STATus:EVENt:BITS:COUNt? [<RegExp>]

Returns the number of events listed by `STATus:EVENt:BITS:ALL?`.

Query parameters:

<RegExp> String, optional regular expression filtering the events before they are counted, see also Regular Expressions.

Example: `STATus:EVENt:BITS:COUNt?`
List the number of events returned by `STATus:EVENt:BITS:ALL?`.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:EVENt:BITS:NEXT? [<RegExp>]

Searches, returns and deletes the next event at the lowest level of the STATus:OPERation register hierarchy. An entry consists of a string composed of the complete path of the status register reporting the event and the state.

This command can be used to supply state transitions to a remote control program one by one. The program can then react on the transitions, e.g. fetch the results of a measurement that reached the RDY or SDR state. Or start a new measurement after a measurement has been finished.

A list of all events in the STATus:OPERation register hierarchy can be returned using `STATus:EVENt:BITS:ALL?`.

Query parameters:

<RegExp> String, optional regular expression. Events not matching the regular expression are ignored when searching for the next event. See also Regular Expressions.

Example: `STATus:EVENt:BITS:NEXT?`
Query and delete the next event.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:GENerator:CONDition:OFF? [<RegExp>]
STATus:GENerator:CONDition:PENDing? [<RegExp>]
STATus:GENerator:CONDition:ON? [<RegExp>]
STATus:MEASurement:CONDition:OFF? [<RegExp>]
STATus:MEASurement:CONDition:QUED? [<RegExp>]
STATus:MEASurement:CONDition:RDY? [<RegExp>]
STATus:MEASurement:CONDition:RUN? [<RegExp>]
STATus:MEASurement:CONDition:SDReached? [<RegExp>]

Lists all generator tasks or measurement tasks whose current state equals the state indicated by the last mnemonic.

The results are collected from the CONDition parts of the lowest level registers of the STATus:OPERation register hierarchy. They are returned as a comma-separated list of strings. Each string is composed of the complete path of the status register plus the current state.

Query parameters:

<RegExp> String, optional regular expression filtering the returned results, see also Regular Expressions.

Example: `STATus:MEASurement:CONDition:RDY? 'GPRF'`
List all tasks with current state "Ready". Limit the results to strings containing 'GPRF'.

Usage: Query only

Firmware/Software: V1.0.4.11

8.19.5 STATus:QUEStionable

The STATus:QUEStionable subsystem controls the status reporting structures of the STATus:QUEStionable register, see STATus:QUEStionable.

Unless otherwise stated, all following commands are SCPI-confirmed. Note that *RST does not influence the status registers (see also Reset Values of the Status Reporting System).

The available commands are listed below.

STATus:QUEStionable:CONDition?

Returns the contents of the CONDition part of the status register. Reading the CONDition registers is nondestructive. See also Structure of a SCPI Status Register.

Return values:
<ConditionBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:QUES:COND?`
Query the CONDition part of the QUEStionable register to check for questionable instrument states.

Usage: Query only

Firmware/Software: V1.0.0.4

STATus:QUEStionable:ENABle <MaskBits>

Sets the enable mask which allows true conditions in the EVENt part of the status register to be reported to the next higher level in the summary bit. If a bit is 1 in the enable register and its associated event bit changes to true, a positive transition occurs in the summary bit. See also Structure of a SCPI Status Register.

Parameters:
<MaskBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:QUES:ENAB 1536`
Set bits no. 9 and 10 of the QUEStionable:ENABle register (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.0.4

STATus:QUEStionable[:EVENt]?

Returns the contents of the EVENt part of the status register. Reading an EVENt register clears it. See also Structure of a SCPI Status Register.

Return values:
<EventBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:QUES?`
Query the EVENt part of the QUEStionable register to check whether an event has occurred since the last reading.

Usage: Query only

Firmware/Software: V1.0.0.4

STATus:QUEStionable:NTRansition <FilterBits>

Sets the negative transition filter. If a bit is set, a 1 to 0 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

Parameters:
<FilterBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:QUES:NTR 1536`
Set bits no. 9 and 10 of the QUEStionable:NTRansition register (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.0.4

STATus:QUEStionable:PTRansition <FilterBits>

Sets the positive transition filter. If a bit is set, a 0 to 1 transition in the corresponding bit of the condition register writes a 1 to the corresponding bit of the event register. See also Structure of a SCPI Status Register.

Parameters:
<FilterBits> Range: 0 to 65535 (decimal representation)

Example: `STAT:QUES:PTR 1536`
Set bits no. 9 and 10 of the QUEStionable:PTRansition register (1536 = 512 + 1024 = 2^9 + 2^{10}).

Firmware/Software: V1.0.0.4

STATus:QUEStionable:BIT<n>:CONDition?
STATus:QUEStionable:BIT<n>[:EVENt]?

Returns bit no. <n> of the CONDition or EVENt part of the STATus:QUEStionable register, see Structure of a SCPI Status Register. To return the entire parts, see `STATus:QUEStionable:CONDition?` and `STATus:QUEStionable[:EVENt]?`.

Suffix:
<n> 8 to 12
Number of the bit

Return values:
<BitValue> 0 | 1

Example: STAT:QUES:BIT9:COND?
Query bit no. 9 of the CONDition part of the STATus:QUEStionable register.

Usage: Query only

Firmware/Software: V1.0.4.11

STATus:QUEStionable:BIT<n>:ENABle <BitValue>
STATus:QUEStionable:BIT<n>:NTRansition <BitValue>
STATus:QUEStionable:BIT<n>:PTRansition <BitValue>

Sets bit no. <n> of the ENABle, NTRansition or PTRansition part of the STATus:QUEStionable register, see also Structure of a SCPI Status Register. To set the entire parts, see STATus:QUEStionable:ENABle, STATus:QUEStionable:NTRansition and STATus:QUEStionable:PTRansition.

Suffix:
<n> 8 to 12
Number of the bit

Parameters:
<BitValue> 0 | 1

Example: STAT:QUES:OPER:BIT9:ENAB 1
Set bit no. 9 of the ENABle part of the STATus:QUEStionable register.

Firmware/Software: V1.0.4.11

8.20 System Date and Time

The following commands configure or query the date and time settings of the operating system. Note that *RST does not affect these settings.

SYSTem:DATE[:UTC] <Year>, <Month>, <Day>

Sets the UTC date of the operating system calendar.

Parameters:
<Year> Range: four-digit number
<Month> Range: 1 to 12

<Day> Range: 1 to n (depending on the <Month>)

Example: `SYSTem:DATE:UTC?`
Query the UTC date. Possible response: 2012, 09, 19

Firmware/Software: V3.0.12

SYSTem:DATE:LOCal <Year>, <Month>, <Day>

Sets the local date of the operating system calendar.

Parameters:
<Year> Range: four-digit number

<Month> Range: 1 to 12

<Day> Range: 1 to n (depending on the <Month>)

Example: `SYSTem:DATE:LOCal?`
Query the local date. Possible response: 2012, 09, 20

Firmware/Software: V3.0.12

Manual operation: See "Date" on page 91

SYSTem:TIME[:UTC] <Hour>, <Minute>, <Second>

Sets the universal time coordinated (UTC) of the operating system clock.

Parameters:
<Hour> Range: 0 to 23

<Minute> Range: 0 to 59

<Second> Range: 0 to 59

Example: `SYSTem:TIME:UTC?`
Query the UTC. Possible response: 13, 09, 20

Firmware/Software: V3.0.12

SYSTem:TIME:LOCal <Hour>, <Minute>, <Second>

Sets the local time of the operating system clock.

Parameters:
<Hour> Range: 0 to 23

<Minute> Range: 0 to 59

<Second> Range: 0 to 59

Example: `SYSTem:TIME:LOCal?`
Query the local time. Possible response: 15, 09, 20

Firmware/Software: V3.0.12

Manual operation: See "Time" on page 91

SYSTem:TIME:DSTime:MODE

Configures whether the operating system automatically adjusts its clock for daylight saving time (DST) or not.

The rules defining when exactly the clock must be adjusted by which offset depend on the configured time zone, see `SYSTem:TIME:DSTime:RULE` on page 275.

If the automatism is disabled, the local time is calculated as:

Local time = UTC + time zone offset (no DST offset)

Parameters:

<Enable> **1**: automatism enabled
0: automatism disabled

Example: `SYSTem:TIME:DSTime:MODE 1`
The clock is automatically adjusted.

Firmware/Software: V3.0.12
V3.0.14: support of setting (before: query only)

Manual operation: See "Consider Daylight Saving Time" on page 91

SYSTem:TIME:DSTime:RULE:CATalog?

Returns all time zone values that can be set via `SYSTem:TIME:DSTime:RULE` on page 275.

Return values:

<TimeZoneList> Comma-separated list of all supported values. Each value is represented as a string.

Usage: Query only

Firmware/Software: V3.0.14

Manual operation: See "Time Zone" on page 91

SYSTem:TIME:DSTime:RULE <TimeZone>

Sets the time zone in the date and time settings of the operating system.

The used daylight saving time (DST) rules depend on the configured time zone. So this setting influences the automatic adjustment of the local time and date for DST. See also `SYSTem:TIME:DSTime:MODE` on page 275.

Modifying the time zone modifies also the configured time zone offset, see `SYSTem:TZONe` on page 276.

Parameters:

<TimeZone> Time zone as string
To query a list of all supported strings, use `SYSTem:TIME:DSTime:RULE:CATalog?`.

Example: `SYSTem:TIME:DSTime:RULE "W. Europe Standard Time"`
The time zone is set and the related DST rule set is used.

Firmware/Software: V3.0.14

Manual operation: See "Time Zone" on page 91

SYSTem:TZONe <Hour>, <Minute>

Specifies the offset of the local time to the universal time coordinated (UTC) due to the time zone. There can be an additional offset due to daylight saving time (DST).

Changing the time zone (offset) does not affect an eventual DST offset or the time zone configured via `SYSTem:TIME:DSTime:RULE` on page 275.

The local time is calculated as: *local time = UTC + time zone offset + DST offset*

Parameters:

<Hour> Range: -12 to 15

<Minute> Range: -59 to 59

Example: `SYSTem:TZONe?`
Query the time zone information. Possible response: -3, -30
Meaning: local time = UTC - 3 hours - 30 minutes + DST offset

Firmware/Software: V3.0.12

8.21 Hardcopy Commands

The following HCOPy commands allow you to create screenshots of the "CMW" window. The formats BMP, JPEG and PNG are supported. You can save screenshots into files or return them as block data.

HCOPy:AREA <Area>

Selects the part of the screen to be captured.

Parameters:

<Area> AWINdow | MWINdow | FSCReen

AWINdow: active window
MWINdow: main window
FSCReen: full screen

*RST: AWIN

Firmware/Software: V3.5.130

Manual operation: See "Print Area" on page 114

HCOPy:DEVice:FORMat <Format>

Specifies the format of screenshots created via the commands `HCOPy:FILE`, `HCOPy:DATA?`, `HCOPy:INTerior:FILE` or `HCOPy:INTerior:DATA?`.

Parameters:

<Format> BMP | JPG | PNG

BMP: Windows bitmap format
JPG: JPEG format
PNG: PNG format

Example: `HCOPy:DEVice:FORMat PNG`
Screenshots are created in PNG format.

Firmware/Software: V3.0.10

Manual operation: See "File Type" on page 114

HCOPy:INTerior:FILE <FileName>
HCOPy:FILE <FileName>

Captures a screenshot and stores it to the specified file.

`HCOPy:FILE` captures the entire window, `HCOPy:INTerior:FILE` only the interior of the window.

If a "Remote" dialog is displayed instead of the normal display contents, this command switches on the display before taking a screenshot, and afterwards off again.

Parameters:

<FileName> String parameter specifying the absolute path and name of the file. The file name extension is added automatically according to the configured format (see `HCOPy:DEVice:FORMat` on page 277).
Aliases are allowed (see `MMEMory:ALIases?` on page 209).
Wildcards are not allowed.

Example: `HCOPy:FILE '@PRINT\image1'`

Usage: Event

Firmware/Software: V3.0.10
V3.2.20: `INTerior` command

Manual operation: See "Save" on page 115

HCOPy:INTerior:DATA?
HCOPy:DATA?

Captures a screenshot and returns the result in block data format, see also Block Data Format.

`HCOPy:DATA` captures the entire window, `HCOPy:INTerior:DATA` only the interior of the window.

It is recommended to "switch on" the display before sending this command, see `SYSTem:DISPlay:UPDate` on page 283.

Return values:

<Data> <dblock>
Screenshot in 488.2 block data format

Usage: Query only

Firmware/Software: V3.0.10
V3.2.20: `INTerior` command

Manual operation: See "Save" on page 115

8.22 Miscellaneous Commands

SYSTem:UPDate:DGRoup <DeviceGroup>

Sets the "Device Group" that the instrument belongs to. For remote installation, this setting must match the corresponding setting in the R&S Software Distributor options.

Parameters:

<DeviceGroup> Device group as string

Example: `SYSTem:UPDate:DGRoup "DEFAULT"`
Set the device group to `"DEFAULT"`.

Firmware/Software: V1.0.0.4

Manual operation: See "Software Update" on page 89

SYSTem:BASE:DISPlay:FONTset <FontSize>

Selects the font size for the GUI labels.

Parameters:
<FontSize> DEF | LRG

DEF: Small fonts
LRG: Large fonts

*RST: DEF

Example: `SYSTem:BASE:DISPlay:FONTset LRG`
Select large fonts.

Firmware/Software: V1.0.5.3

Manual operation: See "Font Size, Color Set" on page 88

SYSTem:DISPlay:MONitor <Enable>

Turns the built-in display / the external monitor on or off.

Parameters:
<Enable> ON | OFF

Example: `SYSTem:DISPlay:MONitor ON`
Turn on the monitor.

Usage: Event

Firmware/Software: V3.2.20

Manual operation: See "Turn Off Monitor" on page 88

SYSTem:DISPlay:ROLLkeymode <Mode>

Selects the tree traversal mode of the rotary knob.

Parameters:
<Mode> ZIGZag | CURSors | VERTical

ZIGZag: always zigzag
CURSors: [⇦] / [⇨] keys change to "zigzag" mode, [⇧] / [⇩] keys change to "vertical" mode.
VERTical: always vertical

Example: `SYSTem:DISPlay:ROLLkeymode ZIGZag`
Use zigzag movement.

Firmware/Software: V3.2.10

Manual operation: See "Tree Navigation Mode" on page 88

SYSTem:STARtup:PREPare:FDEFault <Enable>

Enables startup with factory default state.

Parameters:

<Enable> 0 | 1

Behavior during the startup:
0: Restore previous settings.
1: Set instrument to its factory default state.

Example: `SYSTem:STARtup:PREPare:FDEFault 1`
Set factory default during the startup.

Firmware/Software: V3.2.20

Manual operation: See "Startup" on page 89

SYSTem:BASE:DISPlay:MWINdow <Status>

Enables or disables the multiple-window mode of the graphical user interface.

Parameters:

<Status> 0 | 1

1: multiple-window mode
0: single-window mode

Example: `SYSTem:BASE:DISPlay:MWINdow 1`
Enables the multiple-window mode.

Firmware/Software: V3.5.110

Manual operation: See "Multiple Window" on page 90

SYSTem:BASE:STICon:ENABle <OnOff>

Selects whether an icon for the CMW software is added to the system tray of the operating system.

Parameters:

<OnOff> **ON | 1**: icon in system tray
OFF | 0: no icon in system tray

Example: `SYSTem:BASE:STICon:ENABle ON`
Adds an icon to the system tray.

Firmware/Software: V3.7.110

Manual operation: See "Show System Tray Icon" on page 90

SYSTem:BASE:STICon:CLOSe

Hides all windows and taskbar entries of the CMW application.

Prerequisite: A CMW software icon has been added to the system tray (`ENABle` command).

Usage: Event

Firmware/Software: V3.7.110

Manual operation: See "Show System Tray Icon" on page 90

SYSTem:BASE:STICon:OPEN

Restores the windows and taskbar entries of the CMW application after they have been hidden by the `CLOSe` command.

Prerequisite: A CMW software icon has been added to the system tray (`ENABle` command).

Usage: Event

Firmware/Software: V3.7.110

Manual operation: See "Show System Tray Icon" on page 90

SYSTem:CONNector:TRANslation? <Connector>

Queries the relation between physical and virtual connector names. You can query this relation for either a physical or a virtual name. As a result, both physical and virtual name are returned.

For background information and possible connector values, see Chapter 7.4.4.1, "Values for RF Path Selection", on page 157.

Query parameters:

<Connector> Physical or virtual connector name to be queried

Return values:

<VirtualConnector> Returned virtual connector name

<AbsoluteConnector> Returned physical connector name

Example: `SYSTem:CONNector:TRANslation? RA18`
Returns e.g. `RA18,R218`
`SYSTem:CONNector:TRANslation? RA5`
Returns e.g. `RA5,R25`

Usage: Query only

Firmware/Software: V2.0.10

FORMat:BASE[:DATA] <DataType>[, <DataLength>]

Selects the format for numeric data transferred to and from the R&S CMW100, for example query results.

Parameters:

<DataType> ASCii | REAL | BINary | HEXadecimal | OCTal

ASCii
Numeric data is transferred as ASCII bytes. Floating point numbers are transferred in scientific E notation.

REAL
Numeric data is transferred in a definite length block as IEEE floating point numbers, see Block Data Format.

BINary | HEXadecimal | OCTal
Numeric data is transferred in binary, hexadecimal or octal format.

*RST: ASC

<DataLength> The meaning depends on the <DataType> as listed below. A zero returned by a query means that the default value is used.

For ASCii
Decimal places of floating point numbers. That means, number of "b" digits in the scientific notation `a.bbbbbbE+ccc`.
Default: six decimal places

For REAL
Length of floating point numbers in bits:
32 bits = 4 bytes, format `#14...`
64 bits = 8 bytes, format `#18...`
Default: 64 bits

For BINary, HEXadecimal, OCTal
Minimum number of digits. If the number is longer, more digits are used. If it is shorter, leading zeros are added.
Default: 0, no leading zeros

*RST: 0 (six decimal places)

Example: The following examples show the number 123456 in different formats.

```
FORMat:BASE ASC,3       1.235E+005
FORMat:BASE ASC,6       1.234560E+005
FORMat:BASE REAL,32     #14...
FORMat:BASE BIN,20      #B00011110001001000000
FORMat:BASE BIN,2       #B11110001001000000
FORMat:BASE HEX,8       #H0001E240
FORMat:BASE OCT,10      #Q0000361100
```

Usage: SCPI confirmed

Firmware/Software: V1.0.5.3
V3.2.50: <Length> configurable for `ASCii`

SYSTem:BASE:RELiability?

Returns a reliability value indicating errors detected by the base software.

Return values:
<Value> For reliability indicator values, see Chapter 7.4.3.1, "Reliability Indicator", on page 152

Usage: Query only

Firmware/Software: V2.0.10

SYSTem:GENerator:ALL:OFF
SYSTem:MEASurement:ALL:OFF

Switch off all generators or measurements.

Example: `SYSTem:GENerator:ALL:OFF`
Switches off all generators.

Usage: Event

Firmware/Software: V2.1.26

SYSTem:DISPlay:UPDate <Enable>

Defines whether the display is updated or not while the instrument is in the remote state. If the display update is switched off, the normal GUI is replaced by a static image while the instrument is in the remote state.

Switching off the display can speed up the measurement and is the recommended state.

See also Chapter 7.1.4.1, "Using the Display during Remote Control", on page 129

Parameters:
<Enable> **ON | 1:** Display is shown and updated during remote control.
OFF | 0: Display shows static image during remote control.

Example: `SYSTem:DISPlay:UPDate ON`
Switch on the display update.

Firmware/Software: V1.0.5.3

SYSTem:KLOCk <State>

Locks or unlocks the local controls of the instrument, including the (soft-) front panel keys.

Parameters:
<State> ON | OFF
ON: Local key locked (key lock enabled)
OFF: Local keys unlocked
*RST: OFF

Example: `SYSTem:KLOCk ON`
Lock the local keys.

Usage: SCPI confirmed

Firmware/Software: V1.0.0.4

SYSTem:VERSion?

Queries the SCPI version number to which the instrument complies. The instrument complies to the final SCPI version 1999.0.

Usage: Query only
SCPI confirmed

Firmware/Software: V1.0.0.4

SENSe:BASE:TEMPerature:OPERating:INTernal?

Queries the temperature within the instrument. The returned value indicates the average of the temperatures measured at the individual RF modules.

The recommended temperature range is 28 °C to 55 °C.

Return values:
<Temperature> Temperature in degrees
Default unit: °C

Usage: Query only

Firmware/Software: V3.5.61

8.23 List of Commands

Index

Symbols

A

B

C

D

E

F

G

S

T

U

V

W