R&S®WinIQSIM2™
Signal Generation Software
User Manual

1177.5533.02 — 01

Test & Measurement

User Manual

ROHDE&SCHWARZ

This document describes R&S®WinIQSIM2™ and its options.

- R&S®WinIQSIM2-K261
- R&S®WinIQSIM2-K262

This manual version corresponds to software version 3.50.082.xx and later of the R&S®WinIQSIM2™.

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

The following abbreviations are used throughout this manual: R&S®WinIQSIM2™ is abbreviated as R&S WinIQSIM2, R&S®SMW200A is abbreviated as R&S SMW200A, R&S®SGT100A is abbreviated as R&S SGT100A, R&S®SMBV100A is abbreviated as R&S SMBV100A, R&S®SMU200A is abbreviated as R&S SMU200A, R&S®SMJ100A is abbreviated as R&S SMJ100A, R&S®AMU200A is abbreviated as R&S AMU200A, R&S®AFQ100A and R&S®AFQ100B are abbreviated as R&S AFQ100, R&S®EX-IQ-BOX is abbreviated as R&S EX-IQ-BOX, R&S®CMW500 is abbreviated as R&S CMW, R&S®BTC is abbreviated as R&S BTC, R&S®SFU is abbreviated as R&S SFU, R&S.®NPR is abbreviated as R&S NPR.

Contents

1 Preface

The R&S WinIQSIM2 simulation software is a Windows-PC based program that creates digitally modulated signal waveforms. Offering waveform signals in accordance with the definitions in the digital standards or with user-definable characteristics, R&S WinIQSIM2 is an indispensable application for anyone engaged in state-of-the-art digital modulation.

1.1 Key Features

Outstanding features of R&S WinIQSIM2 are:

- Generation of all important digital communication standard signals
- Generation of custom digital signals, like single, multi-carrier CW or multi-segment waveforms
- Generation of noise signals with selectable bandwidth
- Importing I/Q samples for additional filtering or superimposing
- Comprehensive graphic display modes
- Intuitive operation via the block diagram
- Signal transmission to instruments via GPIB, USB, LAN, or file transfer via USB stick
- Direct control of instruments via LAN (remote desktop)

For a detailed specification, refer to the data sheet.

1.2 Related Rohde & Schwarz Instruments

Waveform signals computed by R&S WinIQSIM2 can be loaded onto and processed by the following Rohde & Schwarz Instruments, equipped with an ARB generator, or a waveform memory playback.

Generation of waveforms for ARB generators

- R&S SMW
- R&S SMBV
- R&S SMU
- R&S SMJ
- R&S AMU
- R&S AFQ
- R&S EX-IQ-BOX
  (waveform generation without data transfer)
- R&S BTC
- R&S SFU

- R&S CMW

Control of vector signal generators

The following vector signal generators can be used for generation of radio frequency (RF) signals. Moreover, these vector signal generators can be controlled by R&S WinIQSIM2:

- R&S SMW
- R&S SMBV
- R&S SGT
- R&S SMU
- R&S SMJ

1.3 About the Manual

This user manual describes general functions of R&S WinIQSIM2, the operation of the software and settings common to all firmware options. The settings specific to the firmware options are described in the corresponding user manuals.

The main focus of this manual is on the signal generation capabilities of the software and the tasks required to achieve them. The following topics are included:

- **Welcome to R&S WinIQSIM2**
  Introduction to R&S WinIQSIM2, and getting familiar with the software, including a brief introduction to the features and functions.
- **Getting Started**
  Information on the basic steps for starting up the software.
- **Configuring the Baseband Source and AWGN**
  Descriptions of the individual generation modes, including configuration settings and task descriptions.
- **Setting Up Instruments**
  Descriptions to the configuration steps for connecting to an instrument, signal transmission and activation of signal generation, including configuration settings and task descriptions.
- **File and Data Management**
  Description of general functions to handle data files.
- **General Functions**
  Description of the general settings and functions.
- **Network and Remote Operation**
  Information on setting up the software in a network and operating it remotely.
- **Remote Commands**
  Remote commands required to configure and perform measurements in a remote environment, sorted by tasks
  Remote commands required to set up the environment and to perform common tasks, sorted by tasks
  Programming examples demonstrate the use of many commands and can usually be executed directly for test purposes

- **Troubleshooting and Error Messages**
  Hints and tips on how to handle errors.
- **Appendix**
  Extensive reference information on remote control, etc.
- **List of Commands**
  Alphabetical list of all remote commands described in the manual
- **Index**

1.4 Documentation Overview

This section provides an overview of the R&S WinIQSIM2 user documentation. You find it on the product page at:

http://www.rohde-schwarz.com/product/WinIQSIM2.html > "Downloads"

Online help

Offers quick, context-sensitive access to the complete information for the base unit and the software options directly on the instrument.

User manual

Separate manuals for the base unit and the software options are provided for download:

- Base unit manual
  Contains the description of all instrument modes and functions. It also provides an introduction to remote control, a complete description of the remote control commands with programming examples, and information on maintenance, instrument interfaces and error messages. Includes the contents of the getting started manual.
- Software option manual
  Contains the description of the specific functions of an option. Basic information on operating the R&S WinIQSIM2 is not included.

The **online version** of the user manual provides the complete contents for immediate display on the Internet.

Basic safety instructions

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

Data sheet and brochure

The data sheet contains the technical specifications of the R&S WinIQSIM2. It also lists the options and their order numbers as well as optional accessories.

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

Release notes and open source acknowledgment (OSA)

The release notes list new features, improvements and known issues of the current software version, and describe the software installation.

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

See http://www.rohde-schwarz.com/product/WinIQSIM2.html > "Downloads" > "Software"

Application notes, application cards, white papers, etc.

These documents deal with special applications or background information on particular topics, see http://www.rohde-schwarz.com/appnotes.

1.5 Typographical Conventions

The following text markers are used throughout this documentation:

Convention	Description
"Graphical user interface elements"	All names of graphical user interface elements on the screen, such as dialog boxes, menus, options, buttons, and softkeys are enclosed by quotation marks.
KEYS	Key names are written in capital letters.
`File names, commands, program code`	File names, commands, coding samples and screen output are distinguished by their font.
Input	Input to be entered by the user is displayed in italics.
Links	Links that you can click are displayed in blue font.
"References"	References to other parts of the documentation are enclosed by quotation marks.

1.6 Notes on Screenshots

When describing the functions of the product, we use sample screenshots. These screenshots are meant to illustrate as much as possible of the provided functions and possible interdependencies between parameters. The shown values may not represent realistic test situations.

The screenshots usually show a fully equipped product, that is: with all options installed. Thus, some functions shown in the screenshots may not be available in your particular product configuration.

2 Welcome to the Simulation Software R&S WinIQSIM2

The simulation software R&S WinIQSIM2 enables you to generate waveform files of various digitally modulated signals in accordance with the definitions in the digital standards or with user-definable characteristics.

Waveforms are files with settings provided for repeatable tests with the same test signal. Irrespectively of the way these waveform files are generated, they are always played from an instrument, e.g. the vector signal generator R&S SMW. The signal calculation is performed in advance and the instrument acts as a player.

Features and Functions

In addition to the wide range of digital standards, the functions of R&S WinIQSIM2 also comprise single carrier modulation, the generation of multi carrier signals, WCDMA and third-order signals.

The software calculates I and Q baseband signals on a Windows PC system comprising almost the same functionalities as the Rohde & Schwarz vector signal generators. R&S WinIQSIM2 includes the characteristics of an R&S instrument, if defined. In addition, an internal AWGN noise generator allows you to superimpose noise on the generated signal.

The graphical signal display function visualizes a generated signal in various graphical views for quick evaluation and verification of the signal characteristics.

You can transmit a generated signal directly to a connected instrument in the network over LAN, or via USB or GPIB. Alternatively, you can store it locally in a file and retrieve it subsequently on the instrument. The instrument plays back the loaded waveform file and thus generates the corresponding signal.

Required equipment of an instrument

The following requirements must be met for playing the waveform files...

- ARB (Arbitrary Waveform Generator) or a waveform memory playback
  An instrument must be equipped with an ARB or a waveform memory, allowing the playback of waveform files.
- Digital standard option
  To process waveform file generated by the simulation software R&S WinIQSIM2, the instrument must have installed the corresponding digital standard option for using R&S WinIQSIM2.
  For example, if you purchase "Option SMW-K255", the EUTRA/LTE *Digital Standard option for using R&S WinIQSIM2*, you can process EUTRA/LTE waveform files generated by R&S WinIQSIM2 on the R&S SMW.
- Multicarrier CW Signal generation
  To process multicarrier waveforms, the instrument must be equipped with the corresponding option provided for R&S WinIQSIM2 waveform files.
- AWGN
  Similarly, an instrument must have installed the corresponding R&S WinIQSIM2 option for processing additive white gaussian noise.

See Chapter 1.2, "Related Rohde & Schwarz Instruments", on page 13 for the list of instruments from Rohde & Schwarz that can process waveform files.

For information on the available options, see the data sheet of the instrument.

GUI Appearance and Operation

The graphical user interface of R&S WinIQSIM2 is based on the GUI design of the R&S SMW vector signal generator, and also resembles the user interface of the R&S SMBV, or the R&S SMU signal generator family.

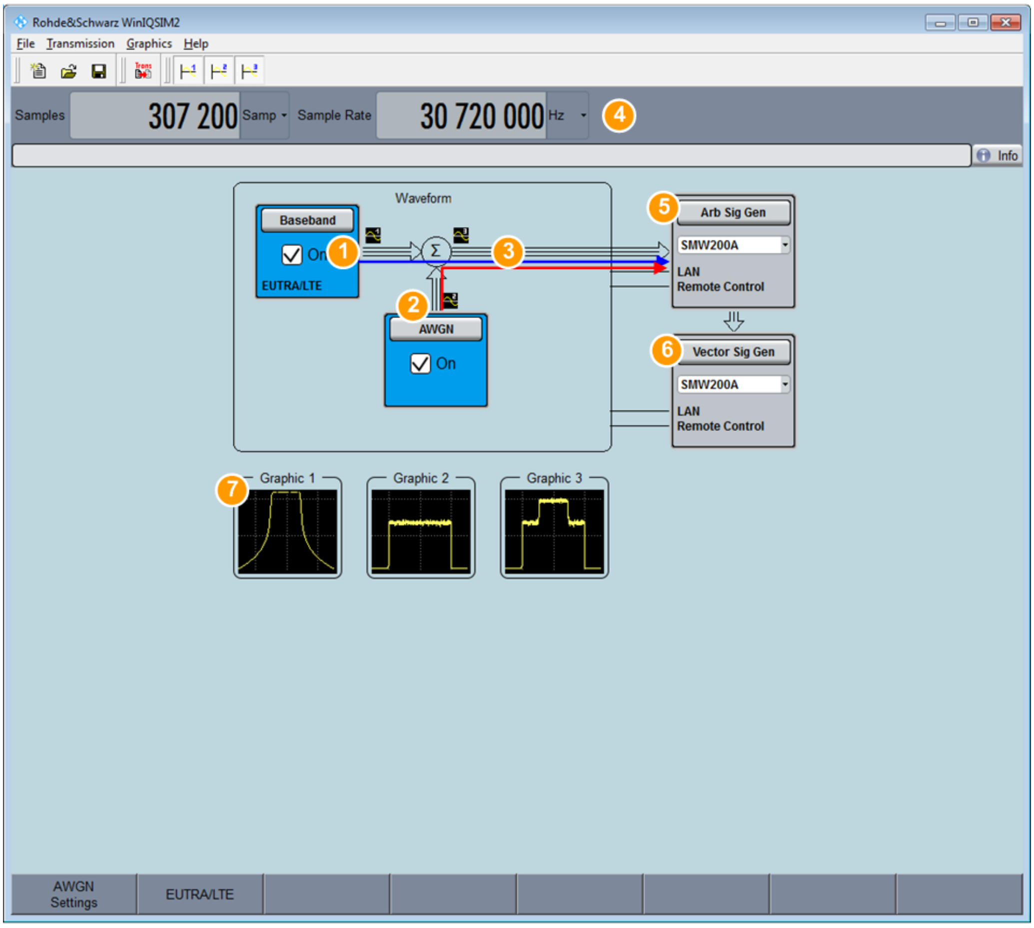


Figure 2-1: R&S WinIQSIM2 application window with signal flow

1 = digital waveform signal
2 = noise signal
3 = superimposed signal
4 = main parameters of the waveform signal
5 = connected arbitrary waveform signal generator
6 = connected vector signal generator
7 = graphical display of the digital, noise and superimposed signals

The application window displays the signal generation in a block diagram. It indicates the current state of the functional blocks and allows you to display the current signal configuration graphically. The signal flow follows the logical left-to-right direction. The status bar above the block diagram shows the main characteristics of the signal. Thus you can see the status of signal configuration and active interfering components at a glance.

Comprehensive graphic display modes in the time and frequency domain, e.g. I(t) and q(t), vector diagram or spectrum, allow simulation and analysis of characteristics as early as in the design phase of new digital communication systems.

You can configure the signal directly in the block diagram. R&S WinIQSIM2 offers intuitive and straight forward operation with a high degree of self-explanation due to the logically structured signal flow, dialogs and menus.

Related Descriptions

The embedded context-sensitive help systems provide the help content related to the element you are currently interacting with. The table of contents, the index and the find functions supports you in finding the information if you are using the documentation in pdf format or printable form. In particular in printed documents, you can find it helpful to use this section and get familiar with the structure of this description.

The description follows the procedure as you likely configure a new signal generation task. It starts with configuring the baseband source, describes how you can assign a noise or interferer signal and validate the signal with the graphics display. Setting up an instrument and transmitting the waveform file to the destination instrument complete the process description.

Finally, the description deals with general functions of R&S WinIQSIM2, information about remote control, like network connection and remote control commands, as well as error messages and troubleshooting.

3 Getting Started

This section describes the basic steps to be taken when starting up R&S WinIQSIM2 the first time.

3.1 Installing the R&S WinIQSIM2 Software

R&S WinIQSIM2 is a stand-alone, PC-based application that creates waveform files of digitally modulated signals.

You find the up-to-date version of R&S WinIQSIM2 and the corresponding release notes describing the improvements and modifications on the product homepage "http://www.rohde-schwarz.com/product/WinIQSIM2.html > Download > Software".

Software and hardware requirements

To install and run R&S WinIQSIM2, the following hardware and software requirements have to be met:

Table 3-1: Software requirements

	Options	Remarks
Operating System	Microsoft® • Windows 8 • Windows 7 • Windows Vista • Windows XP SP2	Any other versions or operating systems are not supported. The installation process checks the operating system and aborts the installation if your PC is not compatible.
VISA runtime library	• R&S VISA, IO libraries for instrument control • National Instruments VISA 4.0. or higher (optional) • other VISA runtime library	VISA driver is bundled with a GPIB (IEC/IEEE)-bus card from National Instruments (NI) or Agilent. If none of these cards are used, there is the option to perform data transmission via TCP/IP connection over LAN. Drivers can be obtained from R&S or NI directly.

Table 3-2: Hardware requirements

	Minimum requirements	Recommended hardware
AMD / Intel CPU	1 GHz	2 GHz
RAM	1 GB **Note:**The installation process checks the RAM size and aborts the installation if the minimum memory size is not given.	2 GB
Display resolution	1024 x 800 pixels	≥ 1024x800 pixels
GPIB (IEC/IEEE)-bus card	• National Instruments • Agilent	optional
Free HD space	600 MB	2 GB

To install the software

To install the software, perform the following:

- Administrator rights are required to install and run the program.
- It is recommended that you uninstall any previous version of R&S WinIQSIM2 before installing the new version (see To uninstall a previous software version).

1. In the windows explorer, execute **WinIQSIM2_v.vv.vvv.vvv.exe**.
2. Follow the instructions of the installation wizard.

   <v.vv.vvv.vvv> stands for the current version.

   The setup file installs all software components R&S WinIQSIM2 requires for operation.

Table 3-3: Default file location (software installation for all users)

File type	File location	File name
User settings and data (Waveforms, Save/ Recall, etc.)	`%APPDATA%\Rohde-Schwarz\winiqsim2`	`*.wv`, `*.savrcl`, etc.
Program data	`%PROGRAMFILES(X86)%\Rohde-Schwarz\WinIQSIM2\xx.x`	`WinIQSIM2_Start.cmd`

To uninstall a previous software version

To uninstall a previous version, perform the following:

1. In the taskbar, select the "Windows Start" button.
2. Select "Settings > Control Panel > Programs > Programs and Features".
3. In the list of programs, select "R&S WinIQSIM2_v.vv.vvv...".

4. Remove the program with "Uninstall/Change > Uninstall".

   The script file identifies and removes any currently installed items of R&S WinIQ-SIM2.

3.2 Starting R&S WinIQSIM2

To access R&S WinIQSIM2 easily, create a shortcut to the `*.exe` file and place it on the desktop of your computer.

Access:

1. On your computer, perform one of the following:
   a) Select "Start > All Programs > R&S WinIQSIM2 > R&S WinIQSIM2 v.vv.vvv.vv"
   b) Double-click the shortcut icon on the desktop.

   After starting up, the main application window appears.

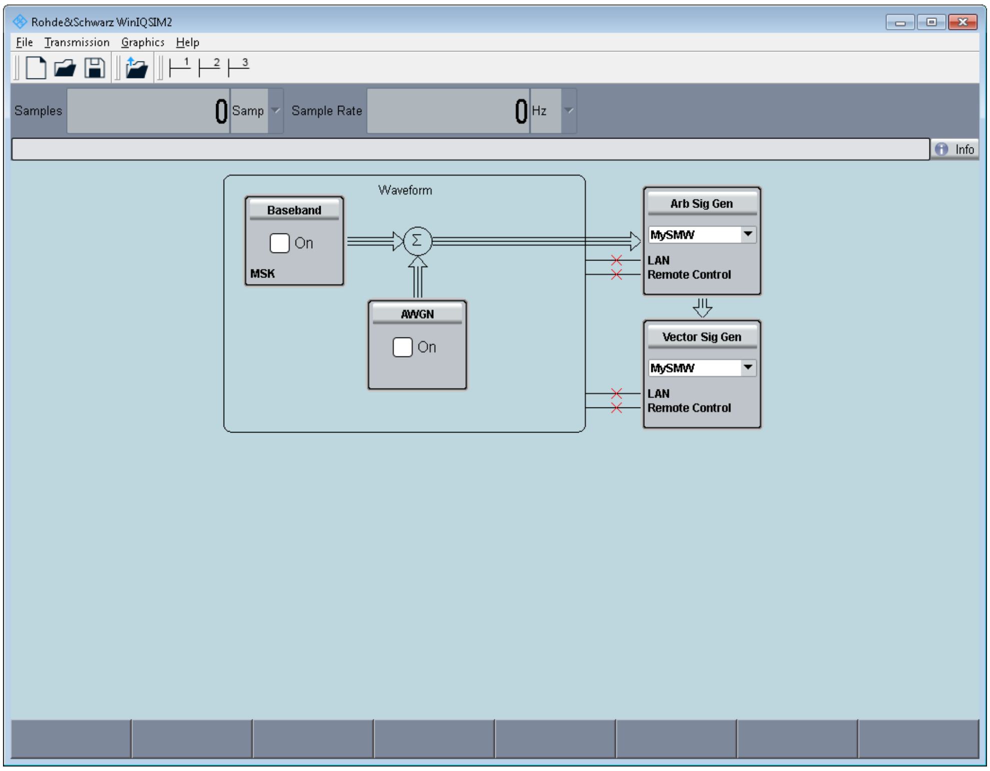


Figure 3-1: R&S WinIQSIM2 main application window in initial state

2. Select "File > New" (or the icon alternatively) to start from an initial state.

3.3 Trying Out R&S WinIQSIM2

The following is an example on how to use R&S WinIQSIM2 to generate a single carrier waveform with AWGN and load it in the ARB of an R&S SMW200A .

The workflow consists of the following steps, each described in a separate step-by-step instruction:

- Chapter 3.3.1, "Configuring the Instrument", on page 24
- Chapter 3.3.2, "Selecting the Instrument", on page 26
- Chapter 3.3.3, "Generating the Waveform Signal", on page 26
- Chapter 3.3.4, "Adding Noise (AWGN) to the Signal", on page 29
- Chapter 3.3.5, "Displaying and Configuring the Graphical Representation of the Waveform Signal", on page 30
- Chapter 3.3.6, "Transmitting the Generated Waveform to the R&S SMW200A", on page 37
- Chapter 3.3.7, "Transmitting the Generated Waveform to a File", on page 39

3.3.1 Configuring the Instrument

Before you can transmit a signal to a signal generator, you must configure the destination instrument and the connection between the instrument and R&S WinIQSIM2.

R&S WinIQSIM2 allows you to scan the LAN, GPIB, or USB interfaces for connected instruments ("Arb Sig Gen or Vector Sig Gen"). Detected instruments are automatically assigned to the list of "Available Instruments", including the associated information on the connection.

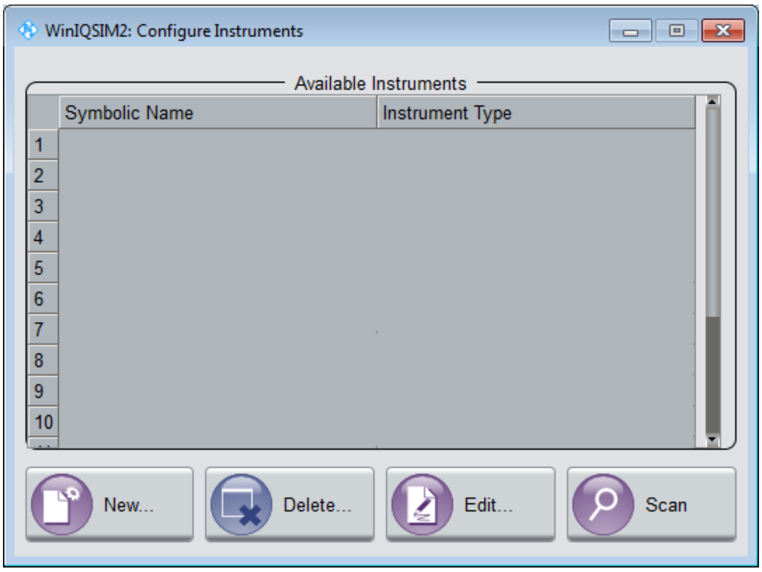
Alternatively, you can create and configure an instrument manually, as described in "To create an instrument manually" on page 196.

The general instrument settings affect various functions, e.g. the maximum size of the waveform file. It is recommended that you configure a connected instrument first before creating the waveform file.

To scan and configure an instrument

1. Connect the instrument to the LAN, e.g. an R&S SMW200A with option R&S SMW-B10.
2. Switch the instrument on.
3. In the menu bar of R&S WinIQSIM2, select "File > New" to start the software from an initial state.
4. In the block diagram, select "Vector Sig Gen > Instruments".

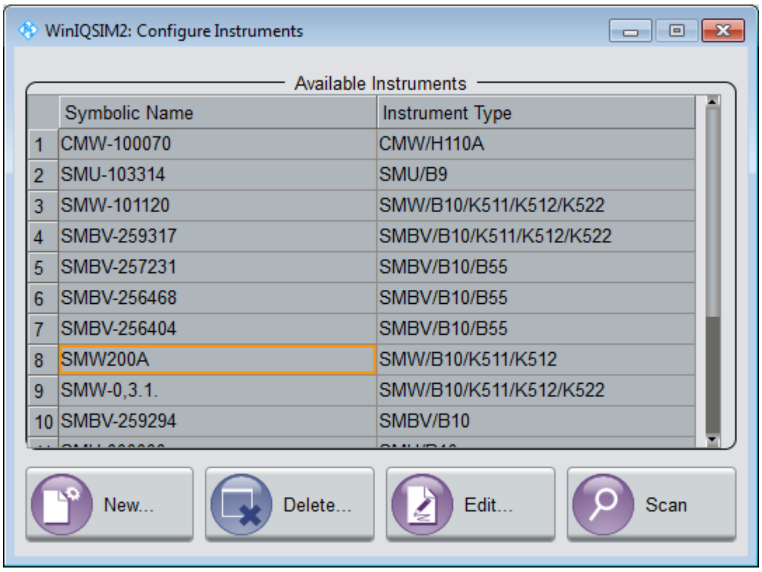
   The "Configure Instruments" dialog opens.



In this dialog, you can search for connected instruments, assign an instrument manually and configure the connection.

5. Select "Scan" to search for connected instruments.

   All instruments found in the network as well as instruments connected via the GPIB or USB interfaces are displayed in the list of "Available Instruments". R&S WinIQ-SIM2 retrieves all information on connection to the instrument automatically, so further configuration is not required. However, you can check and modify the settings with "Edit".



   Tip: R&S WinIQSIM2 does not clear the list of "Available Instruments" before a scan. Any instruments found during a scan are added to the list.

6. If you can find your R&S SMW200A in the list, close the dialog.

If the instrument you want to work with is not listed, check if the following prerequisites are fulfilled, and scan for the instrument again:

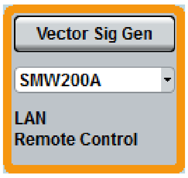
- Is the instrument switched on?
- Is the instrument accessible via LAN, GPIB, or USB?
- Is the firewall of the instruments' operating system deactivated?

3.3.2 Selecting the Instrument

The waveform calculation is based on instruments settings. Thus, before you can create a waveform, you must select and configure any instrument in advance (see also Chapter 3.3.1, "Configuring the Instrument", on page 24).

For the calculation, it is not imperative that a real instrument is connected. You can also create a waveform on the basis of a manually configured instrument (offline).

► In the instrument selection list of the "Vector SigGen" block, select "SMW200A".



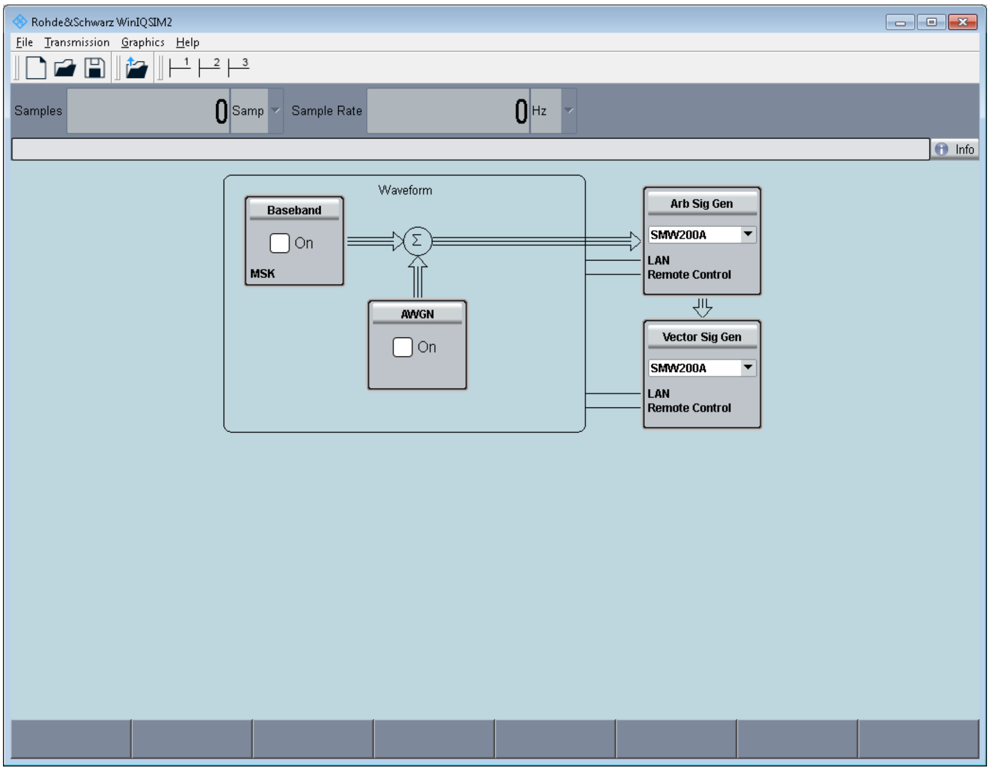
The R&S SMW200A is selected. The prerequisites for waveform calculation are fulfilled.

3.3.3 Generating the Waveform Signal

The example task is to configure a digital signal in accordance with the EUTRA/LTE standard. It introduces the way to access the settings and the configuration principle common for digital standards, and the R&S WinIQSIM2 software.

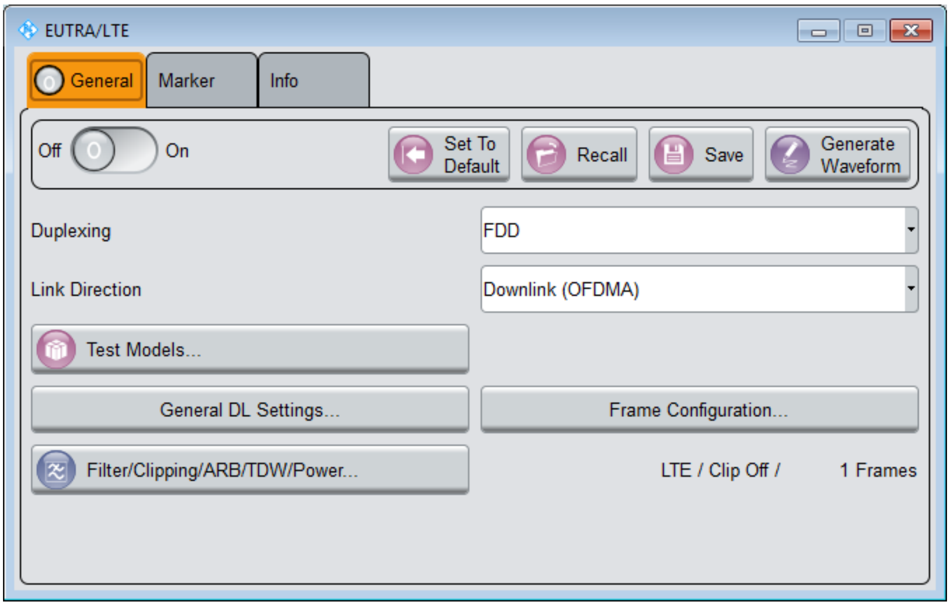
We use one of the provided EUTRA test models (E-TM), to show how to configure and generate a test signal.

1. In the menu bar, select "File > new" to start from an initial state.



R&S WinIQSIM2 sets all settings to default, except of the selected instruments in the "ARB Sig Gen" and "Vector Sig Gen" blocks.

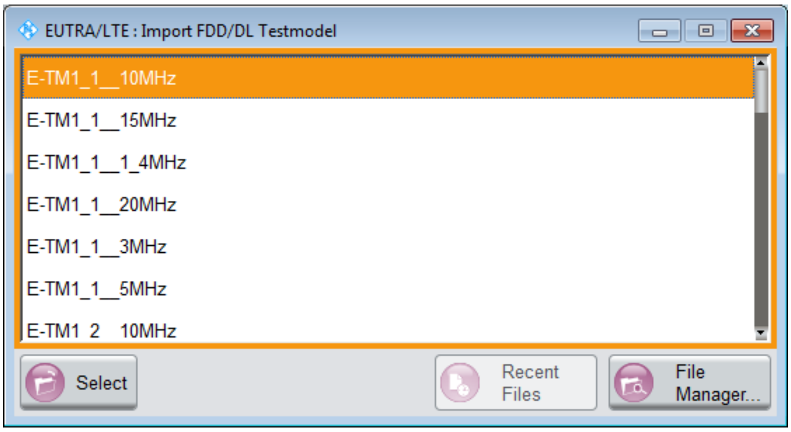
2. In the block diagram, select "Baseband > EUTRA/LTE".



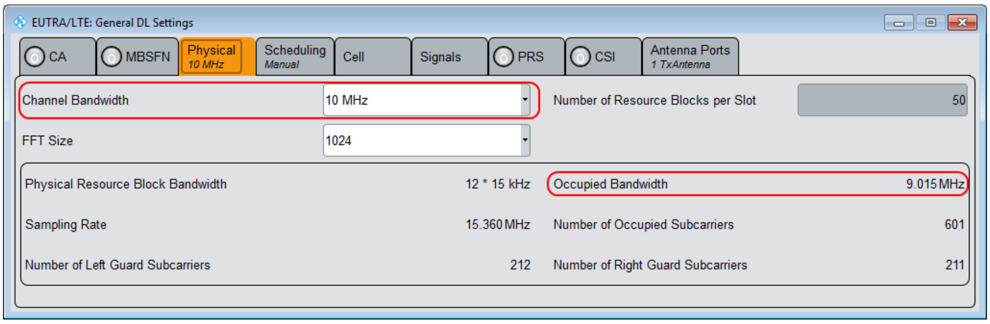
The "EUTRA/LTE" settings dialog contains the parameters to configure the wave-form signal.

3. In the "General" tab, select "Link Direction > Downlink (OFDMA)".

4. Select "Test Models > E-TM1_1__10MHz".
5. Confirm with "Select".
   The "Test Models" is a function for quick selection and settings adjustment according to one of the various EUTRA test models (E-TM). A standard "File Select" function enables you to select form files with predefined settings.

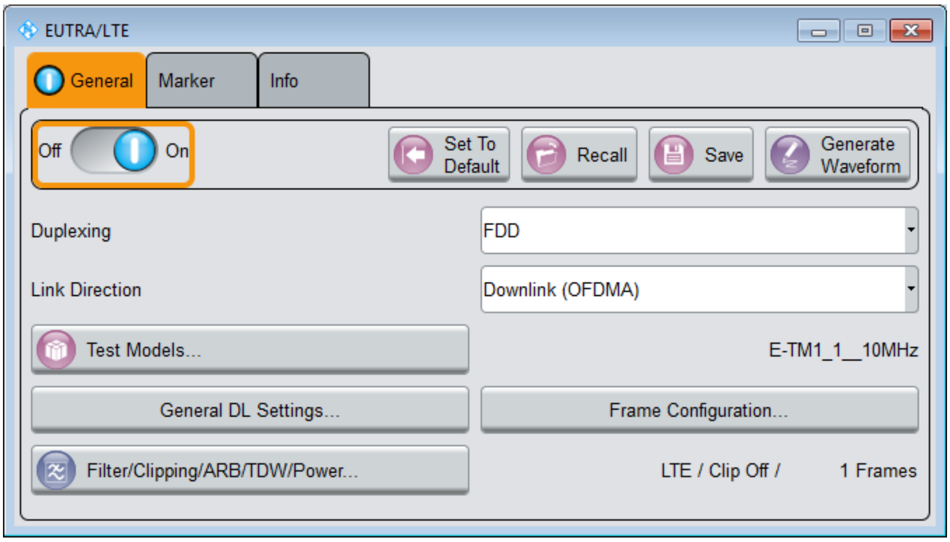


   The dialog closes automatically and the user interface confirms the name of the selected file.
6. Select "General DL Settings" to have a look at the channel bandwidth and the occupied bandwidth of the selected signal.



   The selected test model operates with a channel bandwidth of 10 MHz. The signal occupies 9.015 MHz.
7. Close the "General DL Settings" dialog.

8. In the "General" tab, select state "On".



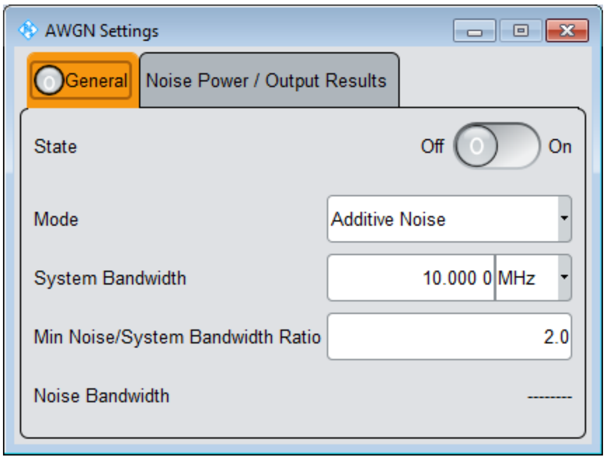
R&S WinIQSIM2 calculates the EUTRA/LTE test signal with the selected channel bandwidth. It displays the characteristic signal parameters, i.e. the number of "Samples" and the used "Sample Rate" in the status bar of the main window.

3.3.4 Adding Noise (AWGN) to the Signal

In R&S WinIQSIM2, you can also superimpose the waveform with noise.

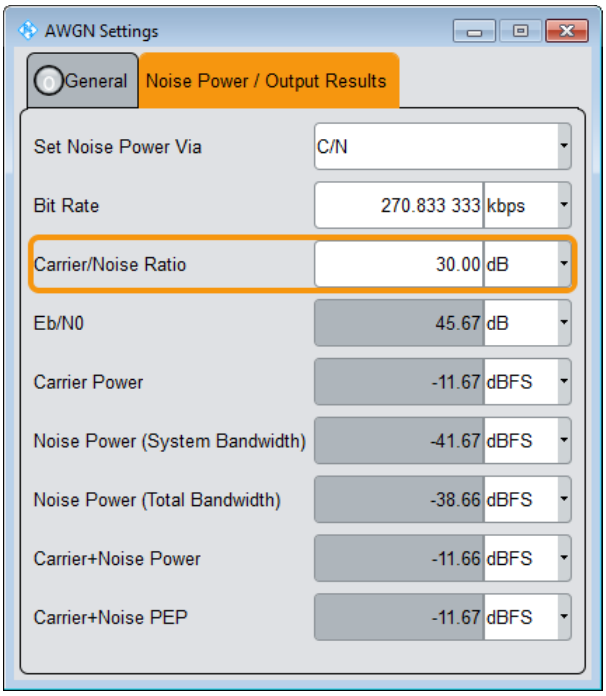
In the example, we generate a "White Gaussian Noise" signal that is to be superimposed with the interference-free LTE signal. The minimum noise to system bandwidth ratio is 2.

1. In the block diagram, select "AWGN".



The "AWGN Settings" dialog contains the parameters for configuring additive white gaussian noise, noise level or CW interfering.

2. In the "General" tab, select "Mode > Additive Noise".
3. Set "System Bandwidth > 10 MHz".
4. Enter "Min. Noise/System Bandwidth Ratio > 2".
5. In the "Noise Power / Output Results" tab, enter "Bit Rate > 270.833333 kbps".



6. Enter "Carrier/Noise Ratio > 30 dB".
7. In the "General" tab, select state "On".

   R&S WinIQSIM2 generates the noise signal.

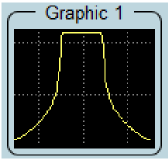
3.3.5 Displaying and Configuring the Graphical Representation of the Waveform Signal

It is often useful to verify a generated waveform before storing or transmission. This example shows you how to configure the display function of the R&S WinIQSIM2, and how to view the signals in particular, both the baseband and the AWGN, and the superimposed signal. The example uses the waveform generated as described in Chapter 3.3.3, "Generating the Waveform Signal", on page 26, and the AWGN signal of Chapter 3.3.4, "Adding Noise (AWGN) to the Signal", on page 29.

The graphics panel is an important tool for checking the signal configuration. In this panel, you can display the generated I/Q signal as I/Q diagram, as well as derived representations thereof like the vector diagram or the signal spectrum. For more information, see Chapter 6, "Displaying Simulated Waveforms Graphically", on page 162.

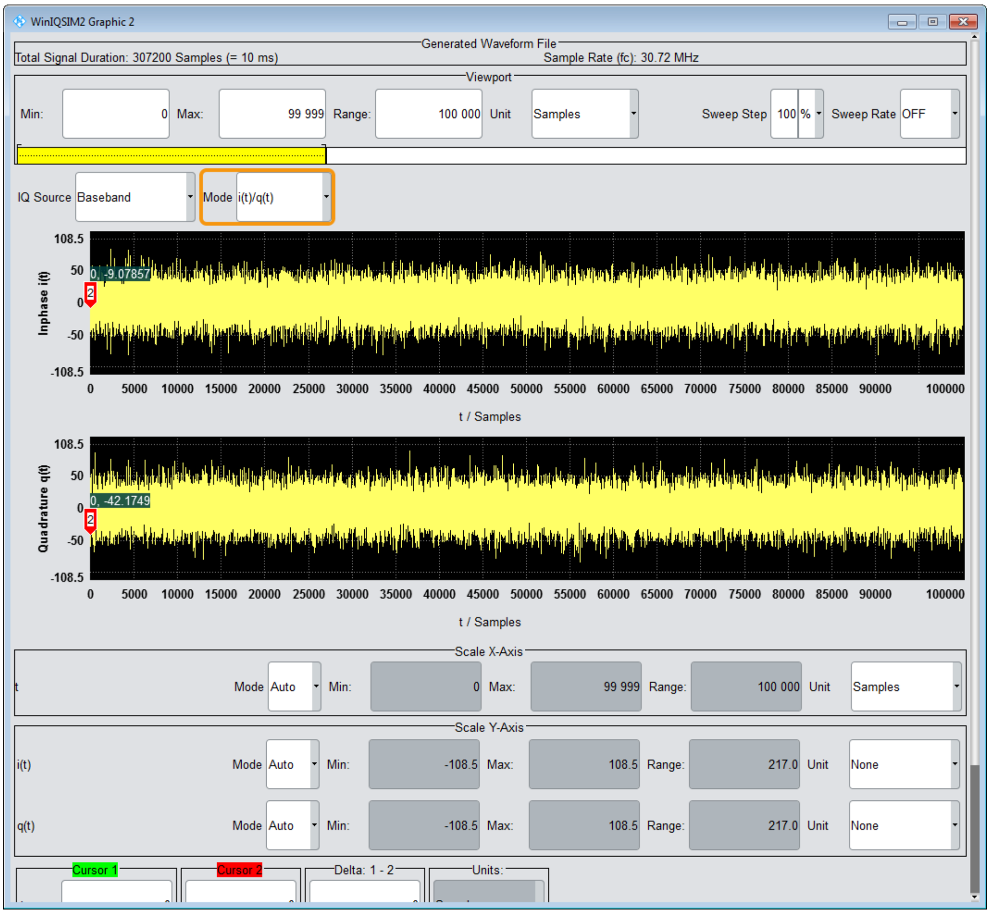
To display the baseband waveform signal

1. In the menu bar of the main window, select "Graphics > Graphic 1 > Graphic 1 (Preview Only)".
   R&S WinIQSIM2 indicates the graphics preview in the block diagram.

   A small icon assigned directly to the signal line indicates the shown waveform.
   To access the graphic settings, perform one of the following:

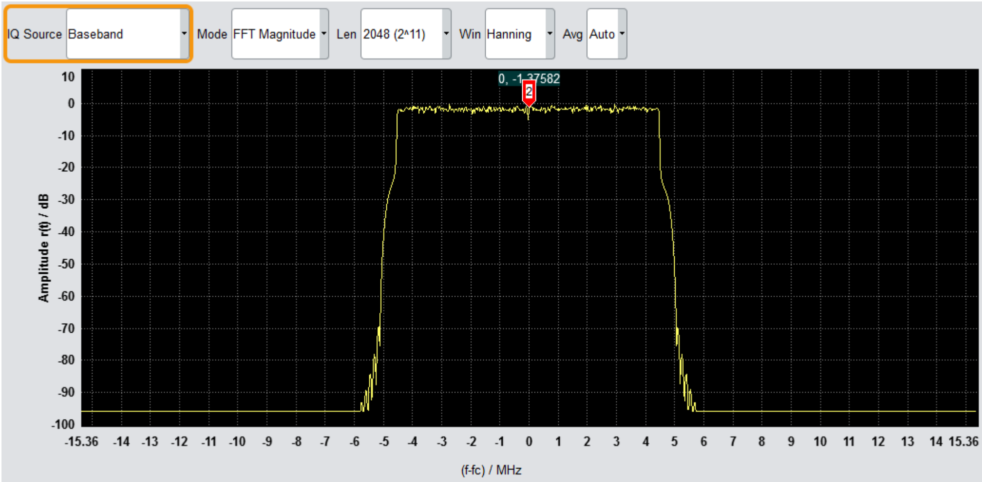
   a) Double-click the small graphics display.
   b) In the menu bar, select "Graphics > Graphic 1 > Graphic 1 (Complete)".

   The "Graphic 1" settings dialog opens.

By default, R&S WinIQSIM2 displays an "I(t)/q(t)" diagram. You can select the graphical representation in the "Mode" combo box.

2. Select "IQ Source > Baseband".
3. Select "Mode > FFT Magnitude".
4. Select "Len > 2048 (2^11)".

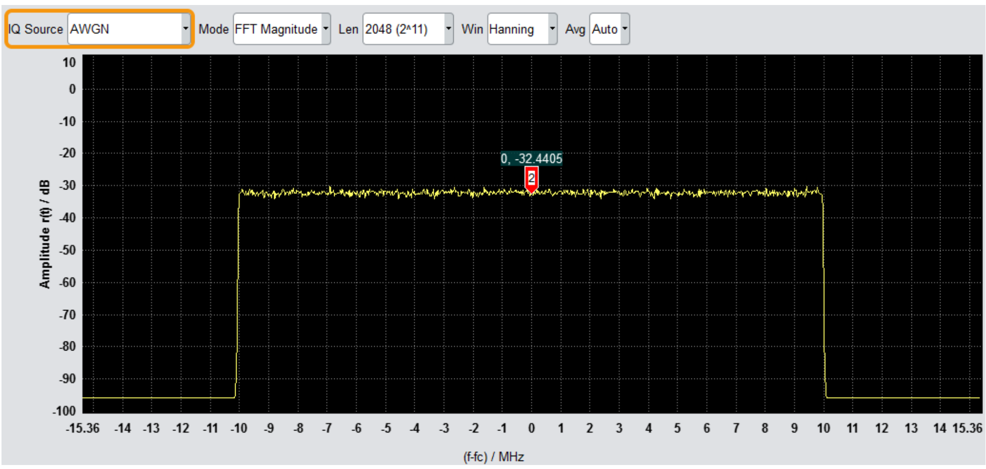


The power spectrum displays the baseband waveform signal.

To display the AWGN signal

In this example, we use the second graphic function to display the AWGN signal. Perform the following steps:

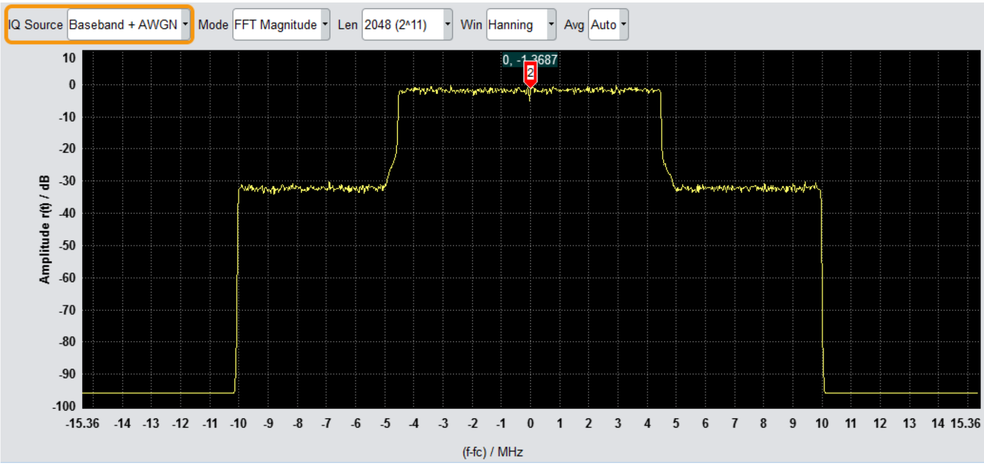
1. In the menu bar of the main window, select "Graphics > Graphic 2 > Graphic 2 (Complete)".
2. In the "Graphic" dialog, select "IQ Source > AWGN".
3. Select "Mode > FFT Magnitude".
4. Select "Len > 2048 (2^11)".



The power spectrum displays the generated additive white gaussian noise signal.

To display the baseband signal superimposed with AWGN

1. In the menu bar of the main window, select "Graphics > Graphic 3 > Graphic 3 (Complete)".
2. Select "IQ Source> Baseband + AWGN".
3. Select "Mode > FFT Magnitude".
4. Select "Len > 2048 (2^11)".



The power spectrum displays the waveform derived from the baseband signal superimposed with the AWGN signal.

To retrieve more information, zoom in the spectrum and place the markers to the positions to measure the corresponding values (see To zoom a particular area of the waveform).

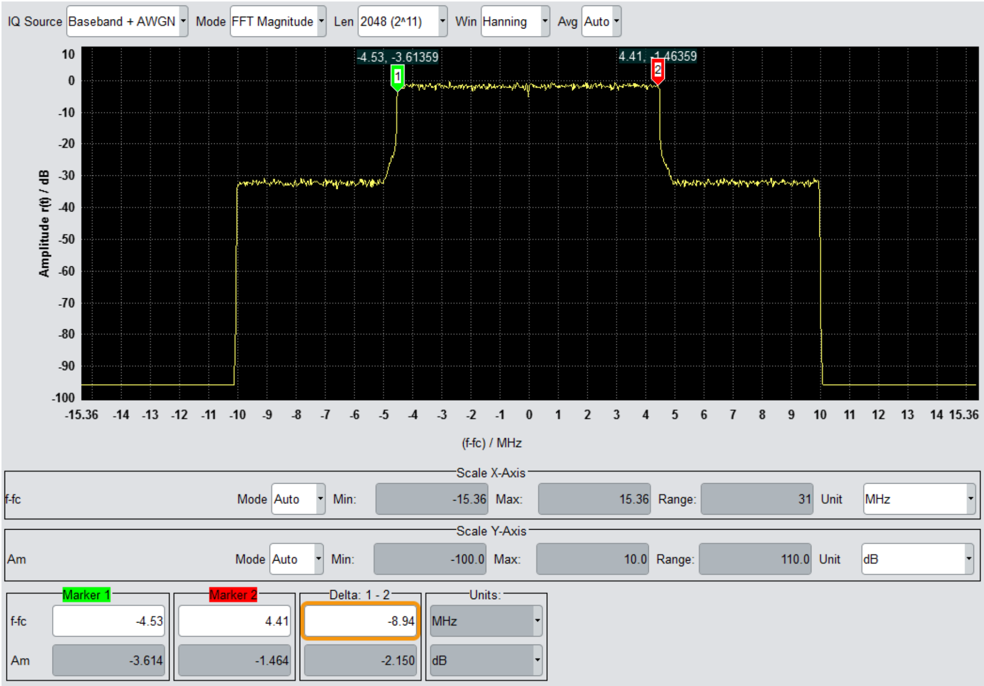
Using markers to analyze the signal

Using the markers you can determine particular readings of the signals, e.g. to verify the settings directly in the graph. You can position the markers in the diagram to find out the appropriate values.

In the following examples, we set the markers to determine the occupied bandwidth of the baseband waveform, and the carrier/noise ratio.

To determine the occupied bandwidth of the baseband signal

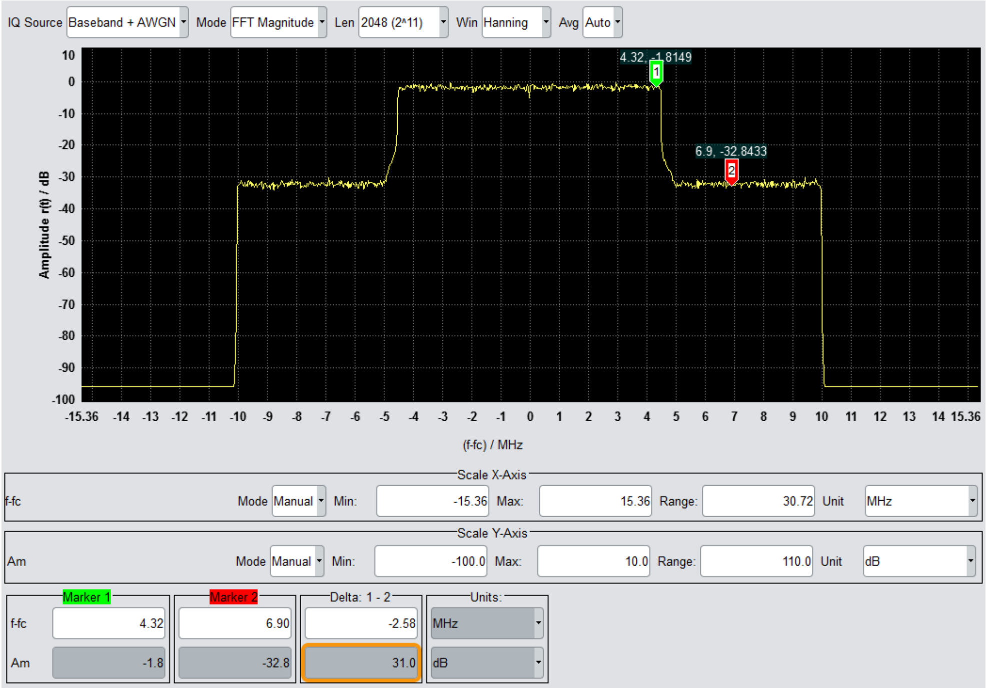
1. Open the "Graphics 3" dialog.
2. In the diagram, left-click the first marker and keep the mouse key pressed.
   R&S WinIQSIM2 denotes the current X-axis position by a colored line.
3. Drag and drop the marker to the start position of baseband signal (On-time).
4. Accordingly, move the second marker to the end position.



You can see the current X and Y-axis positions of the markers in the diagram, or in the "Marker" fields in the lower section of the dialog. The calculated "Delta 1-2" frequency value reflects the occupied bandwidth of the EUTRA/LTE signal approximately, see Generating the Waveform Signal, step 6.

To determine the carrier to noise ratio

1. Open the "Graphics 3" dialog.
2. In the diagram, left-click and hold the first marker.
3. Drag and drop "Marker 1" to the top level of the noise signal.
4. Accordingly, move the second marker to the top level of the baseband waveform signal.

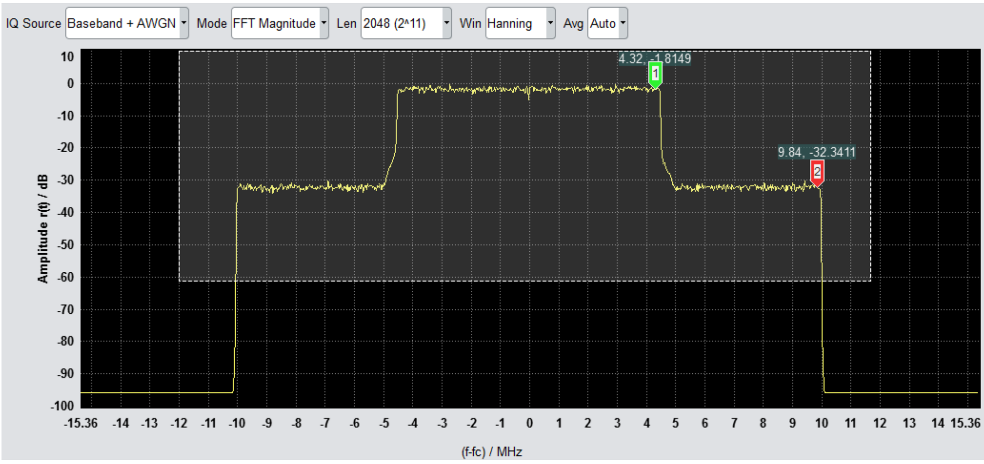


R&S WinIQSIM2 determines the carrier/noise ratio of approximately 30 dB calculated on the basis of the markers amplitude values.

To zoom a particular area of the waveform

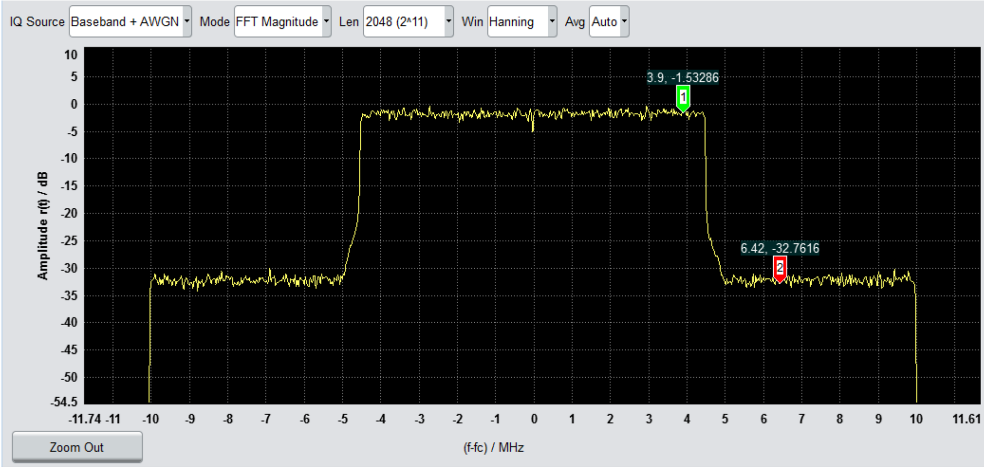
To retrieve more information, you can enlarge any section of the graph and thus visualize particular values of the signal in detail.

1. Left-click and hold the upper left corner of the section you want to zoom.
2. Draw the zoom rectangle.



A dotted rectangular frame denotes the marked area.

3. Release the mouse button.

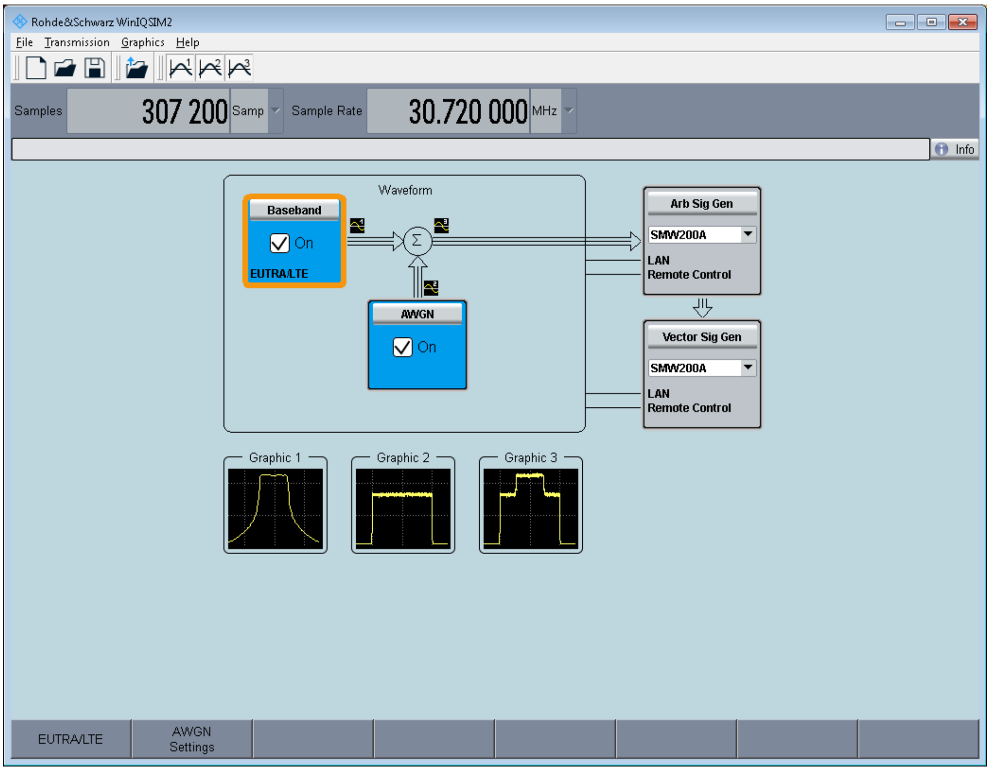


R&S WinIQSIM2 enlarges the selected section. Now you can analyze the signal trace in more detail visually.

4. To return to the initial size, select "Zoom Out".

To display the graphics in the main window

► Close all graphics dialogs.



R&S WinIQSIM2 shows the small graphics preview of all active diagrams. "Graphic 1" displays the baseband signal, in "Graphic 2" you can see the AWGN signal, and in "Graphic 3" the baseband signal interfered with noise.

3.3.6 Transmitting the Generated Waveform to the R&S SMW200A

Transferring the waveform file to the signal generator, requires that the connection between R&S WinIQSIM2 and the R&S SMW200A is established.

You can see the connection at a glance in the block diagram denoted by the two thin control lines, see "Block diagram" on page 45.

In case of crossed lines, establish the connection as described in Chapter 3.3.1, "Configuring the Instrument", on page 24 and Chapter 3.3.2, "Selecting the Instrument", on page 26.

In addition the following requirements must be met:

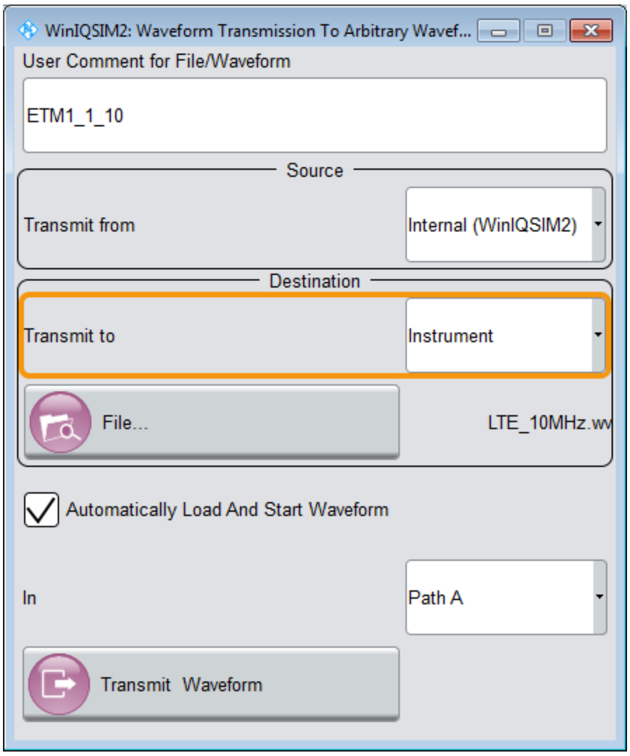
- Enabled file transfer via LAN and the used interface protocol
- Enabled write permission on the instrument's file system
- Disabled firewall in the operating system (applies to particular instruments as e.g. the R&S AFQ100A)

See the user manual of the instrument for more information to its safety systems.

In this example, we assume that the connection over LAN is established and enabled.

To transmit the generated waveform to the R&S SMW200A and activate the ARB signal generation

1. To configure the transmission, perform one of the following:
   a) In the menu bar, select "Transmission > Transmit".
   b) In the tool bar, select .



The "Waveform Transmission To Arbitrary Signal Generator" dialog opens.

2. Select "Source > Internal (WinIQSIM2)"

   R&S WinIQSIM2 selects the last generated waveform data automatically.

3. Select "Destination > Transmit To > Instrument".

4. Select "File" to determine the file name and directory for storing on the instrument.

   a) Select, or create the destination directory on the instrument.
   b) Enter the "File Name".
   c) Confirm with "Ok".

5. Enable "Automatically Load and Start Waveform".

6. Select the required baseband path, for example "Path A".

7. Add a comment to the waveform.

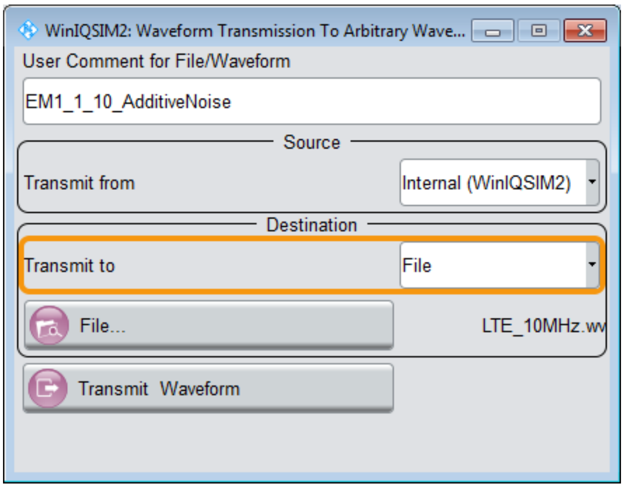
8. Select "Transmit" to start the transmission.

   R&S WinIQSIM2 transmits the waveform file to selected folder of the R&S SMW200A. After successful transmission, the R&S SMW200A loads the file, enables the ARB and plays the transmitted waveform, and automatically provides the waveform signal at the I/Q Modulator outputs.

3.3.7 Transmitting the Generated Waveform to a File

To store the generated waveform file

1. In the tool bar, select .



   The "Waveform Transmission To Arbitrary Signal Generator" dialog opens.

2. Add a comment to the waveform.
3. Confirm with ENTER.
4. Select "Source > Internal (WinIQSIM2)"

   R&S WinIQSIM2 selects the last generated waveform data automatically.

5. Select "Destination > File".
6. Select "File" to determine the file name and directory for storing the waveform.
   a) Select, or create the destination directory.
   b) Enter the "File Name".
   c) Confirm with "Save"
7. Select "Transmit Waveform".

   R&S WinIQSIM2 stores the waveform file in the specified directory on your computer.

3.4 Overview of R&S WinIQSIM2

This section helps you to get familiar with R&S WinIQSIM2. It provides an introduction to the general concept of the software, including the description of the main blocks in the signal generation flow.

3.4.1 Brief Introduction to the Concept of R&S WinIQSIM2

The signal generation software comprises all the functionalities required for creating waveform files of digitally modulated baseband signals. It provides almost all standard-compliant digital signals, user-definable signals with selectable modulation parameters as well as multi carrier signals.

The graphical user interface provides intuitive operation via a block diagram, representing the core functionalities in blocks. You can control the entire process of the signal configuration via the block diagram. At a glance, you can see the status of signal configuration, active interfering signals, the signal flow and connected instruments. In addition, R&S WinIQSIM2 shows the signals graphically.

3.4.2 Possible Ways to Operate R&S WinIQSIM2

This chapter provides an overview on how to work with R&S WinIQSIM2 and describes the manual operation of the software and also the alternative ways of operation.

There are two ways to operate R&S WinIQSIM2:

- Manual operation:
  Run the software on your PC and use the mouse and/or keyboard.
- Remote control:
  A remote control program either installed on the same or another computer, controls R&S WinIQSIM2, see Chapter 3.5.3, "Remote Control", on page 49.
  With remote control, you can create programs to automate repeating settings, tests and measurements.
  This way of operation is described Chapter 11, "Automation of R&S WinIQSIM2", on page 228.

3.4.3 The Signal Flow at a Glance

The framed area symbolizes R&S WinIQSIM2 as one unit. It shows the blocks for generation of the baseband and interfering signals, and the signal flow to the peripheral signal generators.

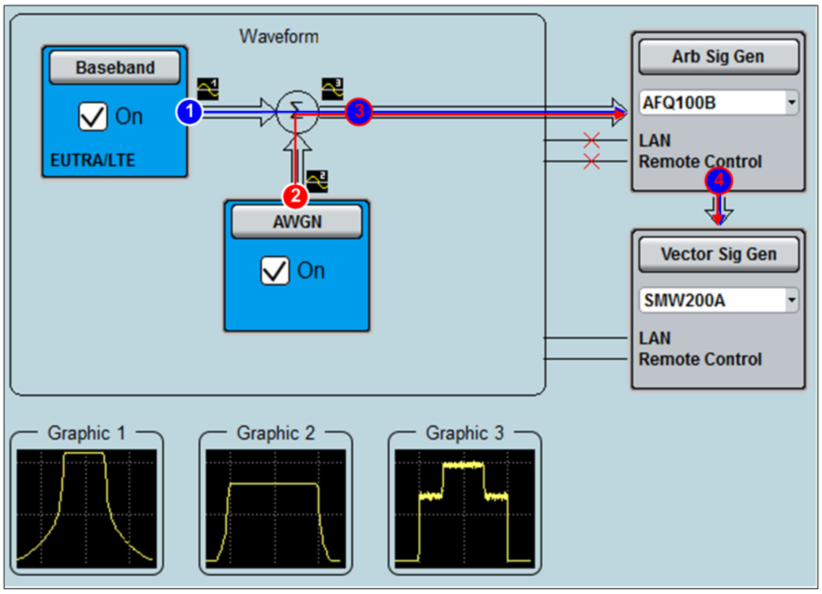


Figure 3-2: R&S WinIQSIM2 signal flow

1 = digital waveform data
2 = noise data
3 = digital waveform data superimposed with noise
4 = electrical signal generated from the waveform data

3.4.4 Baseband Block

The "Baseband" block represents the source of the baseband signals.

This functional block is the access point to:

- *The internal baseband generator*
  With the baseband generator, you can create a user-defined signal ("Custom Digital Modulation"), including MCCW signal generation and "Import IQ Data".
- *The available digital standards*
  Generation of digital signals in accordance with the supported standards require the corresponding R&S WinIQSIM2 digital standard options installed on the instrument.

3.4.5 AWGN Block (Additional White Gaussian Noise)

The "AWGN" block controls the noise generator (AWGN).

You can create a white noise signal ("Additive White Gaussian Noise"), or a sinusoidal signal ("CW Interferer") with adjustable frequency offset, and superimpose this noise signal with the baseband signal. Alternatively, you can generate a pure noise signal ("Noise only").

3.4.6 Arb Sig Gen and Vector Sig Gen Blocks (Connected Instruments)

These blocks represent instruments that can process the waveforms created by R&S WinIQSIM2. You can configure a remote connection to an instrument in the network, transfer the created waveform file directly and even activate the signal generation on the instrument. In addition, you can scan for connected instruments in the network.

R&S WinIQSIM2 distinguishes two types of generators:

- "Arb Sig Gen", the arbitrary waveform generators for generating the I/Q baseband signal.
- "Vector Sig Gen", the vector signal generators for modulating the I/Q signal to RF.

Both generator types can be integrated in a single instrument, as e.g. in the R&S SMW.

But there are also instruments that provide only one component, such as the I/Q modulation generator R&S AFQ or the R&S SGT vector RF source.

R&S WinIQSIM2 assigns detected or manually configured instruments according to their functionality automatically to respective signal generator block. The instrument selection fields of the blocks provide all available instruments listed under their symbolic name.

3.5 Controlling R&S WinIQSIM2

This section provides an overview on how to work with R&S WinIQSIM2. It describes what kind of information is displayed in the diagram area, how to operate R&S WinIQSIM2 manually or in remote control mode, and how to use the online help.

3.5.1 Understanding the Display Information

The application window shows all main settings and control elements of R&S WinIQSIM2. All menus and dialogs use known elements, like for example selection lists, icons, check boxes, and entry fields.

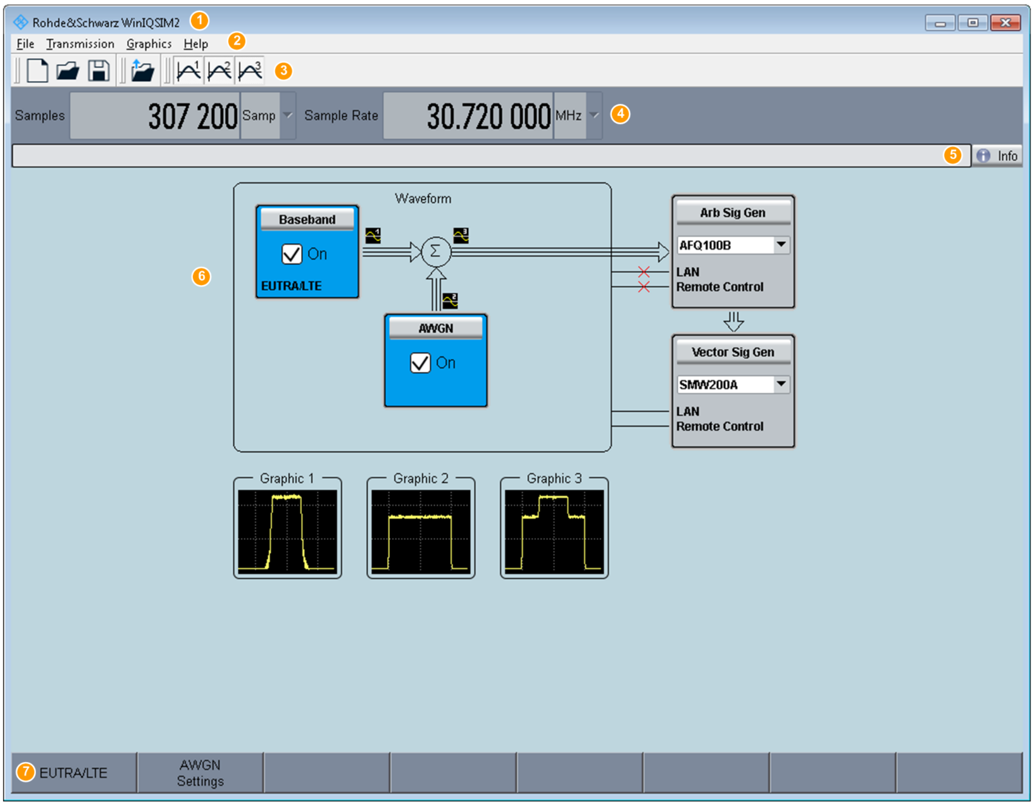


Figure 3-3: R&S WinIQSIM2 main application window

1 = Title bar
2 = Menu bar
3 = Toolbar
4 = Status bar
5 = Info line
6 = Block diagram
7 = Taskbar/Softkey bar

The following sections explain the labeled operation areas in detail.

Menu Bar / Toolbar

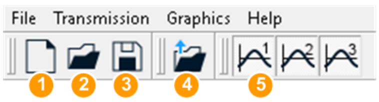


Figure 3-4: R&S WinIQSIM2 menu bar & toolbar

1 = New
2 = Open
3 = Save
4 = Transmit
5 = Graphics

The toolbar contains the most commonly used general functions of the application. The toolbar icons provide quick and easy access with the mouse pointer. Alternatively, you can also find these functions in the menus or you can use keyboard shortcuts to execute a function. For an overview, see Menu bar entries / toolbar icons / keyboard shortcuts.

Table 3-4: Menu bar entries / toolbar icons / keyboard shortcuts

Legend	Toolbar	Menu bar	Shortcut	Description
1		"File > New"	ALT+F > N	Resets R&S WinIQSIM2 to default.
2		"File > Open"	ALT+F > O	Loads an existing configuration file, containing specific settings of a configured application. The file extension is predefined (`*.savrcl`).
3		"File > Save"	ALT+F > S	Saves all settings to the current configuration to an existing file.
-	-	"File > Save as"	ALT+F > N	Saves the complete settings to the current configuration to a new file (`*.savrcl`).
-	-	"File > Setup > Software Options"	ALT+F > P	Displays the current software version and options.
-	-	"File > Setup > Temporary Files"	ALT+F > P	Opens a dialog where you can define the location for temporary files.
-	-	"File > Setup > Undo/Redo"	-	Erases the last change done (undo)., or reverses the undo (redo).
-	-	"File > Exit"	ALT+F > X	Terminates R&S WinIQSIM2. The current configuration settings are saved and restored when you restart the program.
4		"Transmission > Instruments"	ALT+T > I	Opens a list of available instruments.
-	-	"Transmission > Transmit"	ALT+T > T	Opens a dialog for configuring file transmission to an ARB instrument.
5	 	"Graphics 1(2,3) > Graphics 1/2/3 (Preview only)"	ALT+G > 1... > 1... (toggle)	Displays a small graphics preview in the block diagram.
-	-	"Graphics 1/2/3 > Graphics 1/2/3 (Complete)"	ALT+G > 1 > 1 (toggle)	Opens the dialog for graphical display.
-	-	"Help > Contents"	ALT+H > C	Opens the R&S WinIQSIM2 online help.
-	-	"Help > Index"	ALT+H > C	Shows the index of the online help.
-	-	"Help > About"	ALT+H > A	Displays information on the software version.

Status bar

The status bar displays the main characteristics of the active signal, like samples or symbols and data rates.

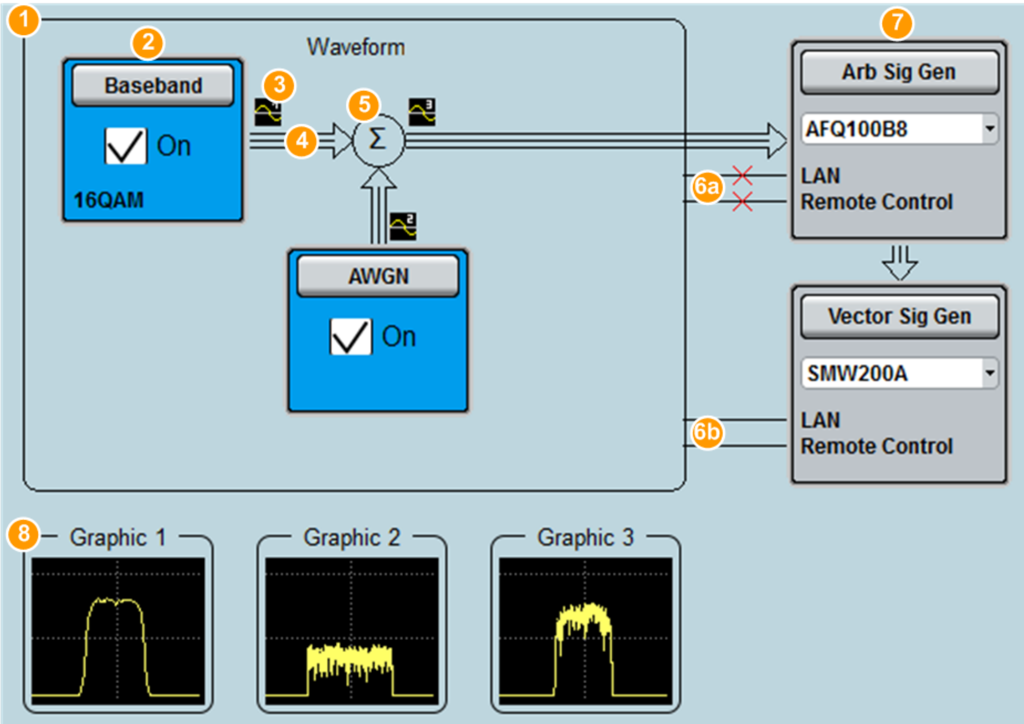
Info Line

The info line shows brief status information and error messages. It appears above the block diagram, when an event generates a message. For information on the error messages and warnings, refer to Chapter 13, "Troubleshooting and Error Messages", on page 392.

You can also access an info window with detailed information on all messages in a history list. For details, see Chapter 10.2, "Querying Error Messages & Info Key", on page 224.

Block diagram

The block diagram shows the current configuration and the signal flow in R&S WinIQSIM2 and to the external instruments with the aid of function blocks, connected by signal lines. The following figure displays elements that can appear in the block diagram.



1 = Waveform generation section
2 = Functional blocks
3 = Graphics indicator
4 = Signal lines
5 = Summation sign
6a / 6b = Network and remote control lines (not connected / connected)
7 = Signal generator blocks
8 = Small graphics preview

Legend	Item	Description
1	Waveform generation section	Covers the functional blocks for generating the waveform and additive white gaussian noise.
2	Functional block	Represents a basic task in signal generation. The push button provides access to any number of associated actions to accomplish the task. The On/Off (checkbox) and the block label quickly activates the basic task.
3	Graphics indicator	Denotes that the signal is displayed graphically (8).
4	Signal lines	Show the signal flow.
5	Summation sign	Denotes merged signals (baseband and noise)
6a / 6b	Control lines	Indicate the connection to external instruments. ● solid line the connection to the instrument exists ● crossed line the connection is interrupted (6a) **Note:** Two parallel solid lines indicate that the interface is set up and ready for remote control. If one of two parallel lines is crossed, the interface configuration and the selected connection do not match. Ee.g. the instrument is visible via LAN, but the GPIB interface is configured for remote control.
8	Small graphics preview	Shows small graphics in the block diagram.

Taskbar

The "Taskbar" contains labeled softkeys of active dialogs.

Whenever you open a settings or a graphics dialog, it is automatically assigned to a softkey in the "Taskbar".

If you minimize a dialog, R&S WinIQSIM2 keeps it active in the background and in the taskbar. Click the corresponding softkey, to maximize it again.

Figure 3-5: Softkeys representing active settings and graphics dialogs

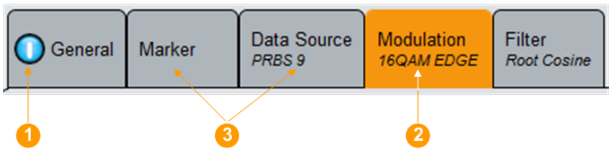
1 = Waveform settings dialogs
2 = Graphics dialog
3 = Remote control settings dialog

R&S WinIQSIM2 maintains up to eight active dialogs in the background. Each additionally opened dialog turns off the function that has been opened first, according to FIFO (first in first out).

Additional display characteristics

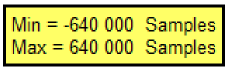
The following section provides a short insight on the indication of the screen in general, and significant elements that you see under specific operating modes, in dialogs or settings.

- Appearance of active elements
  – *Active* elements like On/Off switches, state buttons, blocks and symbols have a blue background.
  – *Selected* elements such as blocks, tab labels, entry fields or check boxes are highlighted orange.
  – *Inactive* elements are gray.
- Dialogs
  The dialogs of R&S WinIQSIM2 contain the setting parameters of the functions.
  – The header of a dialog follows the general naming convention *<DialogName> <SourceDialog>*.
  – Key parameters indicated in tab labels.
  A dialog is divided into tabs with logically grouped parameters. The tab label expresses the content and can additionally contain status indicators or the set value of a key parameter.



1 = status elements
2 = active, selected elements
3 = inactive elements

- Tooltips
  In edit mode, a tooltip indicates the value range of a parameter or shows information on current settings.



- Context-sensitive menus
  Within the entire screen display, including single parameters, you can access context-sensitive menus providing additional functions. The list varies, depending on where you access the menu.

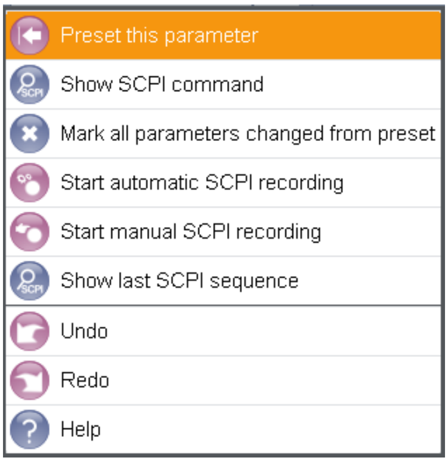
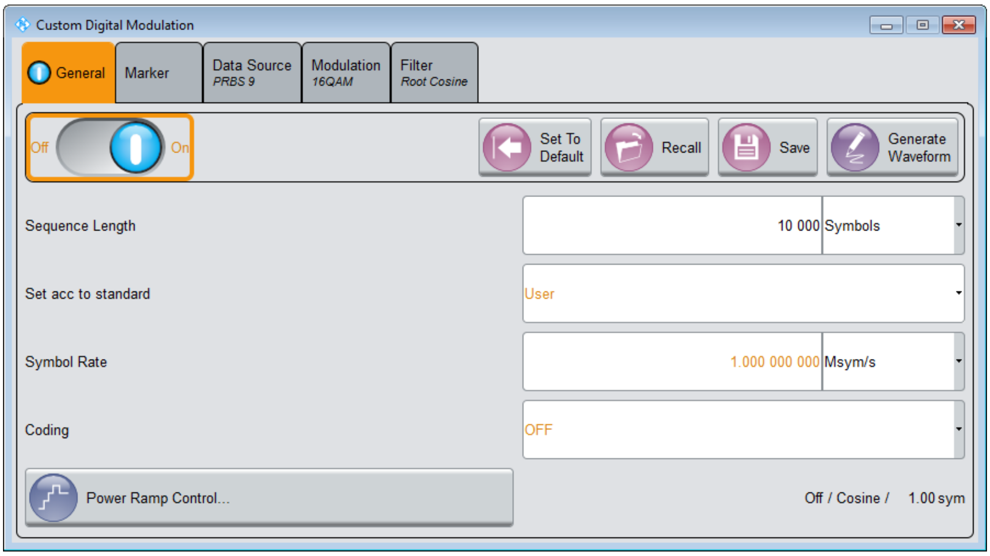


Figure 3-6: Context-sensitive menu

- Parameters changed from preset

Orange displayed parameters indicate that the setting is different from the default value.



3.5.2 Means of Manual Operation

Like any software, you can control R&S WinIQSIM2 directly with the keyboard and mouse. At first, you can operate the application intuitively via the block diagram. Further functions are built in menus and dialogs using familiar elements such as selection lists, check boxes and input fields.

The following overview provides a brief insight on the main operating elements:

- To open a dialog:
  - Select the required block and then the menu entry.
  - Select the minimized view (thumbnail) in the taskbar.
- To minimize a dialog, select the "Minimize" icon in the upper right corner.
- To close a dialog:
  - Select the "Close" icon in the upper right corner.
  - Press ESC on the keyboard.
- To select an item in a list, select the list, scroll in the list and select the required item.
- To enter data, select the entry field and insert the data and confirm with the ENTER key.
- To abort an entry, press the ESC key. R&S WinIQSIM2 cancels the entry without changing the settings.

Undo and redo actions

Accessed via the context-sensitive menu:

- "Undo" allows you to restore one or more actions on the instrument. Depending on the available memory, the "Undo" steps can restore all actions.
- "Redo" restores a previously undone action.

3.5.3 Remote Control

In addition to working with R&S WinIQSIM2 directly via the user interface, you can operate and control it from a remote computer. Remote control operation allows automation of the configuration process and is especially useful when a higher configuration speed is required.

In remote control mode, you can configure the settings of R&S WinIQSIM2 via a controller software using remote control commands (SCPI).

The controller software can run on the same computer as R&S WinIQSIM2 or another PC. For details on this topic, see also Chapter 11, "Automation of R&S WinIQSIM2", on page 228.

- R&S WinIQSIM2 and the controller software on the same PC
  The two programs communicate via the localhost link (IP address = 127.0.0.1).
- R&S WinIQSIM2 and the controller software on different PCs
  Communication requires a LAN connection.

For remote control over LAN or USB, you can use the R&S VISA ("Virtual Instrument Software Architecture") library provided for download at the Rohde & Schwarz website http://www.rohde-schwarz.com/rsvisa.

See also:

- How to operate R&S WinIQSIM2 via remote control is described in Chapter 11.2, "How to Set up a Remote Control Connection", on page 233.
- For basic information on remote control, as interface messages, the SCPI command structure, status reporting system etc. see Chapter A.1, "Reference Information on Remote Control", on page 397.
- SCPI commands are listed in the respective functions description, with a link to the actual description of the command.
  All available SCPI commands of R&S WinIQSIM2 are described in Chapter 12, "Remote Control Commands", on page 251, and alphabetically listed at the end of the user manual.
  Note: In the individual manuals of the digital standards, the specific functions of the standards are described in detail.

The R&S WinIQSIM2 online help, however, includes also the descriptions of digital standards.

3.6 Getting Information and Help

In some dialogs, graphics are included to explain the way a setting works. For further information, you can use the following sources:

- The general help explains a dialog, provides instructions, and general information.
- The context help provides functional description on a setting parameter.
- Tooltips give the value range of the parameter.

To open the general help

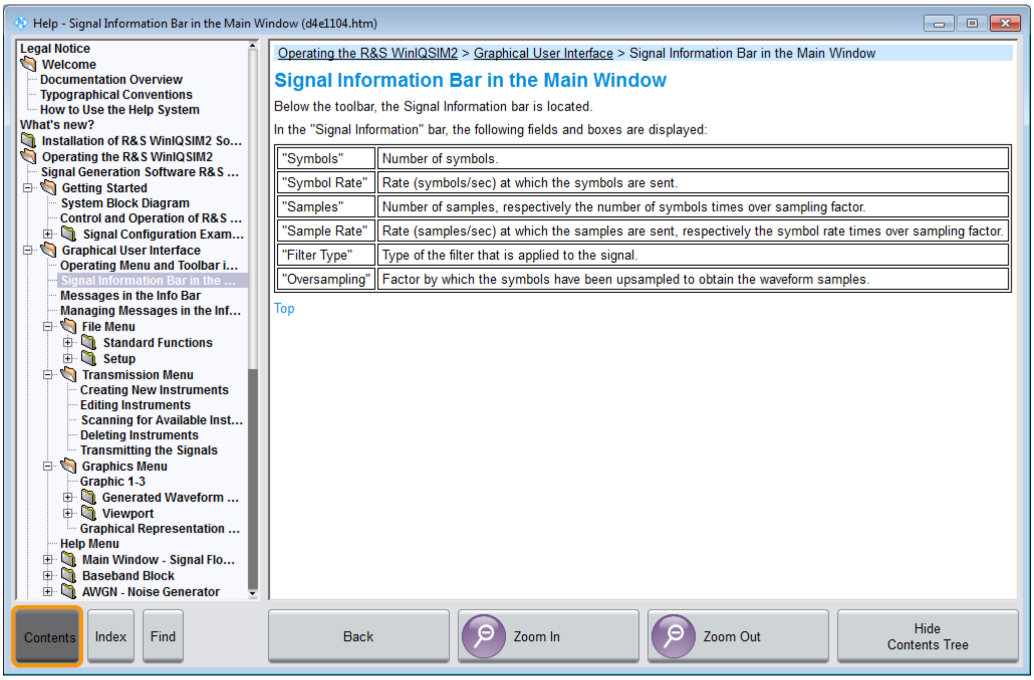
► In the menu bar, select "Help > Contents".

The start page of the online help appears.

To display context help

► For information on a specific parameter, press the F1 key:

The "Help" dialog opens. You can browse the help for further information.



Contents of the help dialog

The help dialog contains two main panels:

- "Contents" - covering a table of help contents
- "Topic" - contains a specific help topic

The help system additionally provides an "Index" and a "Find" area, as well as "Zoom" functions that are accessed by means of the corresponding buttons.

Navigating in the table of contents and in the help topics

1. To navigate within the table of contents entries, select an entry and scroll with the mouse or the UP/DOWN keys.
   Entries that contain further entries show a plus sign for folding out.
   When selected, you can immediately see the description in the "Topic" panel.
2. To scroll up or down in the directory tree or the help text, use the scroll bar on the right side of the panels. Alternatively, you can use the up/down cursor keys.
3. To follow a cross reference, select the link text (marked in blue font).
4. To return to the previous page, select "Back".
   This function scrolls back all steps that you have performed before.
5. To maximize the "Topics" window, you can hide the contents tree with the "Hide Contents" button, and vice versa.
6. To toggle between the "Contents" and "Topic" panels using the keyboard, use the right/left cursor keys.
7. To get from the "Contents" or "Topics" panel with the keyboard to the softkeys, press ESC, and then use the cursor keys.

Using the index

1. Select the "Index" button in the "Help" display.
2. Enter the first characters of a topic you are interested in. R&S WinIQSIM2 displays all entries that start with these characters .
3. Select the index entry.

   When selected, you can immediately see the description in the "Topic" panel.

To display tooltips

For information on the range of a specific parameter:

► Select the entry field.

  In edit mode, the tooltip indicates the possible value range of a parameter.

4 Configuring the Baseband Source

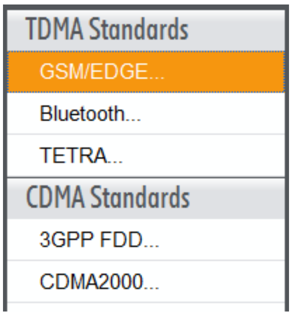
The R&S WinIQSIM2 software enables you to simulate various digitally modulated signals in accordance with the definitions in the digital standards or with user-definable characteristics. In addition, you can configure multi carrier signals or multi segment waveforms.

In addition, you can import unprocessed custom I/Q data via the TCP/IP interface. R&S WinIQSIM2 processes this data as well, i.e. you can add an interference signal, configure a filter, use the graphical display, or save and transmit the waveform.

4.1 How to Access the Functions in the Baseband Block

To access the functions in the baseband block

1. In the block diagram, select "Baseband".
2. Select the corresponding entry of the context menu.



The "Baseband" block provides access to the configuration of the internal baseband source. It offers a selection list with all supported standard compliant digital standards, customer digital modulation as well as multi carrier and multi segment waveforms.
A short designation in the block indicates the currently selected digital standard or modulation.

4.2 Generating Signals According to Digital Standards

R&S WinIQSIM2 generates digital signals in accordance with the specifications of the main communication and radio standards.

This section lists the supported standard-compliant digital signals.

GSM/EDGE, EDGE Evolution

The "GSM/EDGE, EDGE Evolution" functionality generates signal waveforms in accordance with the GSM/EDGE standard. It is based on the GMSK and 8PSK modulation, and in accordance with the "EDGE Evolution" standard with simulation of higher order modulations.

For details, see the R&S WinIQSIM2 GSM/EDGE user manual.

Bluetooth® BR+EDR, LE

The Bluetooth® enhanced data rate functionality generates signal waveforms in accordance with the Bluetooth version 2.1+EDR and the latest Bluetooth low energy technology.

For details, see the R&S WinIQSIM2 Bluetooth EDR user manual.

TETRA Release 2

The "TETRA Release 2" functionality generates signal waveforms in accordance with the standard TETRA 2 ("Terrestrial Trunked Radio Release 2 ").

For details, see the R&S WinIQSIM2 "TETRA Release 2" user manual.

3GPP FDD incl. enhanced MS/BS tests, HSPA, HSPA+

The 3GPP FDD functionality generates signal waveforms in accordance with the WCDMA standard 3GPP FDD incl. enhanced MS/BS tests, HSPA, HSPA+.

For details, see the R&S WinIQSIM2 3GPP FDD user manual.

CDMA2000®

The CDMA2000 functionality generates signal waveforms in accordance with the CDMA2000 standard 3GPP2 C.S0002-C, version 1.0, May 2002 (release C).

For details, see the R&S WinIQSIM2 CDMA2000 user manual.

TD-SCDMA and TD-SCDMA enhanced features

The TD-SCDMA and TD-SCDMA functionality generates signal waveforms in accordance with the TD-SCDMA (3GPP TDD LCR) standard.

For details, see the R&S WinIQSIM2 TD-SCDMA user manual.

1xEV-DO Rev. A and Rev. B

The 1xEV-DO functionality generates signal waveforms in accordance with the CDMA2000 1xEV-DO ("Evolution-Data Optimized"), Rev. A and Rev. B. Standard.

For details, see the R&S WinIQSIM2 1xEV-DO Rev. A Rev. B user manual.

IEEE 802.11a/b/g/n, IEEE 802.11ac

The IEEE 802.11a/b/g/n and IEEE 802.11ac functionality generates signal waveforms in accordance with the Wireless LAN standards IEEE 802.11a/b/g/n/ac/p/j.

For details, see the R&S WinIQSIM2 IEEE 802.11a/b/g/n/ac/p/j user manual.

IEEE 802.11ad

The IEEE 802.11ad functionality generates signal waveforms in accordance with the Wireless LAN standards IEEE 802.11ad.

For details, see the R&S WinIQSIM2 IEEE 802.11ad user manual.

ECMA-368 IEEE 802.15 3a

The ECMA-368 IEEE 802.15 3a functionality generates UWB MB-OFDM signal waveforms to generate in accordance with the "WiMedia Alliance" and "MultiBand OFDM Alliance Unapproved Release Candidate Version 1.2".

For details, see the R&S WinIQSIM2 ECMA-368 IEEE 802.15 3a ("Ultra Wide Band") user manual.

IEEE 802.16 WiMAX™

The IEEE 802.16 WiMAX functionality generates signal waveforms in accordance with the IEEE 802.16 standard WiMAX.

For details, see the R&S WinIQSIM2 WiMAX user manual.

EUTRA/LTE Rel. 8, Rel. 9, Rel. 10, Rel. 11 Rel. 12

The functionality generates signal waveforms in accordance with the 3GPP standard EUTRA/LTE Rel. 8, Rel. 9, Rel. 10, Rel. 11 Rel. 12.

For details, see the R&S WinIQSIM2 EUTRA/LTE user manual.

GNSS

The GNSS (global navigation satellite system) functionality generates signal waveforms in accordance with the GPS, Galileo, GLONASS and COMPASS/BeiDou.

For details, see the R&S WinIQSIM2 GNSS user manual.

DVB-H/T

The DVB-H/T functionality generates signal waveforms in accordance with the DVB-H (Digital Video Broadcasting - "Transmission System for Handheld Terminals") standard.

For details, see the R&S WinIQSIM2 DVB-H/T user manual.

DAB / T-DMB

The DAB-H/T functionality generates signal waveforms in accordance with the "Digital Audio Broadcasting" (DAB) / "Terrestrial Digital Multimedia Broadcasting" (T-DMB) standard.

For details, see the R&S WinIQSIM2 DAB / T-DMB user manual.

NFC/EMV

The NFC/EVM functionality generates signal waveforms in accordance with the short-range wireless connectivity technology NFC-A/B/F and the EMV standard.

For details, see the R&S WinIQSIM2 NFC A/B/F user manual.

4.3 Common Functions and Settings in the Baseband

Basic signal generation settings that are common to many generation tasks, regardless of the selected baseband source or digital standard, are described here. If you, e.g. generate a signal according to a digital standard, check the specific description for settings that can deviate from the common settings.

4.3.1 Basics on Signals, Modulation Types and Filters

This section provides general information on common topics and basic principles.

4.3.1.1 Data and Signal Sources

This section describes the common characteristics of the signals used for generating the baseband waveform signal, irrespective of the selected digital standard or user-specific waveform. The provided selection in the dialogs depends on the parameter and corresponding standard. Some parameters are therefore not available in certain cases. Characteristics which are uniquely specific to particular standards are described in the corresponding user manuals.

For the generation of modulation signals, R&S WinIQSIM2 uses the following input signals:

- Modulation data
- Control signals

Internal Modulation Data

R&S WinIQSIM2 uses the following internal modulation data sources:

- **Data lists**
  Data lists are externally or internally created binary lists with modulation data. R&S WinIQSIM2 provides standard file select function for loading of existing data lists, creating internally new data lists or editing an existing one. Internally, data lists are created in the dedicated "Data List" editor (see Chapter 4.4.2.8, "Data List

Editor", on page 79). A separate file is created for each list and stored in the user-specific directory of R&S WinIQSIM2. The file name is user-defined; the file extension is `*.dm_iqd`.
Note: The maximum length of a data list is determined by the size of the data list memory (see data sheet). For instrument-specific data, see the data sheet of the respective instrument. There is no restriction on the number of lists that can be stored.
Settings for file handling, like transferring external data lists to the instrument, renaming of folders and files are accessed via the standard "File Manger" function (see also Chapter 9, "File and Data Management", on page 206).

- **Data patterns**
  You can use simple data patterns as internal modulation data, e.g. binary strings 0 or 1 ("All 0","All 1"), or variable bit strings with a maximum length of 64 bits.
- **PRBS data**
  The internal PRBS generators deliver pseudo-random binary sequences of differing length and duration. They are known as maximum length sequences, and are generated with the aid of ring shift registers with feedback points determined by the polynomial.
  The pseudo-random sequence from a PRBS generator is uniquely defined by the register number and the feedback. The Table 4-1 describes the available PRBS generators.

Table 4-1: Overview of PRBS generators

PRBS generator	Length in bits	Feedback to	GUI selection
9-bit	2^9 -1 = 511	Registers 4, 0	PRBS 9/PN9
11-bit	2^{11} -1 = 2047	Registers 2, 0	PRBS 11/PN11
15-bit	2^{15} -1 = 32767	Registers 1, 0	PRBS 15/PN15
16-bit	2^{16} -1 = 65535	Registers 5, 3, 2, 0	PRBS 16/PN16
20-bit	2^{20} -1 = 1048575	Registers 3, 0	PRBS 20/PN20
21-bit	2^{21} -1 = 2097151	Registers 2, 0	PRBS 21/PN21
23-bit	2^{23} -1 = 8388607	Registers 5, 0	PRBS 23/PN23

Example:

The example shows the diagram of a 9-bit generator with feedback to registers 4 and 0 (output). The generated serial data is converted internally, e.g 2 Bit/Symbol in case of QPSK.

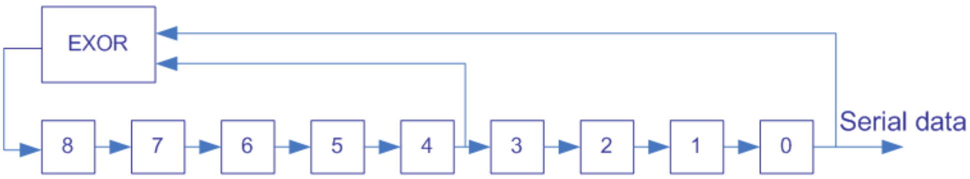
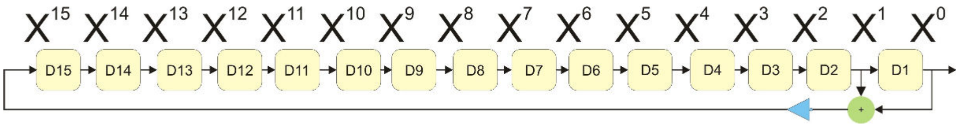


Figure 4-1: A 9-bit PRBS generator

For PRBS15 and PRBS23, a CCITT V.52-compliant data inversion is performed in the feedback path automatically as shown below:



Related settings:

- Chapter 4.4.2.3, "Data Source", on page 70
- Chapter 4.4.2.8, "Data List Editor", on page 79
- Chapter 4.4.2.9, "Control and Marker Lists Editor", on page 81
- Data Source selection in the dialogs of the firmware options

Control Signals

The following control signals are processed by R&S WinIQSIM2:

- "Burst Gate" for power ramping
- "Level Attenuation" for power ramping
- "CW/Mod" for controlling the CW (continuous wave) mode

A dedicated internal "Control Data Editor" is provided for defining the control signals. Refer to Chapter 4.4.2.9, "Control and Marker Lists Editor", on page 81 for a description on the provided settings.

Continuous Wave Mode

"CW" for controlling the CW (continuous wave) mode is not used in R&S WinIQSIM2. However, a control list generated for a Rohde & Schwarz instruments can contain CW controls.

A separate file with the file extension `*.dm_iqc` is created for each defined control signal.

Power Ramping and Level Attenuation

The R&S WinIQSIM2 uses the two control signals "Burst Gate" and "Lev_Att" to trigger the power ramping and level attenuation functions.

The application internally generates control signals as configured in Chapter 4.4.2.9, "Control and Marker Lists Editor", on page 81.

- *Burst gate control signal*
  The "Burst Gate" signal is a rectangular pulse signal with variable low and high periods. Signal generation is restricted to the gate high periods. If the power ramping function is enabled, each transition between two gate periods of the "Burst Gate" signal triggers the generation of a ramp. Further settings define the form and the steepness of this ramp, see "Impact of the Power Ramping Settings on the Generated Signal" on page 58.
- *Level attenuation control signal*

The "Lev_Att" signal is a rectangular pulse signal with variable low and high periods. Level attenuation is applied, if the "Lev_Att" signal is high. If level attenuation is enabled, the modulation signal level is attenuated by a defined value.

Refer to Chapter 4.4.2.6, "Power Ramp Control Settings", on page 76 for a description of the provided settings for level attenuation and power ramping.

Possible applications

- Use the "Level Attenuation" function to simulate radio stations located at various distances.
- Use the "Power Ramp" function if it is necessary to control the RF output signal envelope synchronously, e.g. by the generation of TDMA signals.

Both the GSM/EDGE and the TD-SCDMA firmware options are equipped with embedded power ramping function. In the GSM/EDGE standard for example, a maximum of 7 different level attenuation values can be defined and allocated separately to the 8 slots independently of one another.

Impact of the Power Ramping Settings on the Generated Signal

The Figure 4-2 explains the power ramping function in principle. The "Burst Gate" signal defines the start of the rising and falling edges of the envelope of the output signal, and the "Lev Att" signal defines the start and end of level attenuation. The signal level during the attenuation period is a configurable value.

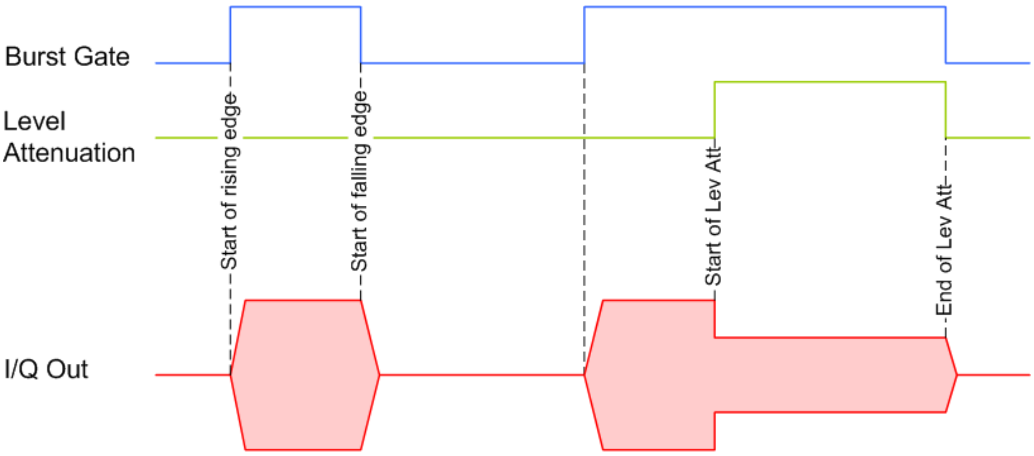


Figure 4-2: Signal behavior when power ramping and level attenuation are enabled

Several parameters are provided for precise definition of the form and the steepness of ramp. The Figure 4-3 depicts the impact of the provided settings.

- Ramp function: defines the shape of the rising and falling edges
- Ramp time: defines the duration of the rising and the falling ramp
- Rise/fall delay: offsets the falling edge of the envelope at the beginning/end of a burst

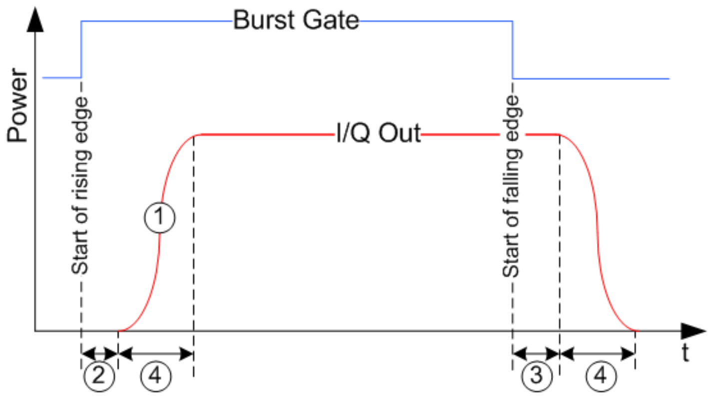


Figure 4-3: Impact of the ramp settings

1 = "Ramp Function"
2, 3 = "Rise Delay", "Fall Delay"
4 = "Ramp Time"

4.3.1.2 Marker Signals

R&S WinIQSIM2 generates user-definable marker information which can be processed by a Rohde & Schwarz instrument to provide the appropriate marker signals at the signal output.

You can define up to four marker signals according to the selection parameters required for the respective digital signal.

Marker Modes

The marker mode is a characteristic for the shape and the periodicity of the marker. R&S WinIQSIM2 provides several different modes to define different marker signals. Most of them are specific for each of the digital standards. This section focuses only on the general commonly available marker signals. Generally, the marker signal can change from "On" (high) to "Off" (low) state or vice versa after some period of time. R&S WinIQSIM2 provides various ways to describe the marker signal. Use the method that best suits your needs.

Marker Mode Restart

The generated marker signal is a single "On" pulse. The rising edge of this pulse is generated at the signal generation start as well as at each subsequent signal restart time. This marker can be used to monitor the effects of the selected trigger, e.g. trigger causing restarts of the signal generation.

Marker Mode Pulse

Periodic marker with consecutive On and Off periods of equal length. The first On period starts at the beginning of the first generated sample/symbol. The marker frequency is defined by a "Divider". The frequency is derived as follows:

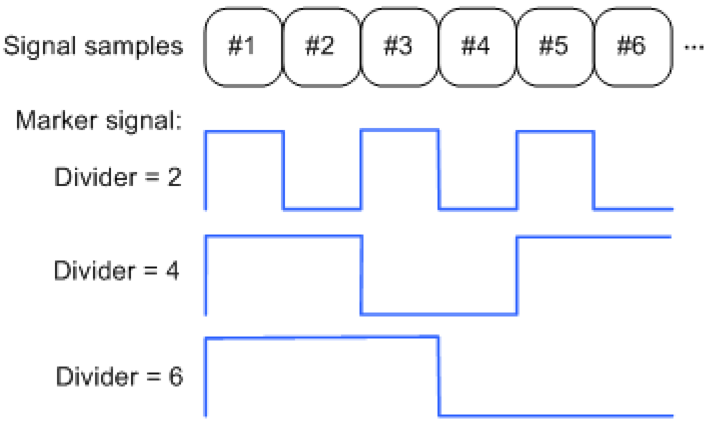
<Frequency> = "Symbol/Sample Rate" / "Divider", respectively

<Frequency> = "Sampling Frequency" / "Divider".

Example:

Symbol Rate = 1 Msym/s, Divider = 2

The marker frequency is 500 kHz, corresponding to a marker period of 2 us. Each On and Off period has a length of 1 us, corresponding to one symbol period. With a divider of 4 (6, 8 ...), the length of each On and Off period is increased to 2 (3, 4, ...) symbol periods.

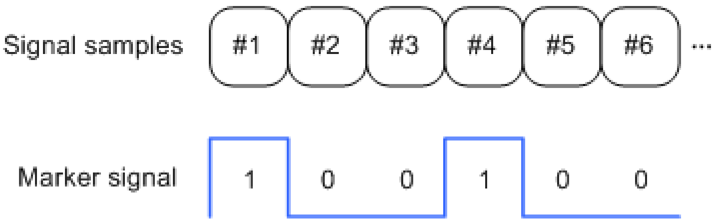


Marker Mode Pattern

Periodic marker where each period is defined by a bit pattern with a maximum length of 64 bits. A "1" ("0") in the pattern denotes an On (Off) signal segment with a duration of one sample/symbol period.

Example:

In the following example, the marker signal is defined by a pattern *100100...*.

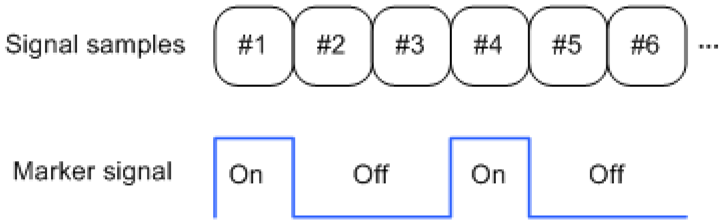


Marker Mode ON/OFF Ratio

Similar to "Pulse" but with independent lengths of the On and Off periods. The length of the periods is entered as a number of symbols/sample periods.

Example:

In the following example, the marker signal is defined by an "On Time" = 1 and "Off Time" = 2.



4.3.1.3 Supported Modulation Types

R&S WinIQSIM2 supports a range of predefined digital modulation types.

In the communication techniques, the commonly used digital modulation schemes are based on keying. From the several existing keying techniques, R&S WinIQSIM2 supports ASK (amplitude shift keying), FSK (frequency shift keying), PSK (phase shift keying) and QAM (quadrature amplitude modulation). The digital modulation procedure is described by mapping, i.e. by the assignment of I and Q values (PSK and QAM) or frequency shifts (FSK) to every modulation symbol. The resulting modulated signal is graphically represented by a constellation diagram, in that each possible symbol is represented by a discrete point on a complex plane. The number of used bits per symbol is a modulation parameter. The exact position of the symbols on the constellation diagram is determined by the used coding and can be influenced by additionally applied rotation.

Most of the provided modulation schemes are implemented according to a communication standard. The QAM procedures 16QAM, 32QAM, 64QAM for instance have been produced in accordance with ETSI standard ETS 300429 for digital video broadcasting (DVB). The QAM procedures 256QAM and 1024QAM are not specified in this standard, but have been produced according to the same basic principles.

In the case of all FSK procedures, you can set the symbol rate f_{SYMB} up to a maximum value (see data sheet). The frequency deviation (FSK deviation) of the MSK modulation is permanently set to ¼ of the symbol rate.

In addition to the common modulation schemes, a variable FSK modulation with definable deviation per symbol is available. For even greater flexibility, you can apply a user defined modulation mapping , see User mapping.

Predefined modulation types

Refer to Chapter 4.4.4.1, "Predefined Modulation Types", on page 92 for an overview of the available modulation types.

User mapping

A user defined modulation mapping file can also be selected as modulation mapping source. The user modulation mapping file must have extension `*.vam` and can be created with the Rohde & Schwarz software tool mapping wizard. The mapping wizard (mapwiz) is a tool from Rohde & Schwarz designed for editing modulation schemes (e.g. QPSK, 32QAM). Its main purpose is the assignment of logical symbol numbers to constellation points and the selection of modulation-specific parameters. In addition, it supports the creation of nearly any arbitrarily chosen constellation diagram. The output of mapwiz is a mapping file (`*.vam`) that can be imported to a R&S WinIQSIM2. The program was developed on a 32-bit Microsoft Windows platform under MATLAB. For more information, refer to the description "Introduction to "mapwiz" Mapping Editor" on the Rohde&Schwarz Internet page.

The remote commands required to define the modulation settings are described in Chapter 12.9, "SOURce:BB:DM Subsystem", on page 324.

Related settings:

- Chapter 4.4.2, "Custom Digital Modulation Settings", on page 67

4.3.1.4 Supported Coding Schemes

Coding is a technique used to improve the signal properties and signal reception and is required only when using some types of modulation. In general, the coding schemes are applied before modulation, i.e. the modulation symbols are coded directly before I and Q values or frequency shifts are assigned. Hence, the applied coding is directly related to the selected modulation methods and explains why coding schemes are not freely combinable with modulation methods.

Refer to Chapter 4.4.4.2, "Common Coding Algorithms", on page 95 for overview on the available coding combinations. This section also defines the modulation types for which the various coding procedures can be used.

Related settings:

- "Coding" on page 69

4.3.1.5 Supported Baseband Filters

In the wireless transmission technique, filters are applied to shape the baseband signal before it is modulated on the RF. The selected baseband filter type and shape affect the baseband signal, especially while generating broadband signals. If the filter is too narrow, the signal is cut by the filter. If the filter is too wide, the signal could be distorted by some unwanted signals.

To fulfill the range of requirements, R&S WinIQSIM2 offers a large selection of predefined baseband filters. The predefined filters are designed for the special spectrum characteristics of the different communication standards. However, depending on the selected filter form additional filter parameters are provided for more precise adjustment of the filter characteristic, like more steeper edges or customization of the transition bandwidth. For more information on the provided settings, refer to "Impact of the Filter Parameters" on page 63.

The selection of user defined filter offers even more flexibility. The later is a useful interface while filters with complex or proprietary form are required. For more information, refer to "User filter" on page 63.

Predefined baseband filters

Refer to Chapter 4.4.4.3, "Predefined Baseband Filters", on page 97 for an overview of the available baseband filters.

User filter

The user filter file must have the extension `*.vaf` and can be created with the Rohde & Schwarz software tool filter wizard.

The filter wizard (filtwiz) is a tool from Rohde & Schwarz designed for creating filter files that can be imported on a R&S WinIQSIM2. Its main purpose is the conversion of user-defined finite impulse response (FIR) filters into the filter format (`*.vaf`). Beyond this filtwiz provides designs for standard filters, e.g. "Root Raised Cosine", Gaussian.

The program was developed on a 32-bit Microsoft Windows platform under MATLAB.

For more information, refer to the description "Introduction to "filtwiz" Filter Editor" on the Rohde & Schwarz Internet page.

The remote commands required to define the filter settings are described in Chapter 12.9, "SOURce:BB:DM Subsystem", on page 324 and the corresponding section in the user manual of each firmware option.

Related settings:

- Chapter 4.4.2.5, "Filter Settings", on page 74
- Filter settings in the dialogs of the firmware options

Impact of the Filter Parameters

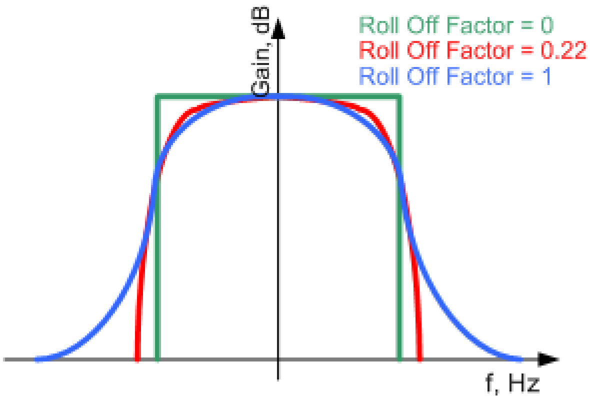
The following is a simple description of the filter parameters and the way they affect the main filter characteristics. Changing filter parameters is an effective way to ensure that the entire bandwidth of the desired signal is allowed to pass and adjust the filter form to reach the spectrum mask requirements

Cut Off Frequency

The cut-off frequency or corner frequency is a filter characteristic that defines the frequency at the 3 dB down point. This frequency is bound to the transition band; here the filter characteristic changes form the passband to the stopband, where the signal is suppressed.

Rolloff Factor

The rolloff factor is a measure for the excess bandwidth compared to the ideal bandwidth of a "brick like" filter. The roll off factor affects the steepness of the filter flanks. A "Rolloff Factor" = 0 would result in the steepest theoretically possible flanks ; values near to 1 make the flanks more flat.



Passband

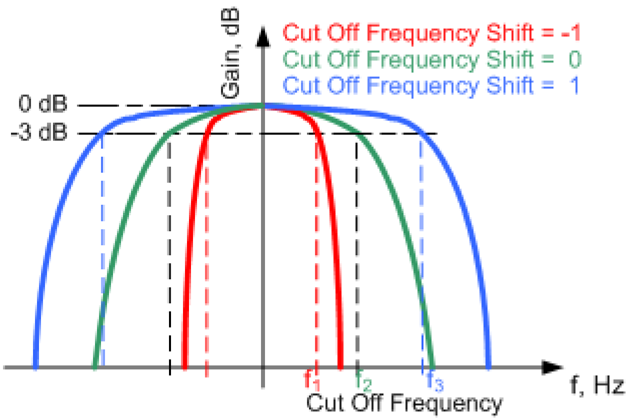
The passband describes the frequency span that the filter passes unchanged. The total passband of a filter is given as follows:

```
Bandwidth = (1 + "Roll Off Factor") * "Symbol Rate"
```

Cut Off Frequency Shift

The "Cut Off Frequency Shift" affects the cut-off frequency in the way that the filter flanks are "moved" and the passband increases by "Cut Off Frequency Shift"*"Sample Rate":

```
Cut Off Frequency = (1 + "Cut Off Frequency Shift") * "Sample
Rate"
```



- A "Cut Off Frequency Shift" = -1 results in a very narrow-band filter
- Increasing the value up to 1 makes the filter more broad-band
- By "Cut Off Frequency Shift" = 0, the -3 dB point is at the frequency determined by the half of the selected "Sample Rate".

4.3.1.6 Methods for Optimizing the Crest Factor

Communication standards utilizing higher order modulation techniques or using multiple carrier and complex signals consisting of the signals of more than one digital standard may feature a high crest factor. The signals of some digital standards may have very high crest factors also particularly with many channels and long sequences.

The **crest factor** represents the ratio of the peak voltage value to the RMS voltage value, i.e. the peak to average ratio (**PAR**). The higher the crest factor and the resulting dynamics of a signal, the greater the requirement for a power amplifier fed by the signal to be linear. A very high crest factor arises for instance, when in a multicarrier signal the carriers feature an identical start phase. This results form the fact that the carriers are periodically superposed that leads to very high peak voltages in relation to the RMS voltage values.

High crest factors entail two basic problems:

- The nonlinearity of the power amplifier (compression) causes intermodulation which expands the spectrum (spectral regrowth).
- Since the level of the D/A converter is relative to the maximum value, the average value is converted with a relatively low resolution. This results in a high quantization noise.

Both effects increase the adjacent-channel power.

Direct approaches

At the individual signal generation stages, R&S WinIQSIM2 offers different direct approaches aimed to reduce the crest factor. While the corresponding parameters are enabled, the implemented algorithms ensure minimizing the crest factor or achieving of predefined target crest factor by applying of automatic settings. Methods of reducing the crest factor differ with regard to both the optimization achievable and the time required for computation.

The provided crest factor reduction methods include:

- internal calculation of optimized carrier phases for the individual carriers in a multi carrier signal
- automatic calculation of the carrier start phases in a multicarrier continuous wave signal

Applying clipping and filtering

Another common and simple approach for achieving a lower PAR is the combination of clipping and filtering. In several of the digital standards, like 3GPP FDD, CDMA2000 etc., R&S WinIQSIM2 provides the possibility to enable baseband clipping and to select the baseband filter and, when applicable, to adjust the filter characteristics.

- **Clipping** is a technique that applies a wanted distortion to the signal.
  The principle includes specifying a threshold, finding out the signal peaks once the defined limits are exceeded and clipping them off. The level limit is specified as a percentage of the highest peak value. Because clipping is done prior to filtering, the procedure does not influence the spectrum. The error vector magnitude (EVM) however increases.

  R&S WinIQSIM2 offers two clipping modes:
  - **Vector | I + jq |**
    The clipping limit is related to the amplitude | I + jq |. The I and Q components are mapped together, the angle is retained.

– **Scalar | I | + | q |**
  The clipping limit is related to the absolute maximum of all the I and Q values | I | + | q | . The I and Q components are mapped separately, the angle changes.

  However, signal clipping not only changes the peak value but also the average value and the effect on the crest factor is unpredictable.

- **Filtering** is applied subsequently. The used filters are specially designed and should filter out the distortion.

4.3.2 Common Settings

The start dialog of each digital standard follows a repeating dialog structure that comprises the tabs "General" and "Marker".

The "General" tab comprises the primary settings of the standard, the functions for storing and recalling settings and provides access to further functions and dialogs, like the "Filter" settings.

The "Marker" tab comprises the settings related to the corresponding function.

This section focuses on the available settings. For more information, refer to Chapter 4.3.1, "Basics on Signals, Modulation Types and Filters", on page 55.

4.4 Generating Custom Digital Modulated Signals

R&S WinIQSIM2 can generate digital modulation signals with user-definable characteristics. The baseband filtering and the symbol rate can be set within wide limits.

4.4.1 About the Custom Digital Modulation

An introduction to the supported filter, modulation and coding schemes is provided in:

- Chapter 4.3.1.5, "Supported Baseband Filters", on page 62
- Chapter 4.3.1.3, "Supported Modulation Types", on page 61
- Chapter 4.3.1.4, "Supported Coding Schemes", on page 62.

Interdependency between selected modulation type and coding scheme and handling of conflicting settings

The applied coding is directly related to the selected modulation methods. The available coding schemes listed in Chapter 4.4.4.2, "Common Coding Algorithms", on page 95 are not freely combinable with modulation methods.

Obviously, having selected a modulation procedure, not every combination of the further modulation parameters "Symbol Rate" and "Coding" is possible. These restrictions inevitably result in conflicting settings if you change a parameter and leads to a prohibited combination.

An inappropriate change of a parameter triggers a settings conflict. A conflicting setting is indicated by a message on the "Info" line in the display. Although R&S WinIQSIM2 displays the selected settings, the generated modulation signal does not correspond to this display. The displayed message disappears as soon as a conflict-free setting is selected.

Refer to Chapter 13, "Troubleshooting and Error Messages", on page 392 for a list of the possible settings conflicts and messages in digital modulation.

4.4.2 Custom Digital Modulation Settings

The "Custom Digital Modulation" dialog enables you to select the data source, standard, symbol rate, coding, modulation type and filter.

To access the "Custom Digital Modulation" settings:

► Select "Baseband > Custom Digital Mod".

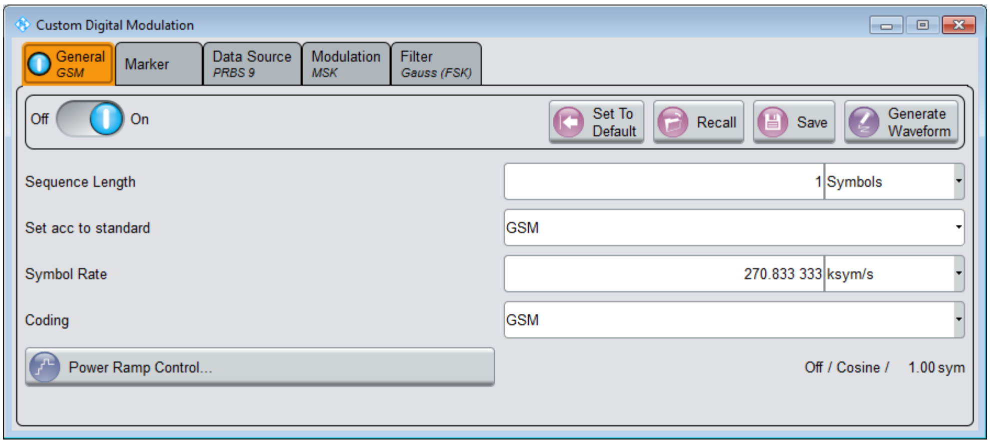
The dialog is divided into several tabs. In each case, the current setting is displayed in the tab name.

The remote commands required to define these settings are described in Chapter 12.9, "SOURce:BB:DM Subsystem", on page 324.

4.4.2.1 General Settings

To access the common settings:

► Select "Baseband > Custom Digital Mod > General".



This tab provides access to the default and the Save/Recall settings, to a quick selection of a digital modulation according to a predefined communication standard, adjusting the symbol rate and the coding.

State
Activates digital modulation. Switching on digital modulation turns off all other digital standards.

Remote command:
`[:SOURce<hw>]:BB:DM:STATe` on page 327

Set To Default
Sets all relevant parameters to default, see Table 4-2.

Remote command:
`[:SOURce<hw>]:BB:DM:PRESet` on page 325

Save/Recall
Accesses the "Save/Recall" dialog, i.e. the standard instrument function for storing and recalling the complete dialog-related settings in a file. The provided navigation possibilities in the dialog are self-explanatory.

The file name and the directory are user-definable; the file extension is however predefined (`*.dm`).

See also Chapter 9, "File and Data Management", on page 206.

Remote command:
`[:SOURce<hw>]:BB:DM:SETTing:CATalog?` on page 349
`[:SOURce<hw>]:BB:DM:SETTing:DELete` on page 350
`[:SOURce<hw>]:BB:DM:SETTing:LOAD` on page 350
`[:SOURce<hw>]:BB:DM:SETTing:STORe` on page 350

Generate Waveform File
With enabled signal generation, triggers the instrument to store the current settings as an ARB signal in a waveform file. Waveform files can be further processed as multi carrier or multi segment signals.

The file name and the directory it is stored in are user-definable; the predefined file extension for waveform files is `*.wv`.

Remote command:
`[:SOURce<hw>]:BB:DM:WAVeform:CREate` on page 328

Sequence Length
Sets the sequence length of the signal in symbols. The signal is calculated in advance, saved as waveform file, and output in the arbitrary waveform generator of the selected instrument.

Note: The product of "Sequence Length" * "Oversampling" must not exceed the maximum number of samples of the arbitrary waveform generator.

Remote command:
`[:SOURce<hw>]:BB:DM:SLENgth` on page 327

Set acc. Standard
Selects a predefined communication standard. A subset of parameters is automatically adjusted: "Modulation Type", "Symbol Rate", "Filter" and "Coding".

A subsequent modification of one of these parameters, sets the standard to "User". Use the "Save/Recall" function to store and recall customized settings.

Refer to Table 4-3 for an overview of the available standards and the associated settings of the modulation parameters.

Remote command:
`[:SOURce<hw>]:BB:DM:STANdard` on page 327

Symbol Rate
Selects the symbol rate. The value range of this parameter depends on the selected modulation type; the range is automatically redefined. R&S WinIQSIM2 generates an error message if the selected symbol rate is outside of the redefined range and sets the symbol rate to the maximum allowed value for the new modulation.

Remote command:
`[:SOURce<hw>]:BB:DM:SRATe` on page 326

Coding
Selects the coding (see Chapter 4.3.1.4, "Supported Coding Schemes", on page 62).

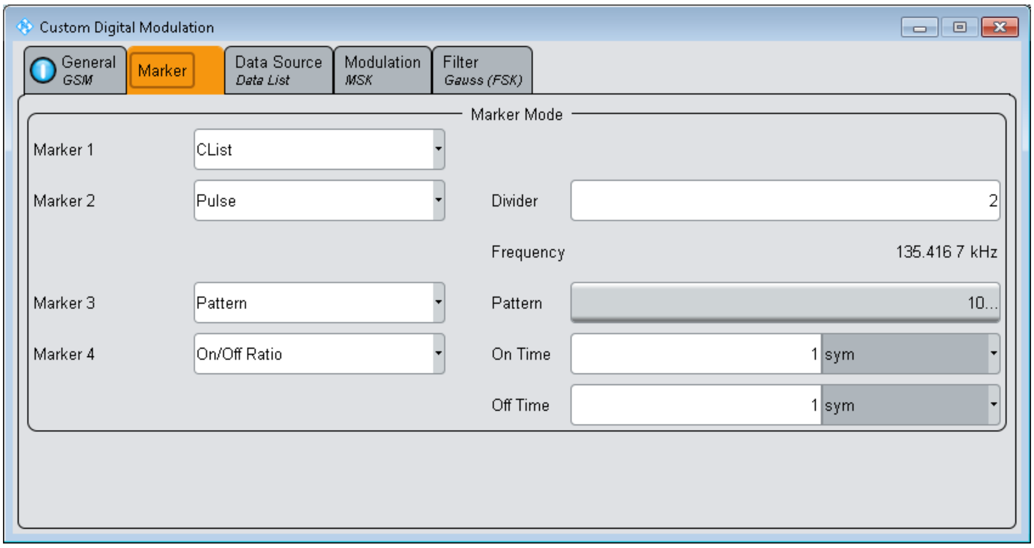
The dialog offers only the coding settings that are permissible for the selected modulation type. All other coding methods are grayed out. A subsequent modification to a modulation type for which the selected coding is not available, automatically disables the coding ("Coding = Off").

Remote command:
`[:SOURce<hw>]:BB:DM:CODing` on page 336

Power Ramp Control
Accesses the power ramp control dialog, see Chapter 4.4.2.6, "Power Ramp Control Settings", on page 76.

4.4.2.2 Marker Settings

This tab provides access to the settings necessary to select and configure the marker mode.



Marker Mode
Marker configuration for up to four marker channels. The settings are used to select the marker mode defining the shape and periodicity of the markers. The contents of the dialog change with the selected marker mode; the settings are self-explanatory.

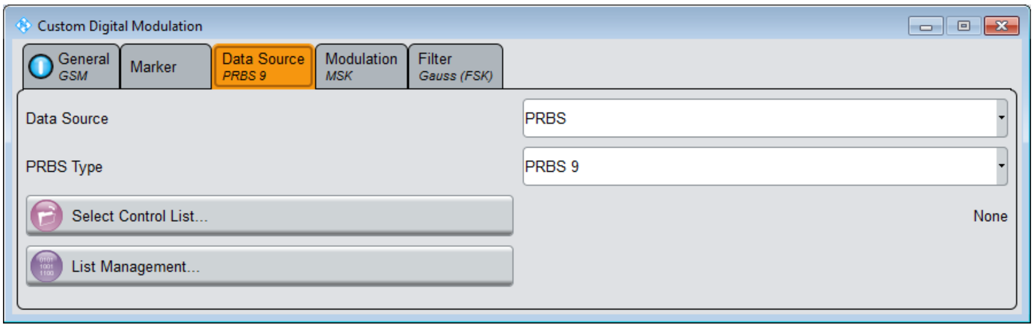
Use the settings to define the shape and periodicity of the markers. See Chapter 4.3.1.2, "Marker Signals", on page 59 for description of the regular marker signals.

Remote command:
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:MODE on page 328
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:DIVider on page 330
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:FREQuency? on page 331
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PATTern on page 330
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:OFFTime on page 329
[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:ONTime on page 329

4.4.2.3 Data Source

This tab provides access to the settings necessary to select and configure the data source, like access to data and list editors or direct selection of PRBS data.

For an overview of the supported data sources, refer to Chapter 4.3.1.1, "Data and Signal Sources", on page 55.



Data Source

Selects the data source (see "Internal Modulation Data" on page 55).

The following data sources are available:

"All 0, All 1" A sequence containing 0 data or 1 data is internally generated

"PRBS, PRBS Type"
Selects internally generated PRBS data in accordance with the IUT-T. Use the parameter "PRBS Type" to define the length.

Remote command:
[:SOURce<hw>]:BB:DM:PRBS[:LENGth] on page 325

"Pattern" Use the "Pattern" box to define a bit pattern with a maximum length of 64 bits.

"Data List"
Uses binary data from a data list, see Select Data List.

Remote command:
[:SOURce<hw>]:BB:DM:DLISt:CATalog? on page 341
[:SOURce<hw>]:BB:DM:DLISt:SELect on page 347

Remote command:
[:SOURce<hw>]:BB:DM:SOURce on page 325

Select Data List

Accesses the standard "Select Data List" browser dialog to select a data list file.

To load an existing data list, select the list file *.dm_iqd and confirm with "Select".

See also Chapter 4.4.3.2, "How to Create and Assign a Data List", on page 87

Remote command:
[:SOURce<hw>]:BB:DM:DLISt:SELect on page 347
[:SOURce<hw>]:BB:DM:DLISt:CATalog? on page 341
[:SOURce<hw>]:BB:DM:DLISt:COPY on page 342
[:SOURce<hw>]:BB:DM:DLISt:DELete on page 346

Select Control List

Accesses the standard "Select Control List" browser dialog to select a control list file.

To load an existing control list, select the list file `*.dm_iqc` and confirm with "Select".

Control lists can be generated in the "Control and Marker Lists" editor, see Chapter 4.4.2.9, "Control and Marker Lists Editor", on page 81.

Irrespective of the way they are created, control lists are not automatically assigned (see "To assign and activate control signals from a control list" on page 87).

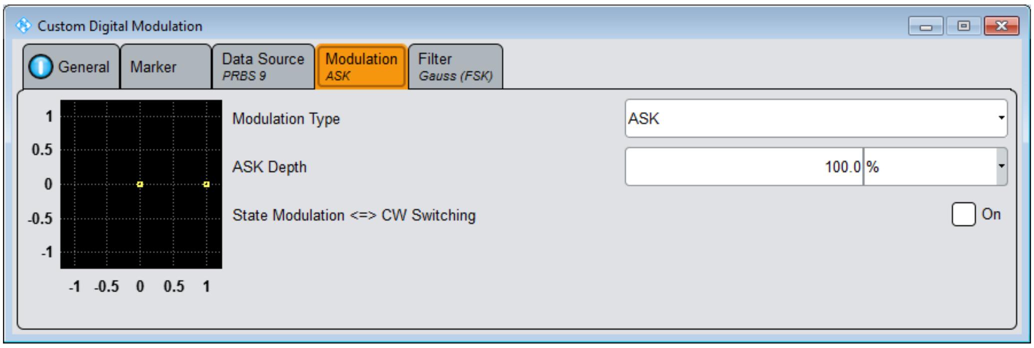
Remote command:
`[:SOURce<hw>]:BB:DM:CLISt:CATalog?` on page 341
`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 345
`[:SOURce<hw>]:BB:DM:CLISt:COPY` on page 342
`[:SOURce<hw>]:BB:DM:CLISt:DELete` on page 344

List Management...
Accesses the standard "List Management" dialog for managing data and control lists, see Chapter 4.4.2.7, "List Management Settings", on page 77.

4.4.2.4 Modulation Settings

This tab provides access to the modulation settings, e.g modulation type, FSK deviation or modulation depth. The dialog shows the theoretical constellation diagram of the selected modulation.

This section focuses on the available settings. For background information on how these settings affect the signal and the filter characteristics, refer to Chapter 4.3.1.3, "Supported Modulation Types", on page 61.



Modulation Type
Selects a modulation type. The associated symbol mapping is displayed.

If the selected "Coding" is not allowed with the configured modulation type, the value of the parameter Coding is set to "Off".

Refer to Table 4-4 for an overview of the allowed combinations.

Remote command:
`[:SOURce<hw>]:BB:DM:FORMat` on page 336

Load User Mapping
Provides access to the "Select List File User Mapping" dialog to select the mapping table (see "User mapping" on page 61). The dialog provides all standard file management functions.

Remote command:
`[:SOURce<hw>]:BB:DM:MLISt:SELect` on page 349
`[:SOURce<hw>]:BB:DM:MLISt:CATalog?` on page 342
`[:SOURce<hw>]:BB:DM:MLISt:DELete` on page 348

ASK Depth
Sets the modulation depth *m* for ASK modulation.

$m = (Amplitude_{max} - Amplitude_{min}) / (Amplitude_{max} + Amplitude_{min})$

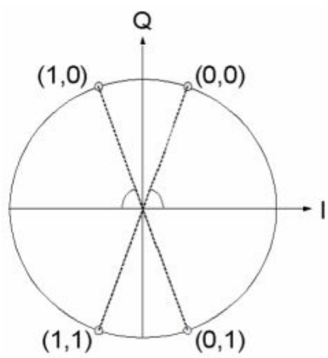
Remote command:
`[:SOURce<hw>]:BB:DM:ASK:DEPTh` on page 335

FSK Deviation
Sets the frequency deviation for FSK modulation. The range of values depends on the selected Symbol Rate (see data sheet).

Whenever "MSK" is selected, the deviation corresponds to 1/4 of the symbol rate and cannot be changed.

Remote command:
`[:SOURce<hw>]:BB:DM:FSK:DEViation` on page 337

Angle Alpha
For AQPSK modulation, sets the angle alpha between the point (0,0) and the I axis.



Remote command:
`[:SOURce<hw>]:BB:DM:AQPSk:ANGLe` on page 335

FSK Type
(Variable FSK only)

Selects the FSK modulation type for selection "Variable FSK".

Available are 4FSK, 8FSK and 16FSK.

Remote command:
[:SOURce<hw>]:BB:DM:FSK:VARiable:TYPE on page 338

Deviation xxxx
(Variable FSK only)

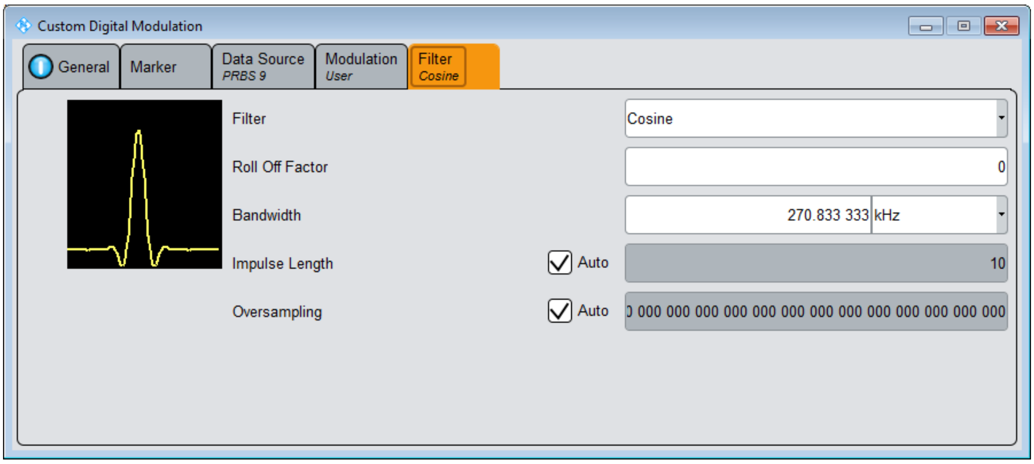
Sets the deviation of the associated symbol. The number of symbols depends on the selected modulation type. The value of each symbol is indicated in binary format.

Remote command:
[:SOURce<hw>]:BB:DM:FSK:VARiable:SYMBol<ch0>:DEViation on page 337

4.4.2.5 Filter Settings

This tab provides access to the filter settings, like filter type and if available further filter settings. A simplified diagram shows the filter characteristic of the selected filter.

This section focuses on the available settings. For background information on how these settings affect the signal and the filter characteristics, refer to Chapter 4.3.1.5, "Supported Baseband Filters", on page 62.



Filter
Selects the baseband filter.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:TYPE on page 334

Filter Parameter
Sets the corresponding filter parameter.

The filter parameter offered ("Roll Off Factor" or "B x T") depends on the currently selected filter type.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25 on page 333
[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine[:ROLLoff] on page 332
[:SOURce<hw>]:BB:DM:FILTer:PARameter:GAUSs on page 332
[:SOURce<hw>]:BB:DM:FILTer:PARameter:PGAuss on page 333
[:SOURce<hw>]:BB:DM:FILTer:PARameter:RCOSine on page 333
[:SOURce<hw>]:BB:DM:FILTer:PARameter:SPHase on page 333

Cut Off Frequency Factor
(available for filter parameter Lowpass and APCO25 (LSM) only)

Sets the value of the cut off frequency factor. The cut off frequency of the filter can be adjusted to reach spectrum mask requirements.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASs on page 332
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASSEVM on page 332
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:GAUSs on page 334
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:LOWPass on page 334

Bandwidth
Determines the bandwidth of the cosine filter, so that the function in `H(f) = 0` is fulfilled for `f >= (1 + RollOff)* Bandwidth /2`.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine:BANDwidth on page 334

Impulse Length
Displays the number of filter taps. If check box "Auto" is activated, the most sensible parameter value is used. The value depends on the coherence check. If the check box is deactivated, you can set the value manually.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:ILENgth:AUTO on page 332
[:SOURce<hw>]:BB:DM:FILTer:ILENgth on page 331

Oversampling
Determines the upsampling factor. If check box "Auto" is activated, the most sensible parameter value is used. The value depends on the coherence check. If the check box is deactivated, you can set value manually.

Remote command:
[:SOURce<hw>]:BB:DM:FILTer:OSAMpling:AUTO on page 332
[:SOURce<hw>]:BB:DM:FILTer:OSAMpling on page 332

Load User Filter

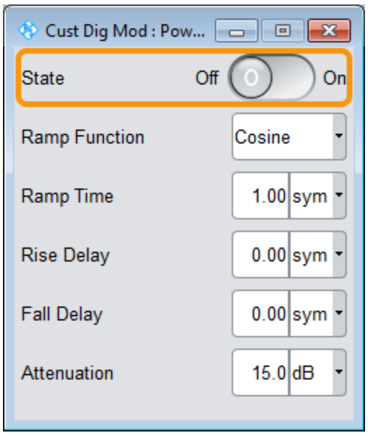
Accesses the "Select User Filter" dialog for selecting a user-defined filter file with extension `*.vaf` (see "User filter" on page 63). The dialog provides access to the standard file management functions, like store, load, delete.

Remote command:
`[:SOURce<hw>]:BB:DM:FLISt:SELect` on page 348
`[:SOURce<hw>]:BB:DM:FLISt:CATalog?` on page 342
`[:SOURce<hw>]:BB:DM:FLISt:DELete` on page 347

4.4.2.6 Power Ramp Control Settings

To access these settings:

► Select "Baseband > Custom Digital Modulation > General > Power Ramp Control".



The dialog provides access to the settings used to configure the power ramping. It includes the source for the necessary control signals "Burst" and "Lev_Att", the form of the ramp function and the applied attenuation (see "Power Ramping and Level Attenuation" on page 57).

Power ramping is possible up to a symbol rate of 5 MHz. A higher symbol rate disables the power ramping automatically and an error message is output.

State

Enables/disables power ramping.

Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp[:STATe]` on page 340

Ramp Function
Selects the ramp function that describes the shape of the rising and falling edges during power ramp control, see Figure 4-3.

"Linear"	The transmitted power rises and falls in a linear fashion.
"Cosine"	The transmitted power rises and falls with a cosine-shaped edge. This setting causes a more favorable spectrum than the "Linear" setting.

Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp:SHAPe` on page 339

Ramp Time
Enters the power ramping rise time and the fall time for a burst. The setting is expressed in symbols. See also Figure 4-3.

Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp:TIME` on page 339

Rise Delay
Sets the offset in the rising edge of the envelope at the start of a burst, see Figure 4-3.

A positive value causes a delay (the envelope length decreases), and a negative value causes an advance (the envelope length increases). The setting is expressed in symbols.

Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp:RDELay` on page 339

Fall Delay
Sets the offset in the falling edge of the envelope at the end of a burst, see Figure 4-3.

A positive value causes a delay (the envelope length increases), and a negative value causes an advance (the envelope length decreases). The setting is expressed in symbols.

Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp:FDELay` on page 339

Attenuation
Determines the level by which the average signal level is attenuated during the signal attenuation period, during the time the "Lev_Att" signal is high. See also Figure 4-2.

For information about the required control signal LEV_ATT, refer to "Power Ramping and Level Attenuation" on page 57.

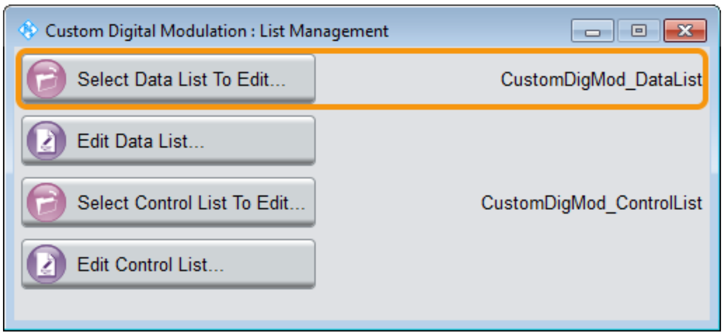
Remote command:
`[:SOURce<hw>]:BB:DM:PRAMp:ATTenuation` on page 338

4.4.2.7 List Management Settings

To access the list management dialog:

1. Select "Baseband > Custom Digital Mod > Data Source > Data List...".

2. Select "List Management...".



The dialog provides access to the respective file functions and editors for selecting, creating or editing data and control lists.

Select Data List To Edit...

Enables you to select or create a data list file, and provides access to the file manager.

You can perform the following tasks

- "New List"
  To create a file, navigate to the target folder, enter the file name and confirm with "Save". R&S WinIQSIM2 automatically assigns the extension `.dm_iqd` to the file name.
- "Select List"
  To load an existing data list file, navigate to the target folder, select the file `*.dm_iqd` and confirm with "Select".
- "File Manager"
  To perform standard file management functions, like create directories, move, copy, delete files and/or directories, use the standard "File Manager" function, see Chapter 9.6, "Using the File Manager", on page 217.

See also Chapter 4.4.3.2, "How to Create and Assign a Data List", on page 87

Remote command:
`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 347
`[:SOURce<hw>]:BB:DM:DLISt:CATalog?` on page 341
`[:SOURce<hw>]:BB:DM:DLISt:COPY` on page 342
`[:SOURce<hw>]:BB:DM:DLISt:DELete` on page 346

Edit Data List ...

Accesses the data list editor, see Data List Editor.

Select Control List

Enables you to select or create a control list file, and provides access to the file manager.

You can perform the following tasks

- "New List"
  To create a file, navigate to the target folder, enter the file name and confirm with "Save". R&S WinIQSIM2 automatically assigns the extension `.dm_iqc` to the file name.
- "Select List"

To load an existing control list file, navigate to the target folder, select the file `*.dm_iqc` and confirm with "Select".

- "File Manager"
  To perform standard file management functions, like create directories, move, copy, delete files and/or directories, use the standard "File Manager" function, see Chapter 9.6, "Using the File Manager", on page 217.

Irrespective on the way they are created, control signals are not automatically assigned (see "To assign and activate control signals from a control list" on page 87).

Remote command:
`[:SOURce<hw>]:BB:DM:CLISt:CATalog?` on page 341
`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 345
`[:SOURce<hw>]:BB:DM:CLISt:COPY` on page 342
`[:SOURce<hw>]:BB:DM:CLISt:DELete` on page 344

Edit Control List ...
Accesses the control and marker list editor, see Chapter 4.4.2.9, "Control and Marker Lists Editor", on page 81.

4.4.2.8 Data List Editor

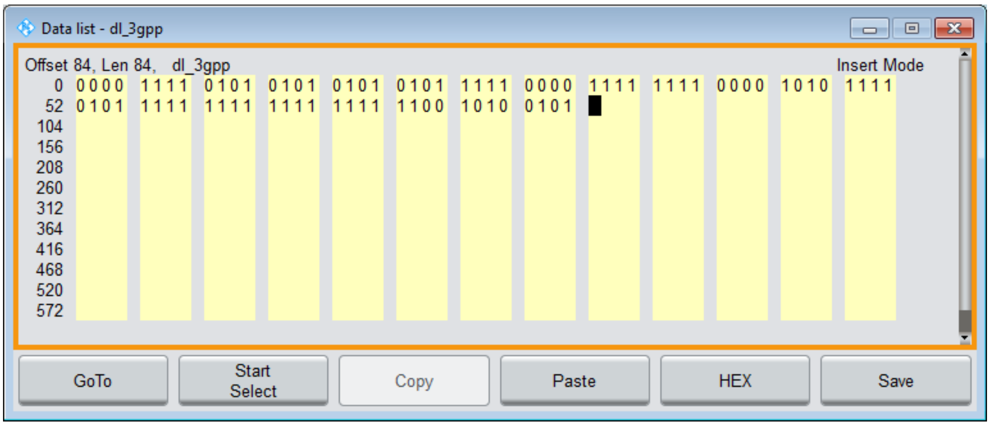
R&S WinIQSIM2 provides the following ways to create a data list file:

- Using the dedicated "Data List Editor" and create a file with extension `*.dm_iqd`, see "To create a data list manually" on page 88
- Using the tag-oriented format and create a data list file, see "To create a data list using tag file format" on page 276
- Using SCPI commands and create a file in binary format, see "To create a data list in binary format" on page 277

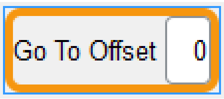
To access the "Data List Editor" dialog:

1. Select "Baseband > Custom Digital Mod > Data Source > Data List...".
2. Select "List Management...".
3. In the "List Management" dialog, select "Select Data List To Edit... > Select List / New List".
4. Navigate to the required directory.
5. Select an existing list, or enter a file name.
   For example: and .
   a) Select the directory `D:\user\`.

b) Enter "File Name > 'dl_3gpp'".



The "Data List Editor" is a list of binary values with a maximum length of 2^{31} bits. This value corresponds to a file size of approx. 268 Mbyte.
To increase readability, the bits are displayed in groups of four. The current cursor position, the length of the list and the list file name are displayed above the list. The offset starts with the value 0 which corresponds to the bit position on the left side of the first row, i.e. the beginning of the list. On the left edge of the editor, the last three offset positions are specified at the beginning of the row.
You can edit the list either in "Insert" or "Replace" mode, toggled with the "Insert" key.
SCPI command:
[:SOURce<hw>]:BB:DM:DLISt:SELect on page 347
[:SOURce<hw>]:BB:DM:DLISt:DATA on page 345
[:SOURce<hw>]:BB:DM:DLISt:DATA:APPend on page 346
The buttons below the binary list simplify the editing. The following table lists the provided functions.

GUI Element	Description
"GoTo"	Opens the entry window for the bit position. The cursor marks the bit at the selected position. 
"Start Select" / "Undo Select"	Defines the current cursor position as the start position for the range to be selected. To define the stop position, select "GoTo > Go To Offset" and define the offset. Selecting "Undo Select" deactivates the selected range.
"Copy", "Cut", "Paste"	Standard copy, cut and paste functions
"Hex"	Switch over to hexadecimal display. Each 4 bits are displayed as a hexadecimal value: To increase readability, the hexadecimal values in turn are displayed in pairs of two.
"Save"	Stores the changes made to the data list file selected for editing.

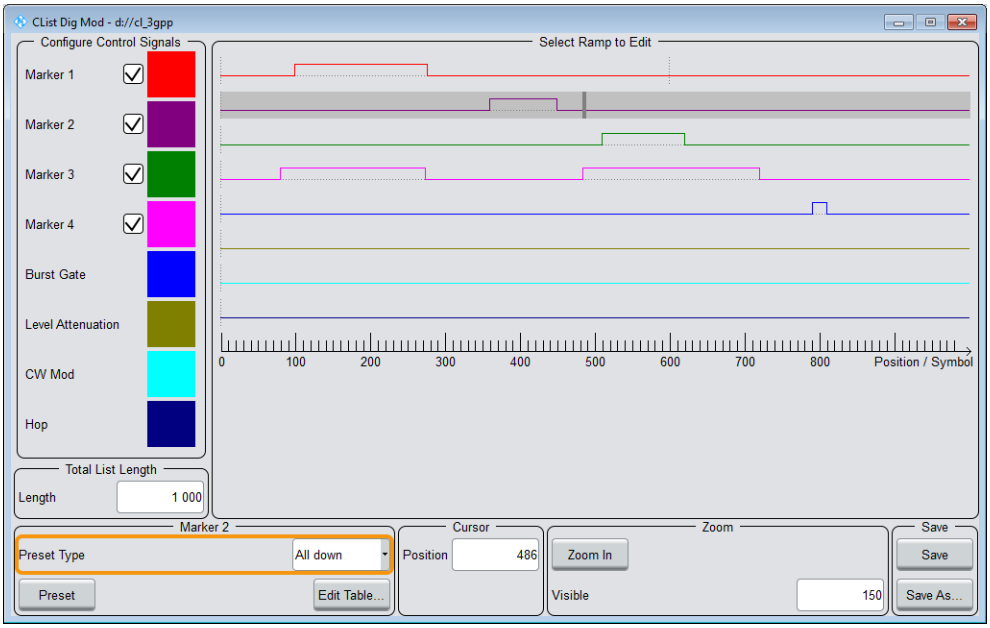
4.4.2.9 Control and Marker Lists Editor

R&S WinIQSIM2 provides the following ways to create a file containing control signals:

- Using the dedicated "Control Data Editor" and create a file in ASCII format and with extension `*.dm_iqc`.
  The "Control Data Editor" is described in this section. Refer to "To create a control list in ASCII format manually" on page 86 for step-by-step instructions.
- Using the tag-oriented format and create a control list file, see "To create a control list using tag file format" on page 275
- Using SCPI commands and create a file in binary format, see "To create a control list in binary format" on page 275

Access:

1. Select "Baseband > Custom Digital Mod > Data Source".
2. Select "List Management".
3. In the "List Management" dialog, select "Select Control List To Edit... > Select List / New List".
4. Navigate to the required directory.
5. Select an existing list, or enter a file name.
   For example:
   a) Select the directory `D:\user\`.
   b) Enter "File Name > 'cl_3gpp'".
6. Select "Edit Control List...".



The dedicated internal "Control Data Editor" is an intuitive graphical interface provided for defining and managing of:

- Marker signals
- Control signals, like the CW, Hop, Burst Gate and Lev_Att control signals (see also "Control Signals" on page 57)

A separate file with the file extension `*.dm_iqc` is created for each defined control signal kept on the hard disk. Control lists created with the editor are files in an ASCII file format.

In the "Control Data Editor" dialog, the available marker and control signals are displayed color-coded. The "Select Ramp to Edit" is a graphical display of the signal characteristics. To define the ramp for the individual markers or control signals, tap on the desired position or use the provided support functions "<Signal> Table" and "Cursor Position". To simplify the settings, use the predefined preset ramp characteristics in the "<Signal> Preset Type" section. The scaling of the x-axis is always adapted to the overall length of the control list to provide constant overview of all defined ramps. For detailed representation, zoom the displayed area around the current cursor position.

In the "Configure Control Signal" section, a status check box indicates whether the individual marker or control signal is assigned or enabled (see Chapter 4.4.3.1, "How to Create and Assign a Control List", on page 86).

SCPI command:

`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 345

`[:SOURce<hw>]:BB:DM:CLISt:CATalog?` on page 341

Configure Control Signal

Displays the color the marker/control signal has been assigned.

The status check box indicates whether the individual marker or control signal is assigned or enabled (see Chapter 4.4.3.1, "How to Create and Assign a Control List", on page 86).

Remote command:

n.a.

Select Ramp to Edit

Graphical representation for editing of the marker/control signals.

Refer to Chapter 4.4.3.1, "How to Create and Assign a Control List", on page 86 for an overview of the editing capabilities of the display.

Remote command:
{[TRACE] LIST [#]: Pos0:State0; Pos1:State1; ...PosN-1:StateN-1} on page 263
[:SOURce<hw>]:BB:DM:CLISt:DATA on page 343

Total List Length

Enters the length of the definition range of the control list in bits. The starting value is always bit 0. The entire definition range is displayed, i.e. the bit scale is adapted to the entry. If the definition range is decreased, the ramps outside the range are lost.

When used, the control list is always repeated over the length of the definition range if the length of the data list exceeds the length of the control list.

Tip: With long control lists, it is useful to zoom the displayed area around the current cursor position ("Zoom in").

Remote command:
{CONTROL LENGTH: ControlLength} on page 261

Preset Type

Triggered with "Preset", presets for the ramp characteristic of the selected control signal as defined with the "Preset Type".

"All Up, All Down "
Continuously high/low marker/control signal.

"Ramp Up, Ramp Down"
A ramp from low to high or high to low is configured in the center of the displayed signal area; ramp position can be subsequently shifted as required.

"Ramp Up/Down, Ramp Down/Up"
Created is a ramp sequence of low to high and high to low transitions, respectively high to low and low to high transitions. The ramps are symmetrically distributed around the center of the displayed signal area but be subsequently shifted as required.

Remote command:
n.a.

Cursor Position

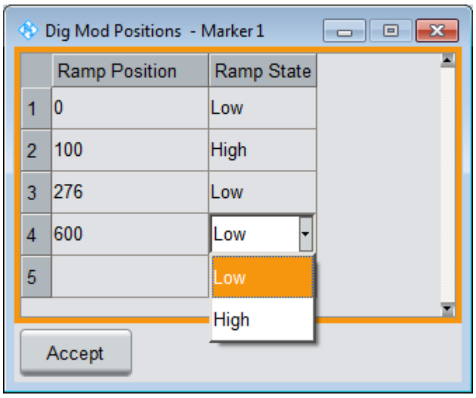
Displays/enters the cursor position in the graphical display

If the entered value exceeds the selected length of the definition range, the length is adjusted automatically.

Remote command:
n.a.

Positions Control Signal

Select "Edit Table" to access a dialog with representation of the ramps of the selected signal in table form.



The bit position is specified in the "Ramp Position" column, the high or low signal status in the "Ramp State" column. Use the last blank row to enter new ramps.

To apply the changes, press "Accept".

Remote command:
n.a.

Zoom/Visible
Zooms the displayed area of the control list. The designation of the button changes from "Zoom in" to "Zoom out".

With long control lists, it can be helpful to display only a part of the control list. In such cases, set the "Visible/Bits Visible" to determine the number of symbols/bits to be displayed and select "Zoom" to focus the displayed area around the current "Cursor Position".

Ramps outside the displayed area are not lost by zooming.

Remote command:
n.a.

Save/Save As
Stores the changes in the selected control list file or in a new file.

Remote command:
n.a.

4.4.3 How to Generate Custom Digitally Modulated Signals and Create Data and Control Lists

This section provides step-by-step instructions on configuring and using the provided settings. For details on individual functions and settings see Chapter 4.4.2, "Custom Digital Modulation Settings", on page 67.

To generate a digitally modulated signal

This example shows you how to generate a simple WCDMA-3GPP (QPSK 45° Offset) signal with the help of the "Custom Digital Modulation" functionality.

1. In the main application window, select "File > New" to start the application in a defined initial state.

2. Select "Baseband > Custom Digital Mod".

   The "Custom Digital Modulation" dialog opens.

3. Select "General > Set acc to standard > WCDMA-3GPP".

4. Select "General > State > On" to enable signal generation.

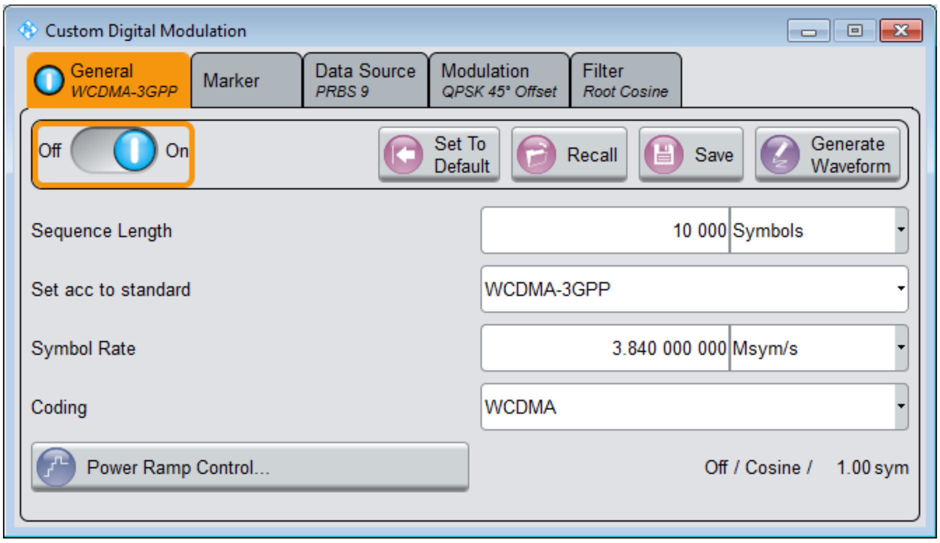


Figure 4-4: Selecting a WCDMA-3GPP standard signal

5. In the "Modulation" tab, consider the used "Modulation Type".

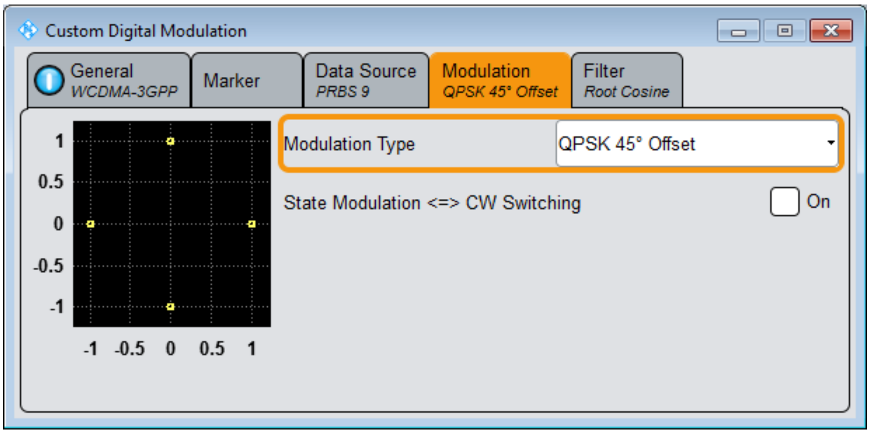


Figure 4-5: Display of the used modulation type

R&S WinIQSIM2 generates a WCDMA-3GPP signal, modulated with a QPSK 45° Offset modulation.

4.4.3.1 How to Create and Assign a Control List

R&S WinIQSIM2 provides the following ways to create a file containing control signals:

- Using the dedicated Control and Marker Lists Editor and create a file in ASCII format and with extension `*.dm_iqc`, see "To create a control list in ASCII format manually" on page 86
- Using the tag-oriented format and create a control list file, see "To create a control list using tag file format" on page 275
- Using SCPI commands and create a file in binary format, see "To create a control list in binary format" on page 275

To create a control list in ASCII format manually

Use the intuitive built-in Control and Marker Lists Editor dialog:

1. To access the "Control Data Editor", select "Baseband > Custom Digital Mod > Data Source > Select Control List" and select an existing file.
   Tip: To create a control list, select "List Management...> Select Control List To Edit... > New List".
2. Select "List Management > Edit Control List...".

   The "CList Dig Mod" dialog opens.
3. Adjust the control signals as required:
   a) Define the "Total List Length", e.g. "Length > 1000".
   b) In the "CList DigMod" dialog, select the graphic editor "Select Ramp to Edit".
   c) Select the color coded trace of the required signal.
   d) To insert a ramp, double-click the position where you want to the ramp.
   The number or ramps per marker is not limited.
   e) To remove a ramp, use the BACKSPACE key at the selected ramp.
   f) For faster marker and control signal definition, apply them with "Preset" via the predefined ramp functions ("Preset Type").
   g) If necessary, readjust the ramps with the help of the "Edit Table" function.
   h) Define the "Total List Length".
   i) To display only a part of the control list, select the "Visible/Bits Visible" to determine the number of symbols/bits to be displayed and select "Zoom" to focus the displayed area around the current "Cursor Position".
   Ramps outside the displayed area still remain during zooming.

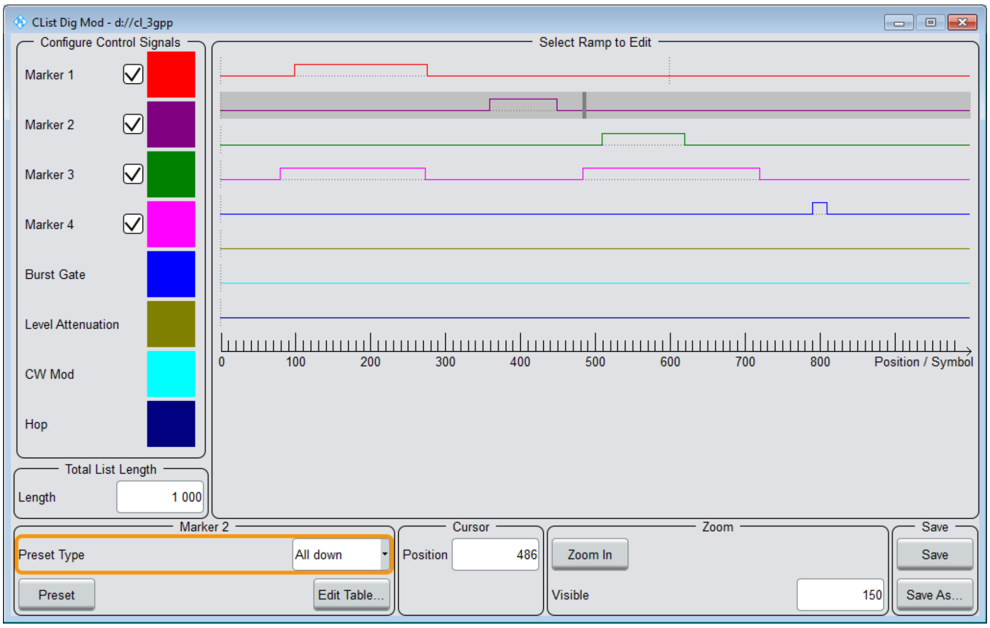


Figure 4-6: Example of control list settings

4. To store the settings in a control list file, select "Save / Save As ".

   The created file is an ASCII file with the extension `*.dm_iqc`

5. To include marker and control signals in the waveform signal, select the corresponding signals in the "Configure Control Signals" section.

To assign and activate control signals from a control list

Irrespective of the way they are created, generated control lists are not automatically included.

1. To use a marker/control signal from a control list, perform the following:
   a) Select "Baseband > Custom Digital Mod > Marker".
   b) Select "Marker x > CList"

2. To enable R&S WinIQSIM2 to include the Burst Gate and Level Attenuation control signals as defined in a control list:
   a) Select "Baseband > Custom Digital Mod > General".
   b) Select "Power Ramp Control > State > On".

   R&S WinIQSIM2 includes the control signals in the waveform signal.

4.4.3.2 How to Create and Assign a Data List

R&S WinIQSIM2 provides the following ways to create a data list file:

- Using the dedicated Data List Editor and create a file with extension `*.dm_iqd`, see "To create a data list manually" on page 88

- Using the tag-oriented format and create a data list file, see "To create a data list using tag file format" on page 276
- Using SCPI commands and create a file in binary format, see "To create a data list in binary format" on page 277

To create a data list manually

Use the intuitive build in Data List Editor dialog:

1. To access the "Data List Editor":
   a) Select "Baseband > Custom Digital Mod... > Data Source > Data List"
   b) Select "List Management...".
   c) In the "List Management" dialog, select "Select Data List To Edit... > New List".
   d) Navigate to the required directory, and enter a file name.
   For example, select the directory `D:\user\` and enter "File Name" = `dl`.

   The "Data List Editor" opens; the data list is empty.
2. Enter a sequence of 0 and 1, for example *01110101*.

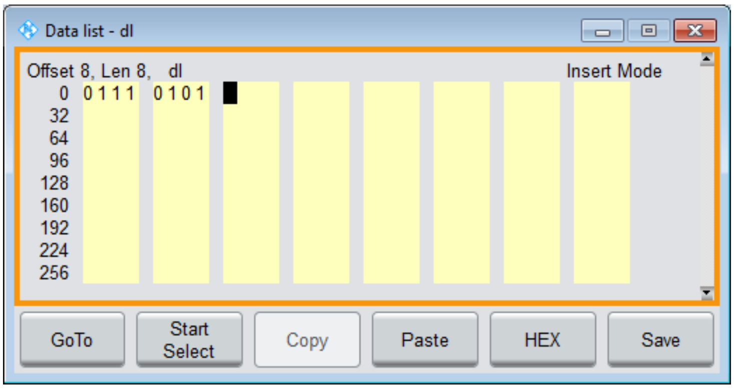


Figure 4-7: Example of data list

3. Select "Save" to store the used settings as a data list file.
   R&S WinIQSIM2 stores the `dl.dm_iqd` file in the `D:\user\` directory.

To assign and activate data lists

Irrespective of the way they are created, generated data lists are not automatically used.

1. To enable R&S WinIQSIM2 to use the data list as data source for the custom digital modulation:
   a) Select "Baseband > Custom Digital Mod... > Data Source > Data List"
   b) Select "Select Data List..."
   c) In the destination folder, select the file.
   d) Confirm with "Select".
2. To enable R&S WinIQSIM2 to use the data list as data source for any of the digital standards:

a) Select the "Data List Name" in the individual dialog.
   For example, for "Baseband > 3GPP FDD > Basestations > BS1 > Channel Table > P-CCPCH > Data = Data List"
b) Select "DList = None".
c) In the destination folder, select the file.
d) Confirm with "Select".
e) Select "State > On".

If you now activate the the corresponding standard, R&S WinIQSIM2 creates the waveform signal of the standard using the list as the data source.

4.4.4 References

Table 4-2: Custom digital modulation default values

Parameter	**Value**
State	Not affected by "Set to Default"
Data Source	PRBS 9
Standard	GSM
Symbol Rate	270.833 ksymb/s
Coding	GSM
Modulation Type	MSK
Filter	Gauss (FSK)
Filter Parameter BxT	0.3
Power Ramp Control	
Attenuation	15 dB
Ramp Time	1 sym
Ramp Function	Cosine
Fall Delay	0
Rise Delay	0
Source	Internal
State	Off
Source	Internal
Mode	Symbol

Table 4-3: Communication standards with their predefined settings

Standard	Modulation	Symbol Rate	Filter	Coding	Parameter for SCPI command
APCO Phase1 C4FM	4FSK, Deviation 1.8 KHz	4.8 ksym/s	APCO25, roll off = 0.2	APCO25(FSK)	`APCOPH1C4fm`
APCO Phase1 CQPSK	pi/4-DQPSK	4.8 ksym/s	Cosine, roll off = 0.2, BW = 4.8 KHz	APCO25(PSK)	`APCOPH1CQpsk`
APCO Phase2 H-CPM	4FSK, Deviation 3 KHz	6 ksym/s	APCO25 (H-CPM)	APCO25(FSK)	`APCOPH2HCpm`
APCO Phase2 H-DQPSK	pi/4-DQPSK	6 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(PSK)	`APCOPH2HDQpsk`
APCO Phase2 H-D8PSK Wide	pi/8-D8PSK	4 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(8PSK)	`APCOPH2HD8PSKW`
APCO Phase2 H-D8PSK Narrow	pi/8-D8PSK	4 ksym/s	Cosine, roll off = 1, BW = 5 KHz	APCO25(8PSK)	`APCOPH2HD8PSKN`
APCO Phase1 LSM	pi/4-DQPSK	4.8 ksym/s	APCO25 (LSM), Gauss Cut Off = 2.04 KHz, Low-pass Cut Off = 6.2 KHz	APCO25(8FSK)	`APCOPH1Lsm`
APCO Phase1 WCQPSK	pi/4-DQPSK	4.8 ksym/s	Cosine, roll off = 1, BW = 7.2 KHz	APCO25(8FSK)	`APCOPH1Wcqpsk`
Bluetooth	2FSK, Deviation 160.0 kHz	1.0 Msym/s	Gauss, B*T = 0.5	OFF	`BLUetooth`
DECT	2FSK, Deviation 288.0 kHz	1.152 Msym/s	Gauss, B*T = 0.5	OFF	`DECT`
ETC (ARIB STD T55)	ASK, ASK Depth 100%	1.024 Msym/s	Split Phase, B*T = 2.0	OFF	`ETC`
GSM	MSK	270.833333 ksym/s	Gauss, B*T = 0.3	GSM	`GSM`
GSM EDGE	8PSK EDGE (3pi/8 8PSK)	270.833333 ksym/s	Gauss linear	OFF	`GSMEdge`
NADC	pi/4 DQPSK	24.3 ksym/s	Root Cosine, alpha = 0.35	NADC	`NADC`
PDC	pi/4 DQPSK	21.0 ksym/s	Root Cosine, alpha = 0.50	PDC	`PDC`
PHS	pi/4 DQPSK	192.0 ksym/s	Root Cosine, alpha = 0.50	PHS	`PHS`
TETRA	pi/4 DQPSK	18.0 ksym/s	Root Cosine, alpha = 0.35	TETRA	`TETRa`
WCDMA 3GPP	QPSK 45° Offset	3.84 Msym/s	Root Cosine, alpha = 0.22	WCDMA 3GPP	`W3GPp`
TD-SCDMA	QPSK 45° Offset	1.28 Msym/s	Root Cosine, alpha = 0.22	OFF	`TCSCdma`
cdma2000 Forward	QPSK	1.2288 Msym/s	cdmaOne + Equalizer	cdma2000	`CFORward`
cdma2000 Reverse	Offset QPSK	1.2288 Msym/s	cdmaOne	cdma2000	`CREVerse`

Standard	Modulation	Symbol Rate	Filter	Coding	Parameter for SCPI command
Worldspace	QPSK	1.84 Msym/s	Root Cosine, alpha = 0.40	OFF	`WORLdspace`
TFTS	pi/4 DQPSK	22.1 ksym/s	Root Cosine, alpha = 0.40	TFTS/ TETRA	`TFTS`

Table 4-4: Permissible coding combinations for modulation symbols and modulation type

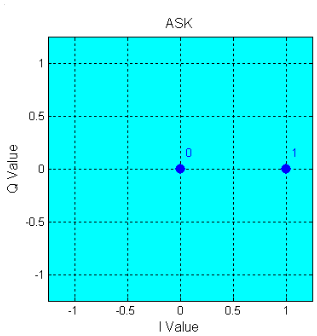
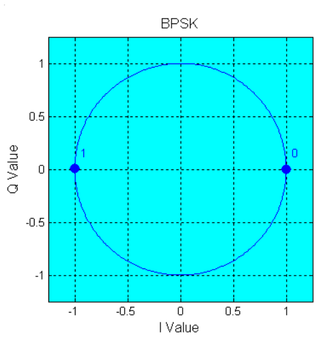
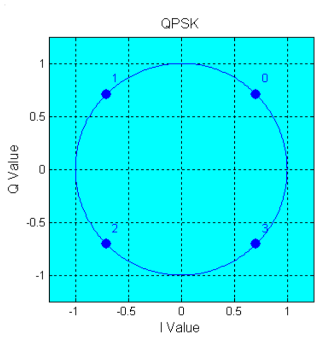
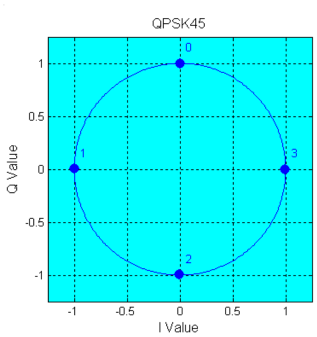
	OFF	Differ-ential	Phase Diff.	Diff. +Gray	Gray	GSM	NADC, PDC, PHS, TETRA, APCO25 (PSK), PWT	TFTS/ TETRA	INMARSAT, ICO, WCDMA, cdma2000	VDL	APCO25 (FSK)	APCO25 (8PSK)
ASK	X	X		X	X							
BPSK	X	X		X	X							
pi/2 DBPSK	X				X							
QPSK	X	X		X	X				X			
QPSK EDGE	X											
QPSK45° Off-set	X	X		X	X							
Pi/4 QPSK	X	X			X							
Pi/4 DQPSK	X				X		X	X				
8PSK	X	X		X	X					X		
AQPSK	X	X		X	X							
OQPSK	X	X		X	X							
8PSK_EDGE	X											
Pi/8 D8PSK	X				X							X
MSK	X	X		X	X	X						
2FSK	X	X		X	X	X						
4FSK	X	X		X	X						X	
8FSK	X	X		X	X							
16FSK	X	X		X	X							
16QAM	X	X	X	X	X							
16QAM EDGE	X											
32QAM	X	X	X	X	X							
32QAM EDGE	X											
64QAM	X	X	X	X	X							

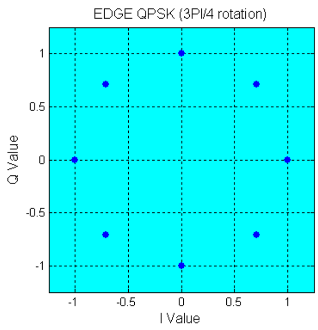
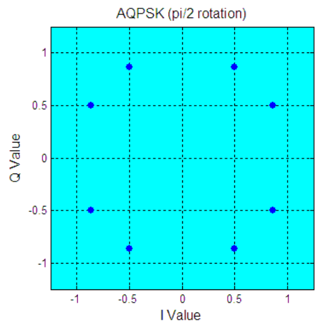
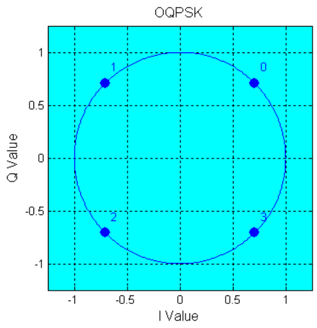
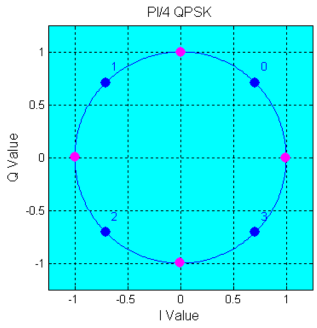
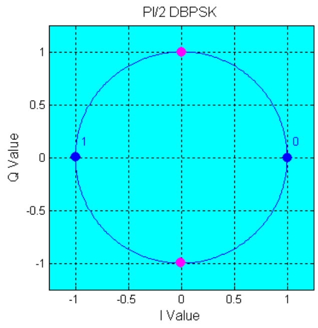
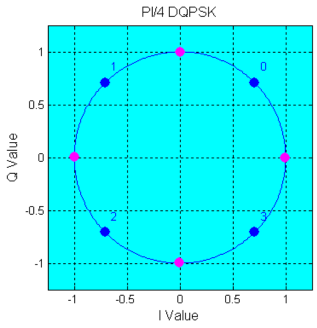
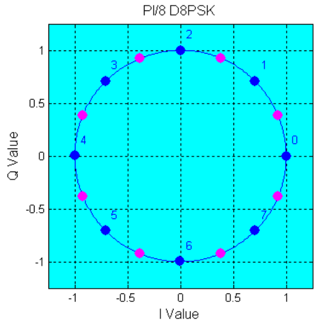
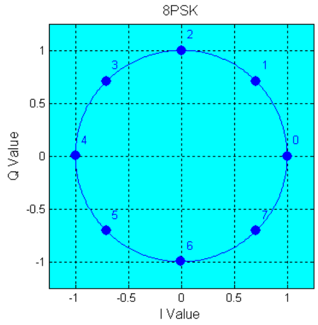
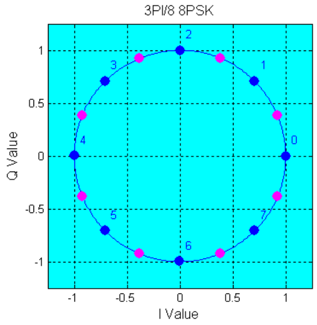
	OFF	Differ-ential	Phase Diff.	Diff. +Gray	Gray	GSM	NADC, PDC, PHS, TETRA, APCO25 (PSK), PWT	TFTS/ TETRA	INMARSAT, ICO, WCDMA, cdma2000	VDL	APCO25 (FSK)	APCO25 (8PSK)
128QAM	X	X	X	X	X							
256QAM	X	X	X	X	X							
1024QAM	X	X	X	X	X							

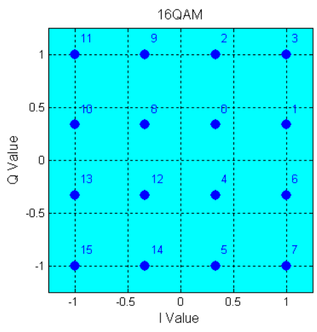
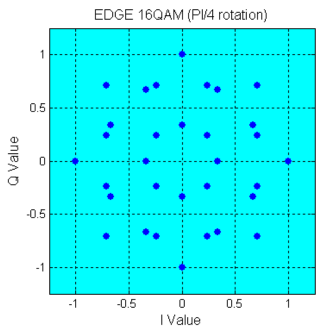
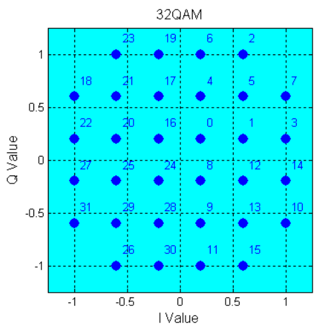
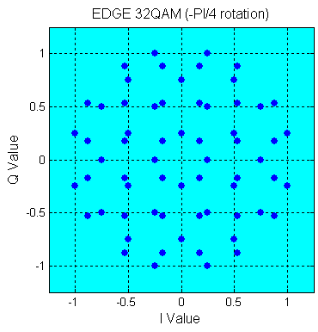
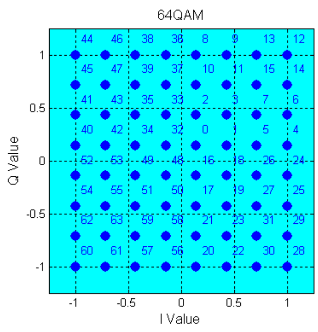
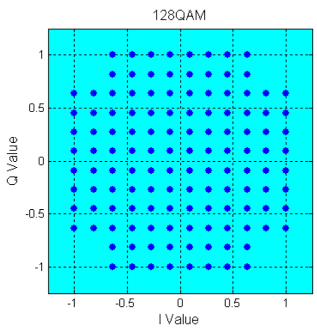
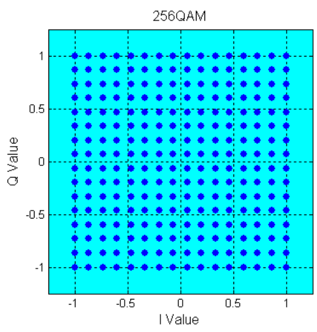
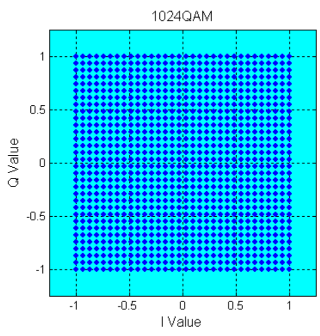
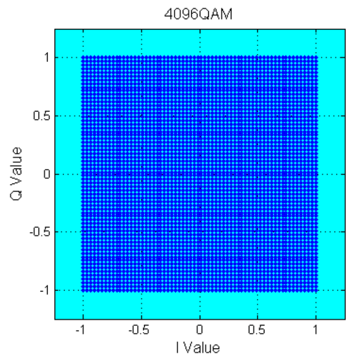
4.4.4.1 Predefined Modulation Types

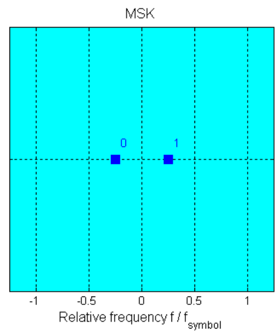
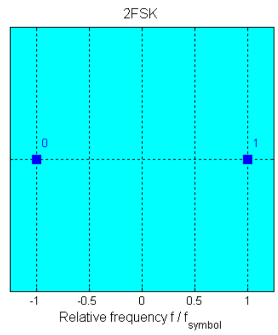
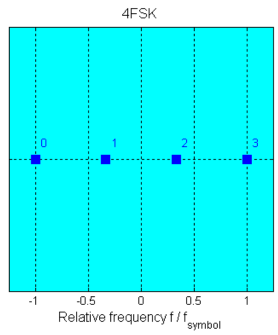
In Table 4-5, you can see the available modulation types and the associated parameters. The mapping for the selected modulation type is displayed in graphical form.

Table 4-5: Modulation type and associated mapping

ASK 1 bit per symbol		
		
PSK		
BPSK 1 bit per symbol	QPSK 2 bits per symbol	QPSK 45° Offset 2 bits per symbol, 45° rotation
		
QPSK EDGE 2 bits per symbol, 3pi/4 rotation	AQPSK 2 bits per symbol, pi/2 rotation	OQPSK 2 bits per symbol, Q offset

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>pi/4-QPSK
2 bits per symbol pi/4rotation</td>
<td>pi/2-DBPSK
1 bit per symbol
Differential coding, pi/2 rotation</td>
<td>pi/4-DQPSK
2 bits per symbol
Differential coding, pi/4 rotation</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>pi/8-D8PSK
3 bits per symbol
Differential coding, pi/8 rotation</td>
<td>8PSK
3 bits per symbol</td>
<td>8PSK EDGE (3pi/8 8PSK)
3 bits per symbol
Edge coding, 3pi/8 rotation</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td colspan="3">QAM</td>
</tr>
<tr>
<td>16QAM
4 bits per symbol</td>
<td>16QAM EDGE
4 bits per symbol, pi/4 rotation</td>
<td>32QAM
5 bits per symbol</td>
</tr>
</table>

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>32QAM EDGE
5 bits per symbol, -pi/4 rotation</td>
<td>64QAM
6 bits per symbol</td>
<td>128QAM
7 bits per symbol</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>256QAM
8 bits per symbol</td>
<td>1024QAM
10 bits per symbol</td>
<td>4096QAM
12 bits per symbol</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td colspan="3">FSK</td>
</tr>
</table>

MSK 1 bit per symbol	2FSK 1 bit per symbol FSK deviation	4FSK 2 bit per symbol FSK deviation
		

4.4.4.2 Common Coding Algorithms

In the notation used below, a_n denotes the n^{th} input symbol and b_n denotes the correspondingly coded output symbol. Individual bits in the symbols from the LSB (least significant bit) to the MSB (most significant bit) are denoted by a_{0n}, a_{1n}, etc. The same applies to the output symbols.

Common coding types are listed in Table 4-6.

Table 4-6: Common coding algorithms

Coding	**Coding algorithm**	**Applicable for K bit/symbol**
"None"	$b_n = a_n$	k = 1...8
"Differential"	$b_n = (a_n + b_{n-1})$ modulo 2^k	k = 1...7
"Differential + Gray"	Gray coding with additional differential coding	k = 1...7
"GSM"	d_n = NOT (d_n EXOR d_{n-1})	k = 1
"Phase differential coding"[1)]	$b_{1n} =$ [NOT (a_{1n} EXOR a_{0n}) AND (a_{1n} EXOR $b_{1(n-1)}$)] OR [(a_{1n} EXOR a_{0n}) AND (a_{1n} EXOR $b_{0(n-1)}$)] $b_{0n} =$ [NOT (a_{1n} EXOR a_{0n}) AND (a_{0n} EXOR $b_{0(n-1)}$)] OR [(a_{1n} EXOR $a_{0(n-1)}$) AND (a_{0n} EXOR $b_{1(n-1)}$)]	

[1)] "Phase differential coding" INMARSAT and PHASE DIFF correspond to system standards Inmarsat-M and DVB according to ETS 300 429. The INMARSAT coding can generally be used for modulation types with 2 bits/symbol, such as QPSK.

Examples

The following examples illustrate how some of the different common coding schemes in combination with a modulation method influence the signal.

Example: Differential coding for QPSK modulation with K = 2 bit/symbol

Decimal display; the value range for modulation symbols is $a_n = \{0, 1, 2, 3\}$

According to Table 4-6 and for $k = 2$, the recursive coding is defined as $b_n = (a_n + b_{n-1})$ *modulo 4*

Depending on the state of a preceding modulation symbol b_{n-1}, the coded modulation symbol b_n is obtained for example from modulation symbol $\mathbf{a_n = 2}$ as follows:

b_{-1}	0	1	2	3
b_n	2	3	0	1

By means of differential coding, the assignment of modulation symbols a_n (binary indication: MSB, LSB) to the phase differences shown in the following table is generated:

Table 4-7: Phase difference for QPSK

Modulation symbol a_n	**00**	**01**	**10**	**11**
Phase difference	0°	90°	180°	270°

Example: Differential coding for modulation type pi/4DQPSK

With differential coding switched on at the same time, the obtained constellation diagram for pi/4DQPSK is similar to that of 8PSK. Phase shifts are however assigned to the individual modulation symbols. The Table 4-8 shows the assignment of modulation symbols a_n (binary indication: MSB, LSB) to phase shifts of the I/Q vector in relation to the selected coding.

Table 4-8: Phase shifts for pi/4DQPSK

Modulation symbol a_n	**00**	**01**	**10**	**11**
"Coding = OFF"	+ 45°	+ 135°	- 135°	- 45°
"Coding = NADC, PDC, PHS, TETRA or APCO25 (PSK)"	+ 45°	+ 135°	- 45°	- 135°
"Coding = TFTS"	- 135°	+ 135°	- 45°	+ 45°

Example: Gray and differential coding for 8PSK modulation

When this coding scheme is used, the gray coding according to the gray code is performed befor the differential coding. The latter uses the recursive coding algorithm quoted above (see Table 4-6). The assignment of modulation symbols a_n (binary indication: MSB, LSB) to the phase differences shown in the following table is generated:

Table 4-9: Differential coding according to VDL[1)]

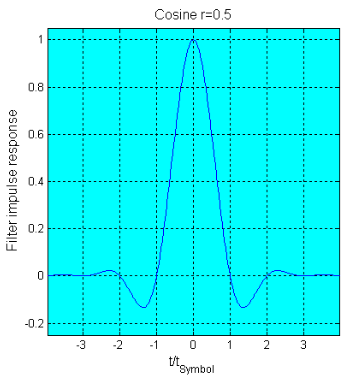
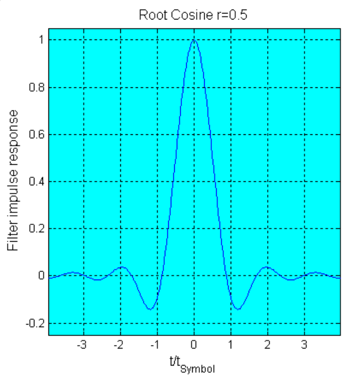
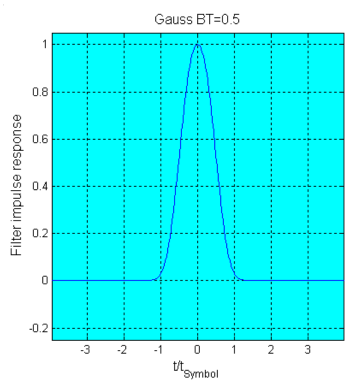
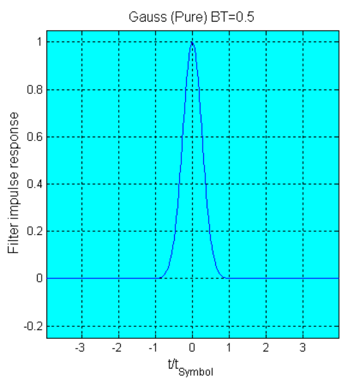
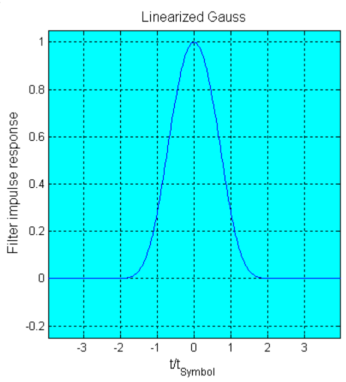
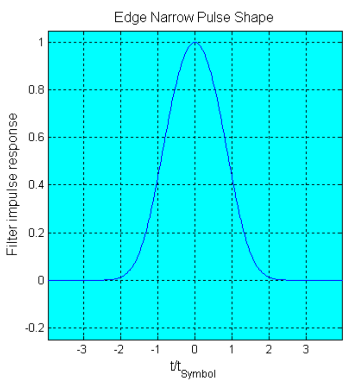
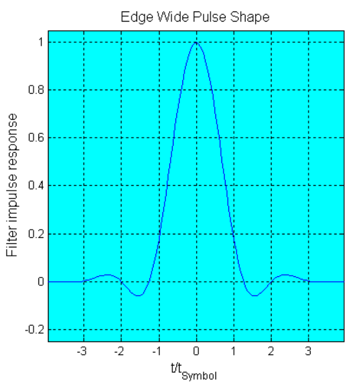
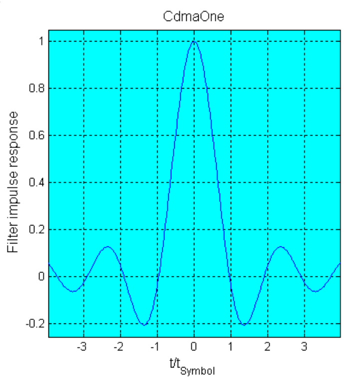
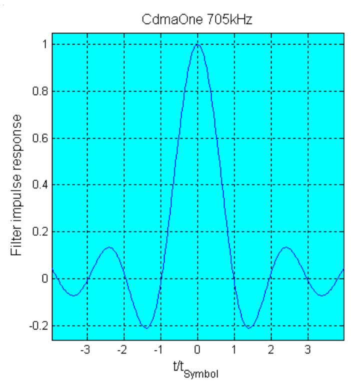
Modulation symbol d_n	000	001	010	011	100	101	110	111
Phase difference	0°	45°	135°	90°	270°	315°	225°	180°

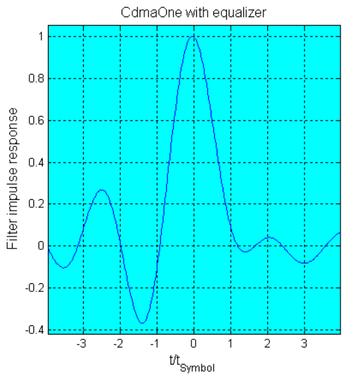
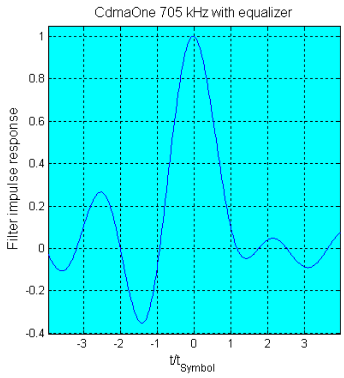
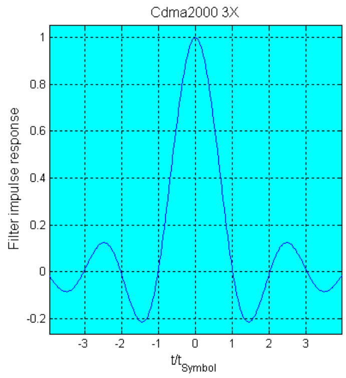
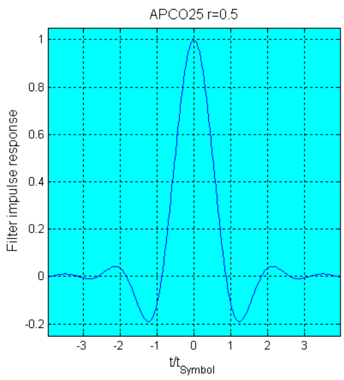
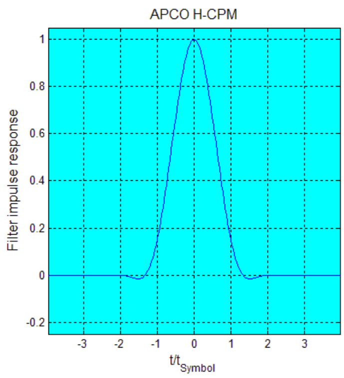
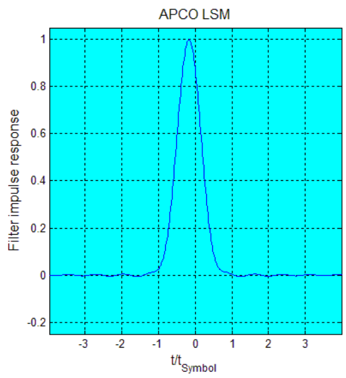
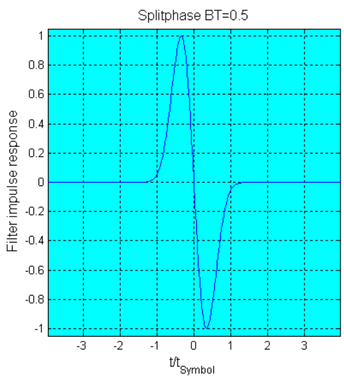
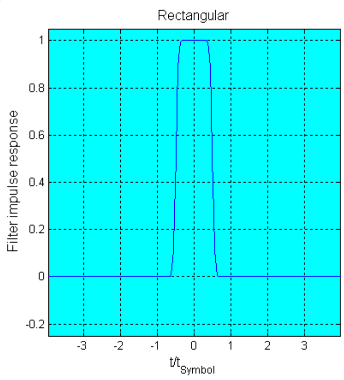
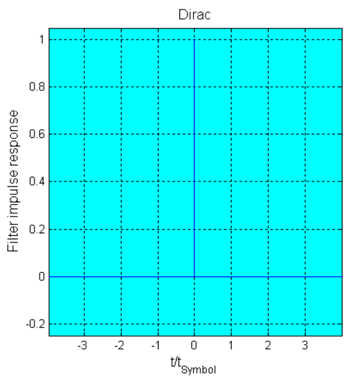
[1)] "Differential coding according to VDL" can be used for modulation types with 3 bits/symbol, e.g. 8PSK.

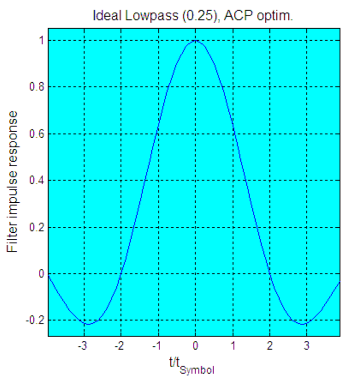
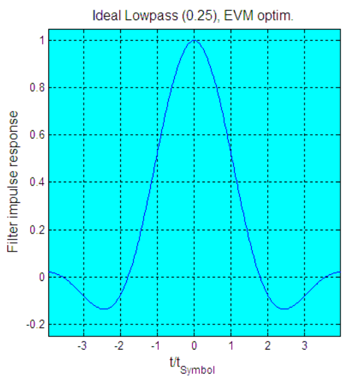
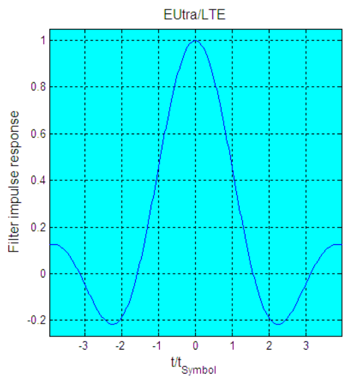
4.4.4.3 Predefined Baseband Filters

The Table 4-10 shows the filters that are available, together with their associated parameters. The filter characteristic is displayed in graphical form.

Table 4-10: Overview of the baseband filters

Cosine Roll Off Factor Bandwidth	Root Cosine Roll Off Factor	Gauss (FSK) B x T
		
Pure Gauss B x T	Gauss Linearized	Edge Narrow Pulse Shape
		
Edge Wide Pulse Shape	cdmaOne	cdmaOne 705 kHz
		

cdmaOne + Equalizer	cdmaOne 705 kHz + Equalizer	cdma2000 3X
		
APCO25 Roll Off Factor	APCO25 (H-CPM)	APCO25 (LSM) Gauss Cut Off Frequency Gauss Cut Off Frequency
		
Split Phase BxT	Rectangular	Dirac
		

Lowpass ACP opt.	Lowpass EVM opt.	EUTRA/LTE
		

4.5 Generating Multi Carrier Continuous Wave Signals

The multi carrier continuous wave (MCCW) function enables you to calculate waveforms with up to 8192 continuous wave carriers with R&S WinIQSIM2.

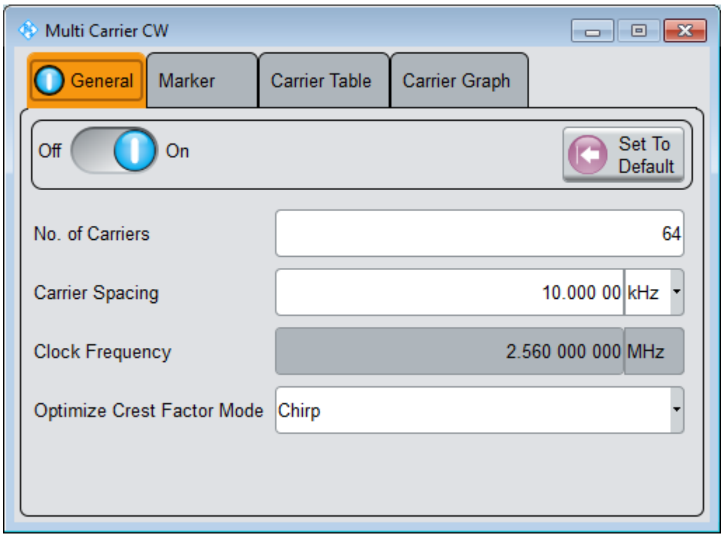
4.5.1 About the MCCW

R&S WinIQSIM2 can calculate and generate a multi carrier continuous wave signal build from up to 8192 unmodulated carriers. The carrier offset and the power level of the carriers are user-definable. The provided settings allow individual configuration of each carrier as well as the joint configuration of subsets of carriers. Automatic start phase setting is provided in order to minimize the crest factor.

The common application fields of the multi carrier CW signals are receiver tests with broadband test signals.

4.5.2 MCCW Settings

► To access the multi carrier CW settings, select "Baseband > Multi Carrier CW...".



The "Multi Carrier CW" dialog contains the parameters for configuring the carriers directly and provides access to the carrier table and the graphical representation.

The remote commands required to define these settings are described in Chapter 12.11, "SOURce:BB:MCCW Subsystem", on page 363.

4.5.2.1 General Carrier Settings

This tab contains the common multi carrier conrinuous wave parameters.

Settings:

State

Activates the multi carrier CW signal generation.

Note: Switching on MCCW turns off all the other digital standards and digital modulation modes.

Since multi carrier CW signals are computed in arbitrary waveform mode, R&S WinIQSIM2 adjusts the modifications of the "Carrier Table" and "Table Setup Assistant" settings only when confirmed with "Accept" (see Chapter 4.5.2.2, "Carrier Table", on page 102).

Remote command:
`[:SOURce<hw>]:BB:MCCW:STATe` on page 364

Set to Default
Sets all relevant parameters to default, see Table 4-11.

Remote command:
`[:SOURce<hw>]:BB:MCCW:PRESet` on page 365

Number of Carriers
Sets the number of carriers composing the multi carrier CW signal.

Note: Interdependence between total bandwidth, carrier spacing and number of carriers. The total bandwidth of the multi carrier CW signal is calculated as follows:
Total Bandwidth = ("Number of Carriers" - 1) * "Carrier Spacing"

The result must not exceed the system bandwidth of the instrument (see the data sheet of the corresponding instrument). If the total bandwidth is not respected when entering the number of carriers, the value selected as "Carrier Spacing" is reduced. Hence, the number of carriers entered defines the maximum carrier spacing.

By default the multi carrier table already lists 64 carriers that are preset to the settings "State > ON", "Power > 0 dB", "Phase > 0°".

If you use fewer carriers than listed in the table, it is recommended that you delete the superfluous entries. Vice versa, if you extend the number of carriers, R&S WinIQSIM2 adds the new entries at the end of the table.

Remote command:
`[:SOURce<hw>]:BB:MCCW:CARRier:COUNt` on page 365

Carrier Spacing
Sets the spacing between carriers for the multi carrier CW signal. The carriers are arranged symmetrically around the RF carrier.

Remote command:
`[:SOURce<hw>]:BB:MCCW:CARRier:SPACing` on page 368

Clock Frequency
Displays the clock rate at which the multi carrier CW signal is output by the arbitrary waveform generator. The output clock rate depends on the number of carriers and the selected carrier offset.

The value indicates the resolution during the marker generation.

Remote command:
`[:SOURce<hw>]:BB:MCCW:CLOCk?` on page 370

Optimize Crest Factor Mode
Selects the mode for automatically minimizing the crest factor. The carrier start phases are automatically configured according to the selected mode.

Methods of reducing the crest factor differ with regard to both the achievable optimization and the time required for computing. For more information, refer to Chapter 4.3.1.6, "Methods for Optimizing the Crest Factor", on page 64.

"Off" There are no automatic settings for minimizing the crest factor. The "Phase" setting as defined in the carrier table is in use.

"Chirp" Rapid crest factor optimization regardless of the number of carriers. A minimal crest factor of < 3 dB is only obtained for multi carrier signals in which all carriers are switched on and the power of the carriers is identical. In a different configuration, the achievable crest factor is worse.

"Target Crest" Optimization of the crest factor to a desired value for all carrier configurations. The optimization time depends on the number of carriers and the desired crest factor. The computing time increases only when the number of carriers exceeds 256 and the crest factor is below 4 dB. The desired value can be entered in "Desired Crest Factor".

Note: You can cancel the optimization at any time. R&S WinIQSIM2 uses the value reached at this time, as indicated.

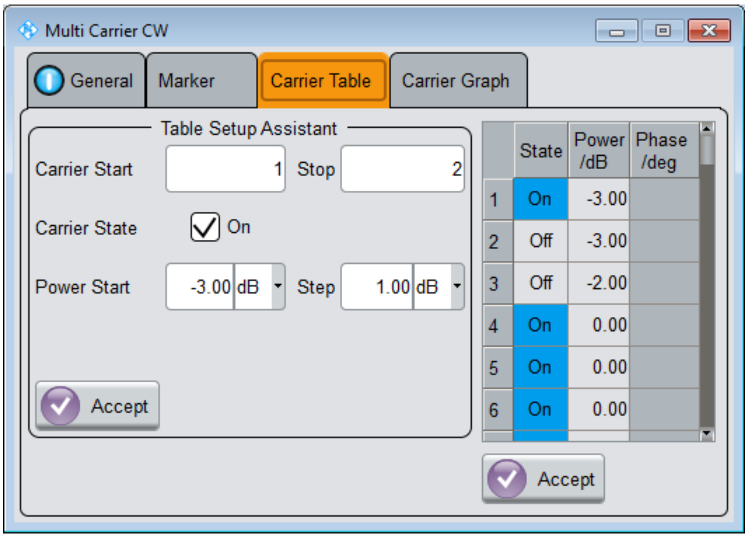
Remote command:
[:SOURce<hw>]:BB:MCCW:CFACtor:MODE on page 370

Desired Crest Factor
Defines the desired "Target Crest" factor for optimization.

Remote command:
[:SOURce<hw>]:BB:MCCW:CFACtor on page 369

4.5.2.2 Carrier Table

This tab contains the settings required for configuring the selectable range of carriers.



Settings:

Table Setup Assistant

Enables joint configuration of a selectable range of carriers.

Carrier Start/Stop ← Table Setup Assistant

Defines the start/stop index of the carrier range to which the following settings are intended to apply.

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STARt` on page 373
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STOP` on page 373

Carrier State ← Table Setup Assistant

Switches the carriers in the carrier range on/off.

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STATe` on page 373

Power Start ← Table Setup Assistant

Sets the power of the starting carrier.

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer[:STARt]` on page 373

Power Step ← Table Setup Assistant

Sets the step width that is used to increment the power from carrier to carrier.

The individual power of each carrier ("Carrier Table > Carrier# >Power") is calculated as "Power Start" + n*"Power Step".

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer:STEP` on page 372

Phase Start ← Table Setup Assistant

Sets the phase of the starting carrier. This setting is only available for "Optimize Crest Factor Mode > Off".

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe[:STARt]` on page 372

Phase Step ← Table Setup Assistant

Sets the step width that is used to increment the phase from carrier to carrier.

The individual phase of each carrier ("Carrier Table > Carrier# >Phase") is calculated as "Phase Start" + n*"Phase Step".

Remote command:

`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe:STEP` on page 371

Accept ← Table Setup Assistant
Adopts the carrier range setting into the "Carrier Table".

Remote command:
`[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:EXECute` on page 371

Carrier Table
A table with additional settings for the configuration of the individual carriers.

R&S WinIQSIM2 computes the multi carrier CW signal only after you have confirmed the settings with "Accept". As long as you have not confirmed changes, the table background is yellow.

Note: The phase/deg settings are only effective if you have deactivated optimization of the crest factor ("Optimize Crest Factor > Off").

Tip: Use the Carrier Graph function to display the current carrier configuration.

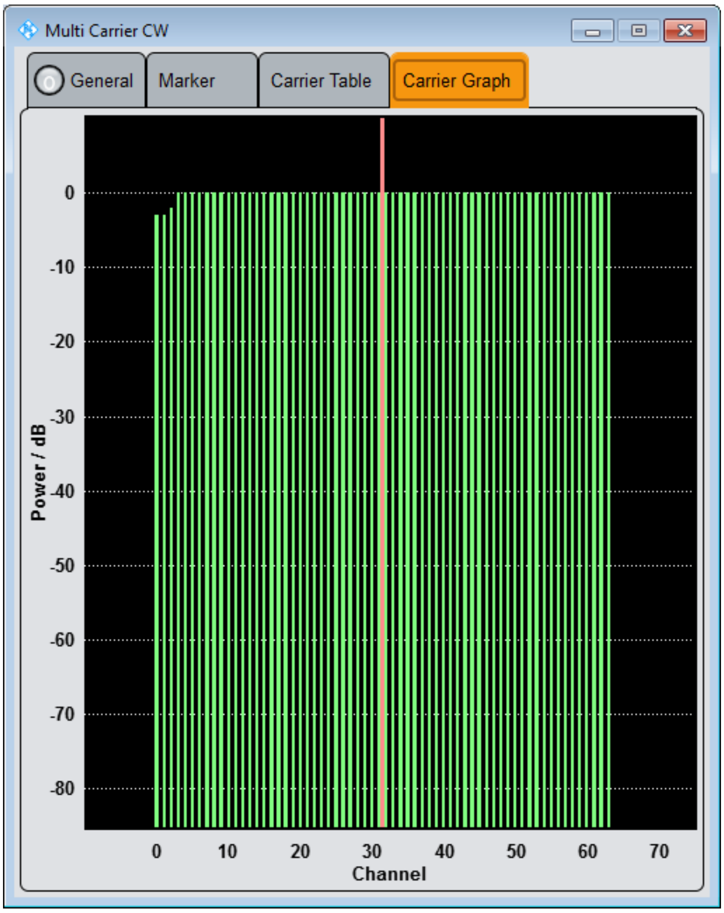
"No."	Indicates the carrier index.
"State"	Switches a carrier on/off
"Power"	Sets the power of a carrier. If you use the Table Setup Assistant settings to fill the individual carrier powers, the power of a carrier is calculated as "Power Start" + n*"Power Step".
"Phase"	Sets the starting phase of a carrier. If you use the Table Setup Assistant settings to fill the individual carrier powers, the power of a carrier is calculated as "Phase Start" + n*"Phase Step".
"Accept"	Applies the settings in the carrier table for signal generation.

Remote command:
`[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe` on page 367
`[:SOURce<hw>]:BB:MCCW:CARRier:STATe` on page 369
`[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer` on page 366
`[:SOURce<hw>]:BB:MCCW:CARRier:POWer` on page 368
`[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe` on page 365
`[:SOURce<hw>]:BB:MCCW:CARRier:PHASe` on page 367

4.5.2.3 Carrier Graph

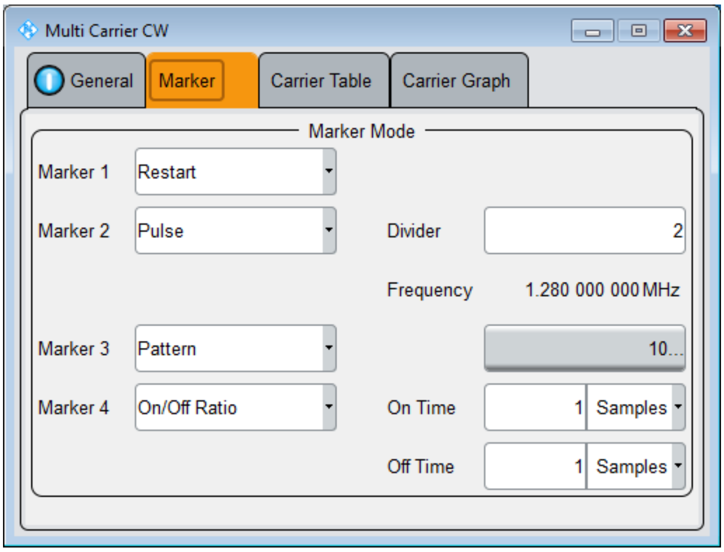
The "Carrier Graph" tab represents the current carrier configuration graphically.

The X-Axis covers the carriers, and the colored bars represent the active carriers. The Y-axis represents the power, that is, the height shows the set power of each individual carrier.



4.5.2.4 Marker Settings

This tab provides access to the settings necessary to select and configure the marker mode settings.



Settings:

Marker Mode

Marker configuration for up to four marker channels. The settings are used to select the marker mode defining the shape and periodicity of the markers. The contents of the dialog change with the selected marker mode; the settings are self-explanatory.

Use the settings to define the shape and periodicity of the markers. See Chapter 4.3.1.2, "Marker Signals", on page 59 for description of the regular marker signals.

Remote command:
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:MODE` on page 374
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:DIVider` on page 376
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:FREQuency?` on page 376
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PATTern` on page 375
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:ONTime` on page 375
`[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:OFFTime` on page 375

4.5.3 How to Use the Multi Carrier Continuous Wave Function

This section provides examples on how to configure a multi-carrier continuous wave signal for some general test cases:

- To generate a test signal for testing the frequency response of a DUT
- To generate a single sideband test signal for testing image rejection

Information on test setups or instrument configurations are not considered here.

To generate a test signal for testing the frequency response of a DUT

To create a test signal with a bandwidth of 80 MHz, e.g. to test a power amplifier or filter, configure the signal as follows:

1. Select "Baseband > Multi Carrier Continuous Wave".
2. In the "General" tab, set "Number of Carriers > 81".
3. Set "Carrier Spacing > 1 MHz".
4. Adjust further settings, e.g. the marker signals.
5. Enable signal generation with "Multi Carrier Continuous Wave > State > ON".

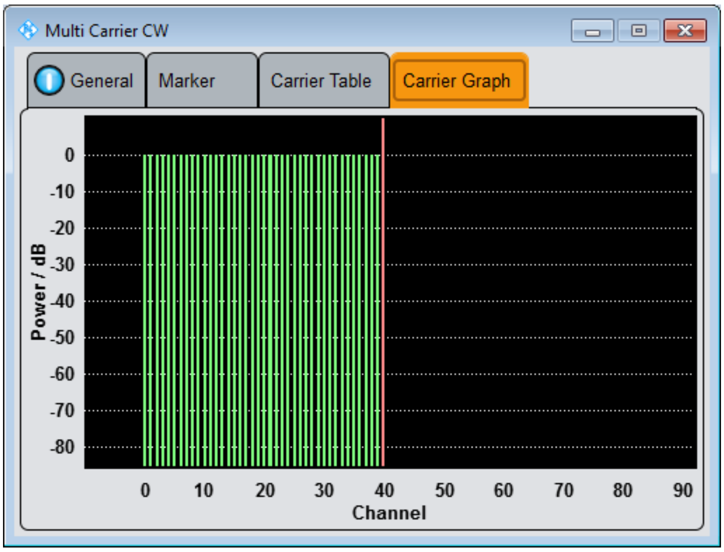
   R&S WinIQSIM2 computes the multi carrier CW signal and provides the waveform for storing or transmitting to a connected instrument.

To generate a single sideband test signal for testing image rejection

If a single sideband test signal for testing the image rejection capabilities of DUT is required, proceed as follows:

1. Select "Baseband > Multi Carrier Continuous Wave".

2. In the "General" tab, set "Number of Carriers > 81".
3. Set "Carrier Spacing > 1 MHz".
4. In the "Carrier Table" tab, perform the following settings:
   a) Set "Carrier Start > 40"
   b) Set "Carrier Stop > 81"
   c) Select "Carrier State > Off".
   d) Confirm with "Accept".
5. Select the "Carrier Graph" tab, to visualize the configured sideband signal.



6. Adjust further settings, e.g. marker signals.
7. Enable signal generation with "Multi Carrier Continuous Wave > State > ON".

   R&S WinIQSIM2 computes the multi carrier CW signal and provides the waveform for storing or transmitting to a connected instrument.

4.5.4 References

Table 4-11: MCCW default values

Parameter	**Value**
State	Not affected by "Set to Default"
Carrier Setup	
Number of Carriers	64
Carrier Spacing	10 kHz
Optimize Crest Factor	Chirp
Desired Crest Factor	3 dB
Marker	

Parameter	Value
Channel 1...4	Restart
Multi Channel Setup	
Start Carrier	0
Stop Carrier	0
State	ON
Power	0 dB
Power Step	0 dB
Initial Phase	0°
Phase Step	0°
Channel Setup	
State	ON
Phase	0°
Power	0 dB

4.6 Generating Multi Carrier Signals

To simulate complex multi carrier scenarios with different baseband signals, R&S WinIQSIM2 provides the possibility to generate multi carrier waveforms. These waveforms can consist of up to 512 carriers, each modulated by the same or by different user-selectable baseband signals.

The multi carrier waveform is a practical solution for the generation of a complex broadband signal. You can compose the signal on the basis of various communication standards, e.g. CDMA2000 or 3GPP FDD. In addition, R&S WinIQSIM2 supports multi carrier signals composed of several signals from the same communication standard as it is the case with LTE Advanced.

4.6.1 Required Options

R&S WinIQSIM2 offers all included digital standards for the creation of multisegment waveform files. However, the used instrument must be fitted with the associated R&S WinIQSIM2 options for the generation of the individual carrier signals.

4.6.2 About the Multi Carrier Waveforms

This chapter provides background information on the ARB functionality for generating multi carrier signals and the impact of the provided settings. Refer to Chapter 4.6.4, "How to Use the Multi Carrier Function", on page 123 for information on how to use the provided settings to configure a multi carrier signal.

Multi carrier waveforms are a convenient way to configure broadband test signals required for transmitter or receiver tests. Even complex multi carrier scenarios composed of signals from different digital standards can be created and used for these tests.

Multi carrier files can be processed by an ARB generator, therefore the composed waveform file must be created before loading and playing by the ARB. Using R&S WinIQSIM2, you can configure and create multi carrier waveforms and transmit the computed waveform directly to a connected instrument or store it in a file. The filename is user-definable; as with the single carrier waveforms, and the file extension is `*.wv`.

General principle for composing a multi carrier signal

The following is a list of the general steps used for composing a multi carrier signal:

- In the default mode, the up to 512 carriers are equidistantly distributed, centered around the carrier index, which represents the RF frequency or baseband DC line. The carrier spacing is adjustable within the total available bandwidth. The total RF bandwidth of the composed multi carrier signal must not exceed the bandwidth of the instruments' ARB (see data sheet).
- Another possibility is to define the center frequency of each of the carrier individually, see also Chapter 4.6.2.1, "Defining the Carrier Frequency", on page 109.
- Each carrier can be separately defined in terms of power, phase and modulated input signal.
  To define the leveling of the composed multi carrier signal, use the parameter "Power Reference" on page 115
- Optionally, crest factor optimization can be applied (see Chapter 4.6.2.2, "Optimizing the Crest Factor", on page 110).
- After all multi carrier processing steps are completed, R&S WinIQSIM2 calculates the resulting peak and RMS power over the total signal and enters the values in the waveform file.

4.6.2.1 Defining the Carrier Frequency

There are two ways to define the carrier frequency of the individual carriers in the multi carrier signal.

- Enabling the **arbitrary carrier frequency** distribution and specifying the carrier frequency of each carrier individually.
- Using the built-in **equidistant carrier spacing** distribution function, i.e. enabling the carriers to be equally spaced and centered toward the RF frequency. The carrier frequencies are automatically calculated depending on the selected number of carriers and the carrier spacing.
  The maximum carrier spacing is calculated as a function of the available total bandwidth and the selected number of carriers as follows:
  Max Carrier Spacing = Total bandwidth /(Number of Carriers - 1)
  To avoid wrap-around problems, the effective applied carrier spacing can be slightly modified.

The value of the carrier spacing is rounded in that way that the carrier closest to the baseband DC line shows no phase jump assuming that the carrier is unmodulated.

– For odd number of carriers:
*RoundedCarrierSpacing = 1/OutputSignalDuration*round(CarrierSpacing*OutputSignalDuration)*
– For even number of carriers:
*RoundedCarrierSpacing=2/OutputSignalDuration*round(0.5*CarrierSpacing*OutputSignalDuration)*

R&S WinIQSIM2 provides the parameter Mode for this purpose.

4.6.2.2 Optimizing the Crest Factor

An introduction to the topic is provided in Chapter 4.3.1.6, "Methods for Optimizing the Crest Factor", on page 64. This section focuses on the settings provided for and related to the multi carrier signals.

R&S WinIQSIM2 provides a crest factor reduction in form of an automatic optimization upon selected values for the following parameters:

- **Crest factor optimization mode**: determines whether the phase settings are selectable or internally calculated to fulfill a requirement for the crest factor value.
- **Clipping:** reduces the peak power of the resulting multi carrier signal according to the input parameter "Target Crest Factor"
The resulting clipped peak power is defined by sum of the RMS level of the unclipped multi carrier signal and the input parameter "Target Crest Factor".
Since clipping reduces also the RMS level, the resulting crest factor of the clipped signal is slightly above the "Target Crest Factor".
- **Target Crest Factor:** determines the desired crest factor. A value above the crest factor of the unclipped multi carrier signal has no effect.
- **Filter Cut Off Frequency:** determines the filter parameter of the final lowpass filter.
When the cut off frequency is set as half of the output sample rate, a final lowpass filter improves the spectrum of the clipped multi carrier signal. However, it can also increase the resulting crest factor.

4.6.3 Multi Carrier Settings

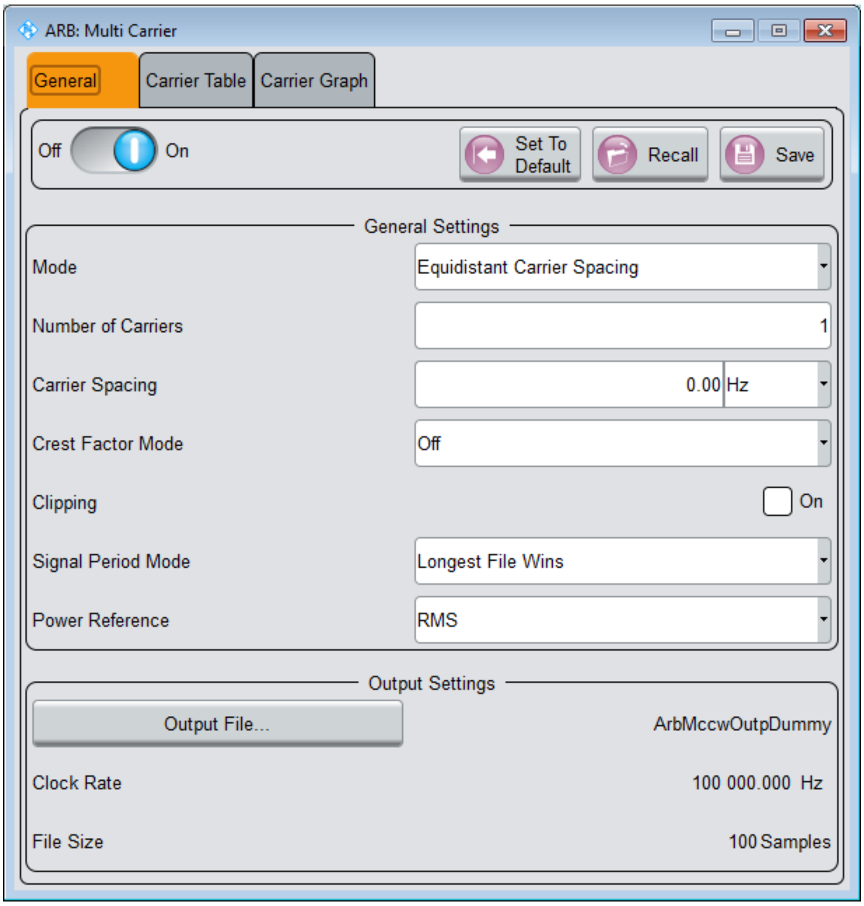
► To access the multi carrier settings, select "Baseband > Multi Carrier".

The "ARB: Multi Carrier" dialog enables direct configuration of the carrier distribution around the center frequency and adjusting some parameters for improving the signal characteristics.
The dialog is divided into several tabs. In each case, the current setting is displayed in the tab name. In the tab "Carrier Table", you can select the waveform file to be processed. Use the "Carrier Graph" to visualize the configured signal.

The remote commands required to define these settings are described in Chapter 12.7, "SOURce:BB:ARB:MCAR Subsystem", on page 301.

4.6.3.1 General Settings

This tab provides access to the default settings, selecting the output waveform file and enabling the generation as well as the configuration of the carrier distribution.



Settings:

State Multi Carrier
Activates multi carrier generation.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:STATe` on page 314

Set to Default
Sets all relevant parameters to default, see Table 4-12.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:PRESet` on page 312

Save/Recall Frame
Accesses the standard "Save/Recall" dialog, i.e. the standard function for storing and recalling the complete dialog-related settings. Refer to Chapter 9.2, "Storing and Recalling Application Settings", on page 208 for a detailed description.

The multi carrier settings are stored as files with the predefined file extension `*.arb_multcarr`. The file name and the directory they are stored in are user-definable.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:CATalog?` on page 313
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:LOAD` on page 313
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:STORe:FAST` on page 314
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:STORe` on page 314

Mode
Selects the way the carriers are distributed within the available bandwidth.

"Equidistant Carrier Spacing"
Sets an equidistant carrier spacing distribution, i.e. the carriers are equally spaced and centered around the RF frequency. The carrier frequencies are automatically calculated depending on the selected number of carriers and the carrier spacing.

"Arbitrary Carrier Frequency"
Enables the specification of the carrier frequency of each carrier individually.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:MODE` on page 302

Number of Carriers
Sets the number of carriers for the multi carrier waveform. By default the multi carrier table lists 1 carrier. A maximum of 512 carriers can be configured and activated.

Tip: You can find and modify the number of carriers in both, the "General" tab as well as in the dialogs "Carrier Table" and "Carrier Table Assistant".

When the number of carriers is increased, the multi carrier table is extended by adding further lines at the end of the table. If these carriers already have been defined before, the settings are preset according to the former settings. Otherwise the parameters are preset with the default values.

See also Chapter 4.6.2.1, "Defining the Carrier Frequency", on page 109.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:COUNt` on page 302

Carrier Spacing
Sets the frequency spacing between adjacent carriers of the multi carrier waveform.

The carriers are arranged symmetrically around the RF carrier and the maximum carrier spacing is limited to:

"Carrier Spacing" = Total baseband bandwidth /("Number of Carriers" - 1).

The total baseband bandwidth depends on the used ARB generator which replays the waveform. The total bandwidth is specified in the data sheet of the corresponding instrument.

For more information, see Chapter 4.6.2.1, "Defining the Carrier Frequency", on page 109.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:SPACing` on page 302

Crest Factor Mode
Selects the mode for optimizing the crest factor by calculating the carrier phases. For more information, refer to Chapter 4.6.2.2, "Optimizing the Crest Factor", on page 110.

The following modes are available:

"Off"	There is no automatic setting for minimizing or maximizing the crest factor. The "Phase" setting as defined in the carrier table is in use.
"Minimize"	The crest factor is minimized by internally calculating optimized carrier phases. The Phase setting displayed in the carrier table is invalid.
"Maximize"	The crest factor is maximized by internally calculating optimized carrier phases. The Phase setting displayed in the carrier table is invalid.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CFACtor:MODE` on page 305

Clipping
Switches the baseband clipping on and off.

Clipping reduces the peak power of the resulting multi carrier signal according to the input parameter "Target Crest Factor". For more information, refer to Chapter 4.6.2.2, "Optimizing the Crest Factor", on page 110.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping[:STATe]` on page 307

Target Crest Factor
Sets the value of the desired crest factor, if baseband clipping is enabled.

A "Target Crest Factor" above the crest factor of the unclipped multi carrier signal has no effect. For more information, refer to Chapter 4.6.2.2, "Optimizing the Crest Factor", on page 110.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping:CFACtor` on page 306

Filter Cut Off Frequency
Sets the cut off frequency of the final lowpass filter, if baseband clipping is enabled.

When the cut off frequency is set as half of the output sample rate, a final lowpass filter improves the spectrum of the clipped multi carrier signal. However, it also increases the resulting crest factor. See also Chapter 4.6.2.2, "Optimizing the Crest Factor", on page 110.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping:CUToff` on page 306

Signal Period Mode
Defines the way the resulting signal period of the multi carrier waveform is calculated. The resulting period is always calculated for all carriers in the carrier table irrespective of their state (ON/OFF). Use the "Carrier Table > Info" function to obtain information on the sample rate and file length data of each carrier.

Note: Wrap-around and timing problems occur when I/Q signals of different length are used. Thus, demodulation of a carrier can be difficult or even impossible. It is therefore recommended, that you consider the timing when creating the input I/Q files or to adjust the signal duration to the carrier which is subsequently demodulated. In this case, the other carriers are for interfering the signal only. These problems do not arise with signals of the same standard (e.g. 3GPP).

The following modes are available:

"Longest File Wins"
The resulting signal period is defined by the longest I/Q file in the carrier table. Shorter I/Q files are periodically repeated.

"Shortest File Wins"
The resulting signal period is defined by the shortest I/Q file in the carrier table. Only the first part of longer I/Q files is used.

"User"
Enables you to define a user-specific Signal Period.
Shorter I/Q files are repeated periodically, and only the first part of longer I/Q files is used.

"Least Common Multiple"
The output file duration is the least common multiple of all input file durations.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:TIME:MODE` on page 315

Signal Period
Sets the signal period in "Signal Duration Mode > User". Shorter I/Q files are repeated periodically, and only the first part of longer I/Q files is used.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:TIME on page 314

Power Reference

Defines the way the individual carriers in a composed multi carrier signal are leveled. The difference between the provided modes is especially important if signals with different crest factors are composed together into a multi carrier signal.

"RMS" The individual carriers are leveled based on their RMS power and the configured "Carrier Gain".

Example:

A multi carrier signal is composed from two waveform files.

First carrier "Gain" = 0 dB

Second carrier "Gain" = -3 dB

In the resulting multi carrier signal, the *RMS power* of the second carrier signal will be 3 dB lower than the RMS power of the first carrier signal.

"Peak" The individual carriers are leveled based on their peak power and the configured "Carrier Gain".

Example:

A multi carrier signal is composed from two waveform files.

First carrier "Gain" = 0 dB

Second carrier "Gain" = -3 dB

In the resulting multi carrier signal, the *peak power* of the second carrier signal will be 3 dB lower than the peak power of the first carrier signal.

Example: Multi carrier signal composed of waveforms with different crest factor

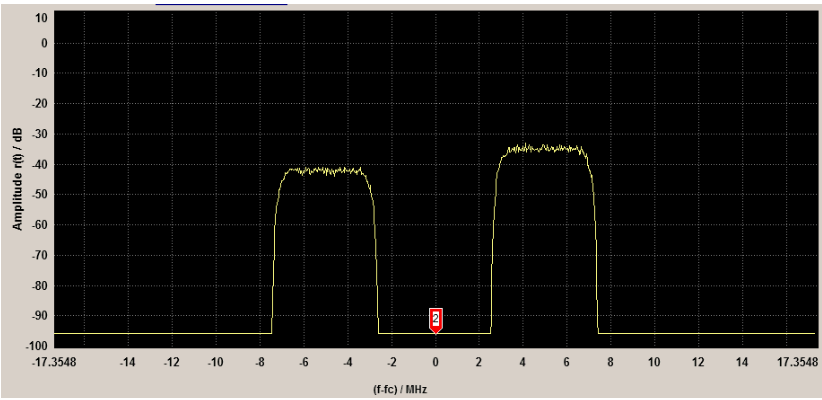
A multi carrier signal is composed from two waveform files with the following characteristics:

3GPP FDD signals with the same "Sample Rate" but *different "Crest Factors"*.

"Peak" = 0 dBFS

"Carrier Gain" = 0 dB

	State	Carrier Freq. Offs. [MHz]	Gain [dB]	Phase [deg]	Delay [ns]	File	Info
0	On	-5.000 000	0.00	0.00	0	....Data/Lists/Arb/ArbMccwDummy	Info...
1	On	5.000 000	0.00	0.00	0	....Data/Lists/Arb/ArbMccwDummy	Info...

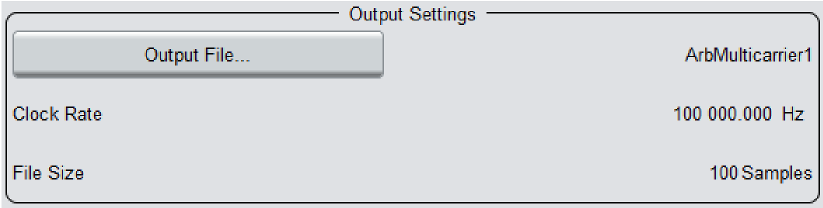


The Peak values of the carrier signals are equal, the RMS values are different.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:POWer:REFerence` on page 305

Output Settings

The output settings indicate the name, the size and the clock rate of the currently calculated multi carrier output file.



Output File ← Output Settings

Accesses the standard "File Select" function to specify the output file name of the multi carrier waveform to be calculated. As with normal waveforms, the file extension is `*.wv`.

To trigger the calculation and storage of this multi carrier waveform, select "State > On".

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:OFILe` on page 312

Clock Rate ← Output Settings

Displays the resulting sample rate at which the multi carrier waveform is output by the ARB generator. The output clock rate depends on the number of carriers, carrier spacing and input sample rate of the leftmost or rightmost carriers.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLOCk?` on page 308

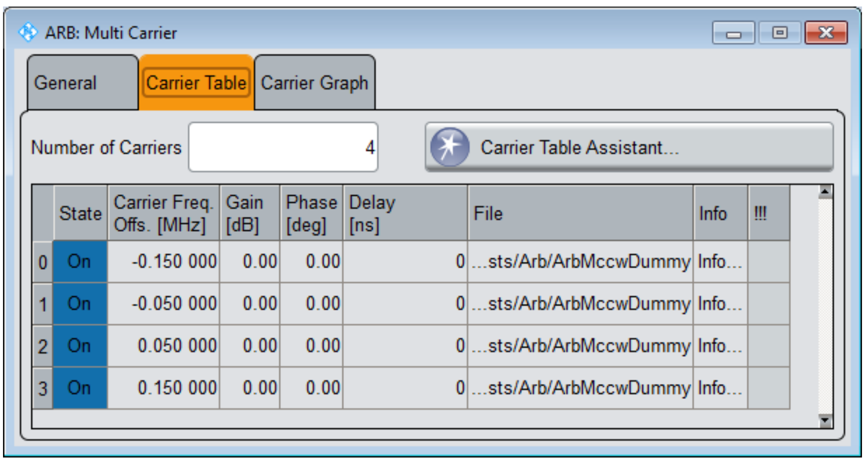
File Size ← Output Settings

Displays the resulting number of samples of the multi carrier waveform.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:SAMPles?` on page 312

4.6.3.2 Carrier Table Settings

► To access the carrier table settings, select "Baseband > Multi Carrier > Carrier Table".



This tab comprises additional settings for configuring individual carriers. A selectable carrier range can be set with the aid of the optional "Carrier Table Assistant". The settings of all available carriers are displayed in table form. Previously applied assistant settings can be further refined. The number of lines corresponds to the number of carriers.

The phase/deg settings are only effective if you have deactivated optimization of the crest factor ("Optimize Crest Factor > Off").

Use the "Carrier Graph" tab to visualize the selected multi carrier configuration.

Settings:

Number of Carriers

Sets the number of carriers for the multi carrier waveform. By default the multi carrier table lists 1 carrier. A maximum of 512 carriers can be configured and activated.

Tip: You can find and modify the number of carriers in both, the "General" tab as well as in the dialogs "Carrier Table" and "Carrier Table Assistant".

When the number of carriers is increased, the multi carrier table is extended by adding further lines at the end of the table. If these carriers already have been defined before, the settings are preset according to the former settings. Otherwise the parameters are preset with the default values.

See also Chapter 4.6.2.1, "Defining the Carrier Frequency", on page 109.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:COUNt` on page 302

No

Indicates the carrier index ranging from 0 to (number of carriers -1).

Individual carriers can be set using the remote control commands by specifying the index in the parameter CARR.

State

Activates a carrier.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STATe` on page 312

Carrier Freq [MHz]
Sets the carrier frequency.

Note: The carrier frequency can only be set in "Arbitrary Carrier Frequency" mode. For "Equidistant Carrier Spacing", the carrier spacing is determined automatically.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:FREQuency
on page 304

Gain
Sets the gain of a carrier.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:POWer on page 305

Phase
Sets the starting phase of a carrier.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:PHASe on page 304

Delay
Sets the starting delay of a carrier.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:DELay on page 303

File
Accesses the standard "File Select" function for selecting the input file with the I/Q signal to be modulated onto the carrier.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:FILE on page 304

Info
Indicates the sample rate, number of I/Q value pairs (number of samples), and the resulting signal period of the selected I/Q input file.

Remote command:
n.a.

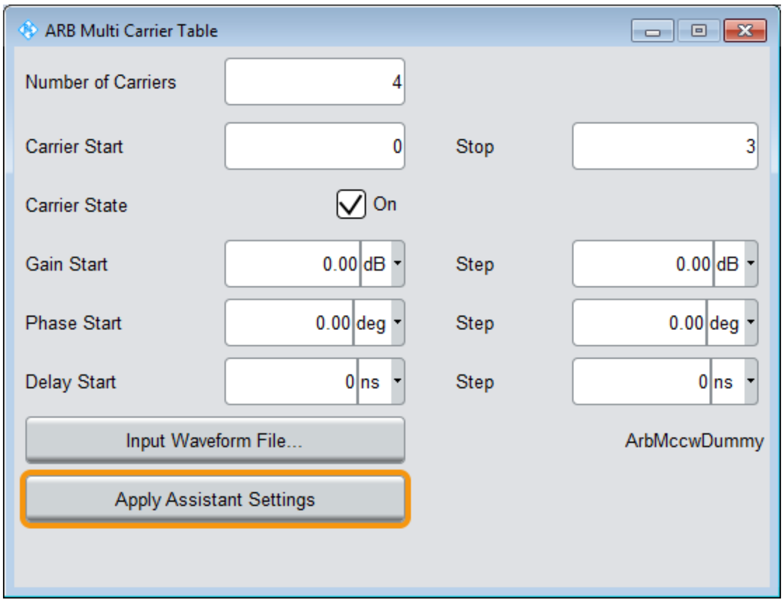
!!!
Indicates an occurred conflict by means of a warning triangle. A conflict arises when the carriers overlap and is also indicated in the header of the "Carrier Table" tab.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:CONFlict?
on page 303

4.6.3.3 Carrier Table Assistant

1. To access the dialog, select "Baseband > Multi Carrier".

2. Select "Carrier Table > Carrier Table Assistant"



The "Carrier Table Assistant" dialog enables you to configure a selectable subset of carriers. Using the assistant, you can quickly set up a multi carrier scenario within a specified carrier range.

Settings:

Number of Carriers

Sets the number of carriers for the multi carrier waveform. By default the multi carrier table lists 1 carrier. A maximum of 512 carriers can be configured and activated.

Tip: You can find and modify the number of carriers in both, the "General" tab as well as in the dialogs "Carrier Table" and "Carrier Table Assistant".

When the number of carriers is increased, the multi carrier table is extended by adding further lines at the end of the table. If these carriers already have been defined before, the settings are preset according to the former settings. Otherwise the parameters are preset with the default values.

See also Chapter 4.6.2.1, "Defining the Carrier Frequency", on page 109.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:COUNt on page 302

Carrier State
Activates the carriers within the range "Carrier Start" to "Carrier Stop".

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STATe on page 312

Carrier Start/Stop
Defines the start/stop index of the carrier range the assistant settings apply to.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STARt on page 311
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STOP on page 311

Gain Start
Sets the gain of the carrier marked by "Carrier Start".

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:POWer[:STARt]
on page 311

Gain Step
Sets the step width that is used to increment the gain.

The resulting carrier gain in the carrier table equals:

Gain = "Gain Start" + n*"Gain Step"

Where **n** ranges from 0 to ("Carrier Stop" - "Carrier Start").

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:POWer:STEP
on page 311

Phase Start
Sets the phase of the carrier marked by "Carrier Start".

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:PHASe[:STARt]
on page 310

Phase Step
Sets the step width that is used to increment the phase.

The resulting phase in the carrier table equals:

Phase = "Phase Start" + n"Phase Step"*

Where n ranges from 0 to ("Carrier Stop" – "Carrier Start")

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:POWer:STEP
on page 311

Delay Start
Sets the delay of the carrier marked by "Carrier Start".

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:DELay[:STARt]
on page 309

Delay Step
Sets the step width that is used to increment the delay

The resulting delay in the carrier table equals:

Delay = "Delay Start" + n"Delay Step"*,

Where n ranges from 0 to ("Carrier Stop" – "Carrier Start").

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:DELay:STEP
on page 308

Input Waveform File
Accesses the standard "File Select" function for selecting the input file with the I/Q signal to be modulated onto all carriers of the selected carrier range.

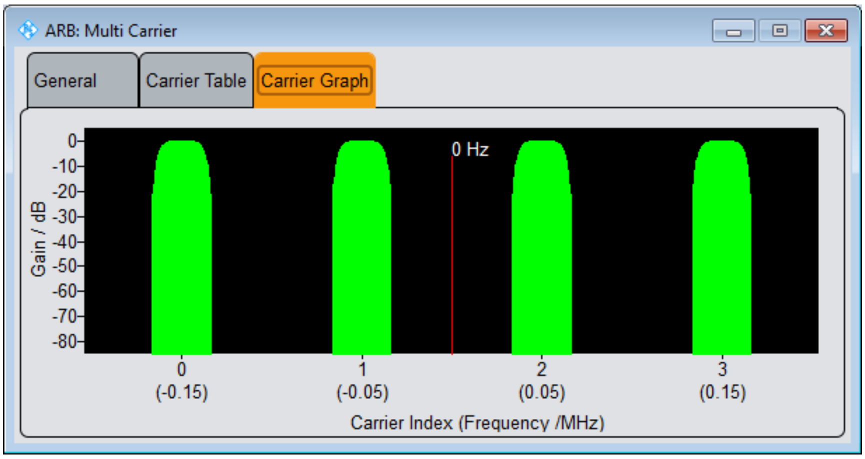
Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:FILE on page 310

Apply Assistant Settings
Transfers the assistant settings to the carrier table.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:EXECute
on page 309

4.6.3.4 Carrier Graph

1. To access the dialog, select "Baseband > Multi Carrier".
2. Select "Carrier Graph"



The carrier graph is a graphical representation of the current multi carrier configuration in the frequency domain.

The height of the bars corresponds to the selected gain of each individual carrier. The bandwidth of the carrier signals is indicated by the width of the bars.

4.6.4 How to Use the Multi Carrier Function

This section provides step-by-step instructions on how to configure and use the multi carrier settings.

To create a multi carrier waveform file (general workflow)

Perform the following general steps:

1. To configure the general settings, select "Baseband > Multi Carrier > General".
2. To configure the carrier table select "Carrier Table"
3. To enter the file name of the multi carrier waveform, select "General > Output File"
4. To save or load a multi carrier waveform, select "State > On".

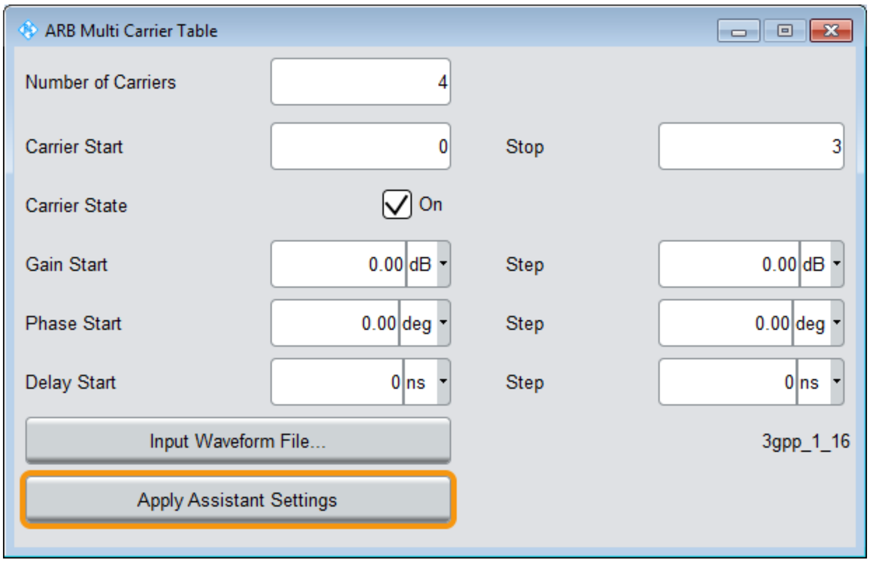
To generate a multi carrier signal for standard transmitter tests

High power amplifiers of multi carrier base stations face increased requirements in terms of linearity and acceptable intermodulation performance.

To set up a standard transmitter test, proceed as follows:

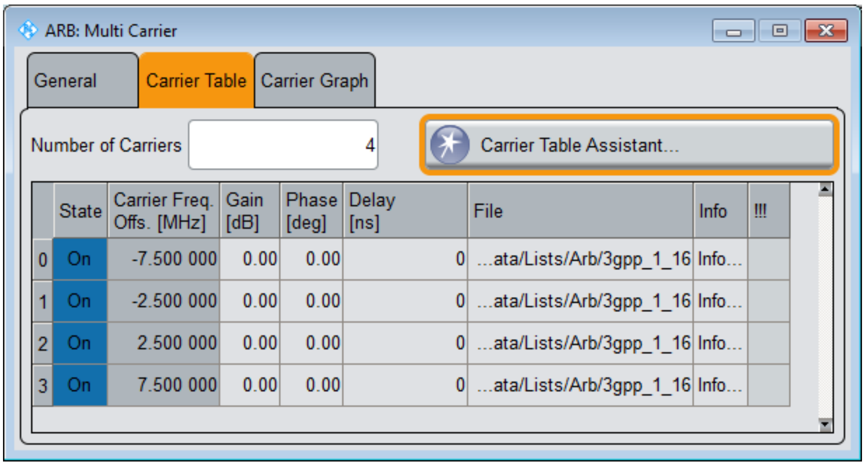
1. Select "Baseband > 3GPP FDD".
2. Adjust the following parameters:
   a) Select "3GPP FDD > Basestations > Test Setups/Models"
   b) Select "3GPP FDD: Downlink/Test Model > Test_Model_1_16channels".
   c) Confirm with "Select".
3. To enable signal generation, select "3GPP FDD > General > ON".
4. To create a 3GPP waveform file, perform the following:
   a) Select "Generate Waveform".
   b) In the destination folder, enter the "File Name" for the waveform file, for example `3gpp_1_16`.
   c) Confirm with "Save"
5. To set up a multi carrier scenario with 4 carriers and a carrier spacing of 5 MHz, select "Baseband > Multi Carrier".
6. Adjust the following parameters:
   a) In the "General" tab, select "Number of Carriers = 4" and "Carrier Spacing = 5 MHz".
   b) Select "Carrier Table > Carrier Table Assistant".
   c) Select "Carrier Start = 0".
   d) Select "Carrier Stop = 3".

e) To load the generated waveform file to all 4 carriers, select "Input Waveform File".
f) Navigate to the generated waveform file `3gpp_1_16`.
g) Confirm with "Select".
h) Select "Carrier State > On".

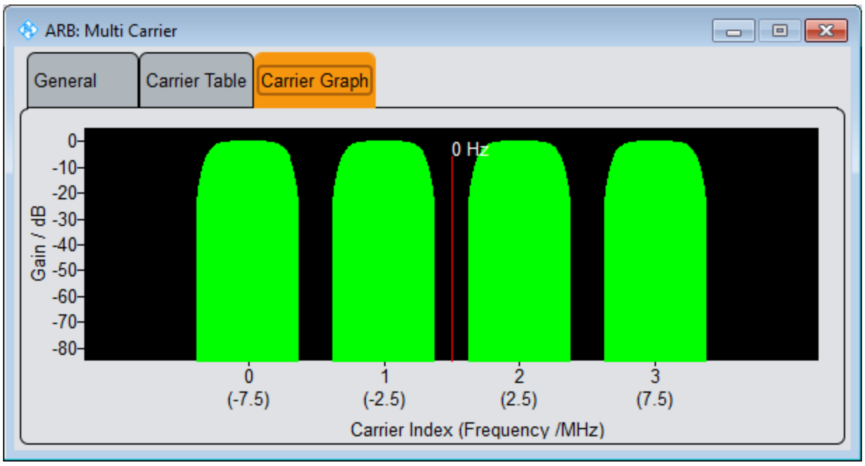


i) Select "Apply Assistant Settings".
j) Close the dialog.

The "ARB: Multi Carrier > Carrier Table" dialog confirms the configuration.



7. Select "Carrier Graph" to visualize the configuration.



8. To store the multi carrier waveform, select "General > Output File".
9. Enter the name for the file, for example `4x5MHz_3gpp_1_16`.
10. To save or load a multi carrier waveform, select "State > On").
11. To transmit the waveform to an ARB generator, proceed as described in Chapter 8.2, "How to Transmit Waveform Data to Instruments or Files", on page 203.

4.6.5 References

Table 4-12: Multi carrier default values

Parameter	Value
Number of Carriers	1
Carrier Spacing	0 MHz
Crest Factor Mode	Off
Signal Period Mode	Longest file wins

4.7 Generating Multi Segment Waveform Files

Modern chip technologies implement several communication standards within one chip and rise spatial verification and test requirements. To fulfill the requirements of these test systems and to enable rapid alternation between different waveforms with differing test signals, R&S WinIQSIM2 provides the functionality to create multi segment waveform files.

This section introduces the concept of multi segment waveform files, including the setting parameters and some typical configuration examples.

4.7.1 Required Options

R&S WinIQSIM2 offers all included digital standards for the creation of waveform files. However, the used instrument must be fitted with the associated R&S WinIQSIM2 options for the generation of the signal from the waveform file.

4.7.2 About Multi Segment Waveforms

A multi segment waveform is a signal composed of several different waveforms, called segments. Each segment contains a fully independent waveform that can be generated with its own specific markers and clock settings. You can also implement blank segments, i.e. segments containing a zero signal.

The Figure 4-8 shows the principle of building a multi segment waveform.

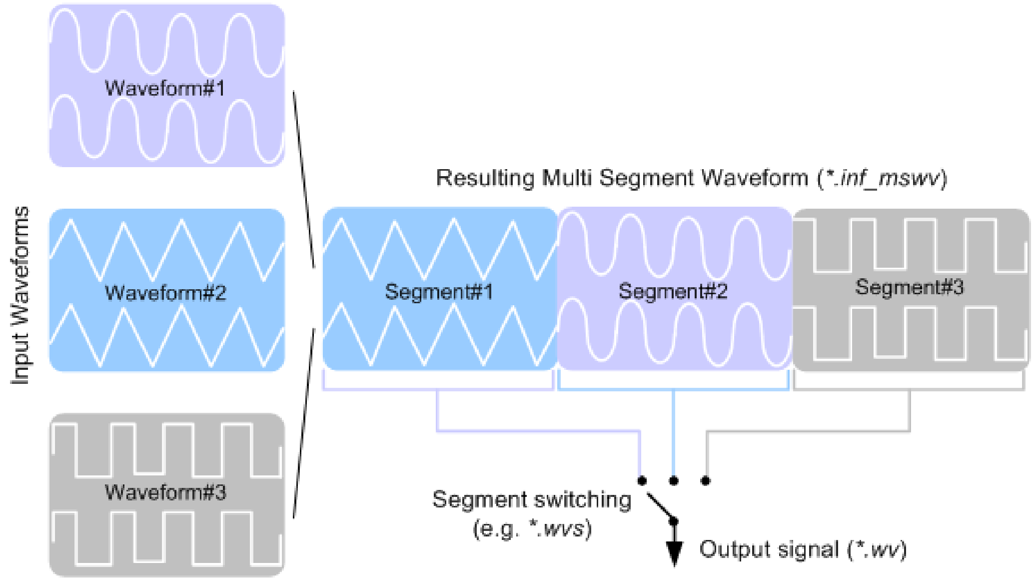


Figure 4-8: ARB Multi Segment Waveform concept

Typical applications for the multi segment mode are described in section Chapter 4.7.4, "How to Create and Work with Multi Segment Waveform Files", on page 134.

4.7.2.1 Multi Segment Waveforms Processing

A multi-segment waveform file is configured and stored with R&S WinIQSIM2. If connected, you can directly transmit the waveform file to a signal generator, that plays the file and generates the signal, triggered by the functions "State" button. An instrument loads the entire multi segment waveform into the memory, and thus allows you to alternate between the individual waveforms without delay by loading. Moreover, in most instruments you can define the output order of the segments, as well as the segment intended to be output at any given moment. See the manual of the used generator to get more information on the processing of waveform files .

4.7.2.2 File Concept

To provide flexible configuration, building a composed multi segment waveform file involves different stages; by completing of each of them, R&S WinIQSIM2 creates and stores a dedicated file. The following files are used:

- **Configuration list:** is a dedicated file with details on how a multi segment waveform is made up from different waveforms, the level and the clock rate settings, and the file name. The file extension is `*.inf_mswv`.
  You can create any number of configurations as a basis for defining further multi segment waveforms.
- **Output file:** is the created output multi segment waveform file. R&S WinIQSIM2 stores it under a user definable name; as with the standard waveforms, the used file extension is `*.wv`. R&S WinIQSIM2 appends additional information to the header of the composed waveform file, e.g. user comments.

4.7.2.3 Impact of the Marker Settings

The general purpose of the marker signal is the triggering of the DUT (device under test) and the synchronization with other measurement instruments.

To be optimally flexible for those tasks, you can define additional marker signals with R&S WinIQSIM2:

- Segment markers
  Since multi segment waveforms act as stand alone waveforms, that can also contain marker signals. You can either include this marker signals in the processing of the composed multi segment waveform signal, or exclude.
- Additional segment restart and sequence restart markers
  Additional marker signals can be defined to restart the multi segment sequence or to restart each of the multiple segments. A marker signal configured and defined for one of these purposes over-writes markers that are defined in the individual waveforms, and markers that are assigned to the same output connector of an instrument.

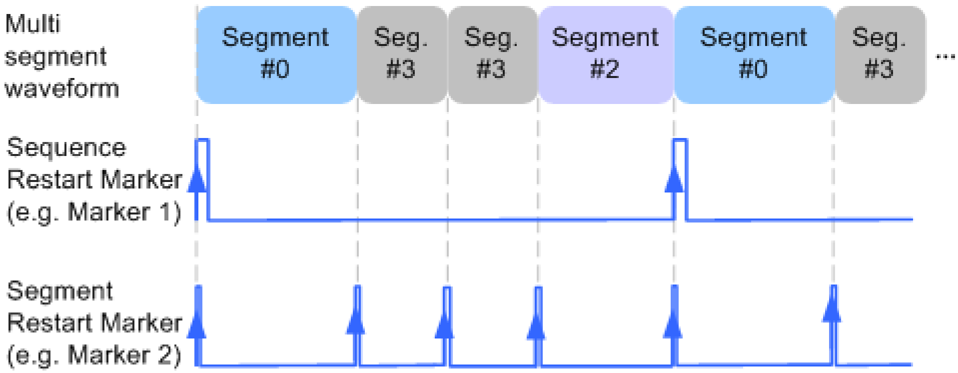


Figure 4-9: Example of marker signals

The segment start is defined by the rising edge of the marker. It applies for switching between two segments as well as in case of segment replay.

4.7.3 Multi Segment Settings

► To access the multi segment settings, select "Baseband > Multi Segment".

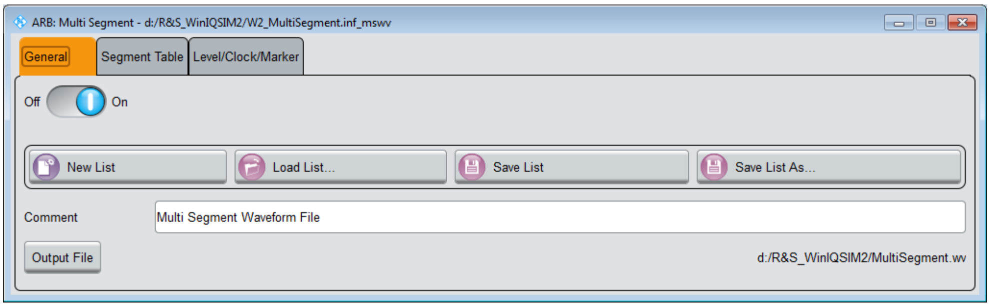
The "ARB: Multi Segment" dialog enables direct assignment of waveforms to the multiple segments, adjusting the clock, level, and marker settings of the composed waveform, and selecting the output file.
For description on the provided settings, refer to Chapter 4.7.3.1, "Settings for the Handling of Multi Segment and Output Files", on page 128, Chapter 4.7.3.2, "Segment Table Settings", on page 129 and Chapter 4.7.3.3, "Level / Clock / Marker Settings", on page 131.

The remote commands required to define these settings are described in Chapter 12.8, "SOURce:BB:ARB:WSEG Subsystem", on page 315.

4.7.3.1 Settings for the Handling of Multi Segment and Output Files

To access the general settings:

1. Select "Baseband > Multi Segment".



The "General" tab comprises standard functions for file handling, like selecting and loading of files, determining the output file name.
The provided settings depend on the current waveform. The "Sequencing List" function for instance requires that the current "Sequence Table" contains more than one segment.

2. Perform one of the following:
   - Select "New List" to create a multi segment waveform file
   - Select "Load List" to load an existing one
3. Add a comment.
4. Select "Output File" to define the file name for the multi segment waveform file.

Settings:

State

Activates multi segment waveform generation.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:STATe` on page 324

New List

Accesses the standard "Create Multi Segment Waveform List" function to enter the name of the new file.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SELect` on page 321

Load List

Accesses the standard "File Select" function to select the configuration file to be edited.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CATalog?` on page 316
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SELect` on page 321

Save List/Save List As...

Saves the current entries of the Segment Table Settings in a configuration list (file), including the level mode, clock mode, segment marker mode and output file name settings.

See also Chapter 4.7.2.2, "File Concept", on page 127.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:OFILe` on page 320

Comment

Adds a comment to the composed multi segment file.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:COMMent` on page 318

Output file

Accesses the standard "File Select" dialog function and requests a file name.

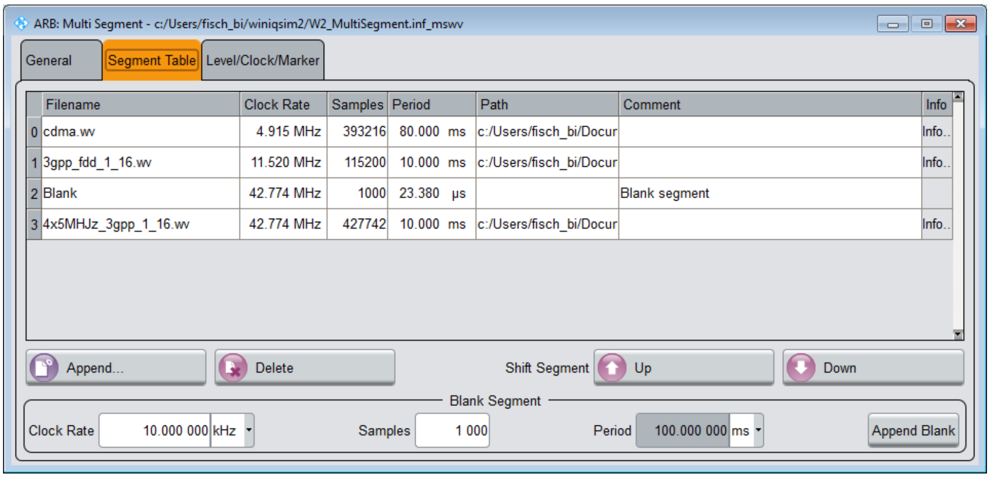
The output file name is required for the internal storage of the multi segment waveform, triggered by the "Save List" function.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:OFILe` on page 320

4.7.3.2 Segment Table Settings

1. To access the multi segment table settings, select "Baseband > Multi Segment".

2. In the "ARB: Multi Segment > General" tab, select an existing list.
3. Select "Segment Table".
4. Select "Append" to add *existing* waveform segments.
5. Select "Append Blank" to add a blank segment.
6. Configure the "Blank Segment Settings" as required.



This section contains the parameters required to create and configure settings of the multi segment waveform file.

Multi Segment Table, Append/Delete/Shift Seg.

The table lists the individual waveforms (segments) of the selected multi segment waveform. The information about the segments is retrieved from the tags of the corresponding waveform files.

Table 4-13: GUI elements in the multi segment table

Function	Description
"Segment#"	Indication of segment index. Within the manual and remote control configuration, this segment index indicates the segment explicitly. **Tip:** Use the segment index for example to define the subsequent sequence for output in the mode "Next Segment" or "Next Segment Seamless".
"Waveform"	Indication of the waveform file name of the segment.
"Clock Rate"	Indication of the clock rate of the selected waveform.
"Samples"	Indication of the number of samples in the segment.
"Period"	Indication of the segment duration.
"Path"	Indication of the location of the waveform file used for the corresponding segment.
"Comment"	Indication of the possible comment contained in the waveform.
"Info"	Opens a dialog with detailed information about the currently selected waveform.

"Append" Opens the standard "File Select" dialog for navigation to and selection of the waveform file to be added on the end of the existing list. Only non-multi segment waveforms can be loaded.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SEGMent:CATalog?
on page 321
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SEGMent:APPend
on page 320

"Delete" Removes the selected entry from the table. The waveform file itself is however not deleted.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:DELete on page 318

"Shift Seg.# Up/Down"
Rearranges the segments, i.e. moves the selected segment up and down.

Blank Segment

Comprises the settings of a blank segment. A blank segment is a zero signal with defined clock rate and number of samples.

"Clock Rate" Selects the clock rate of the blank segment.

"Samples" Selects the number of samples for the blank segment.

"Period" Displays the resulting period for the blank segment.

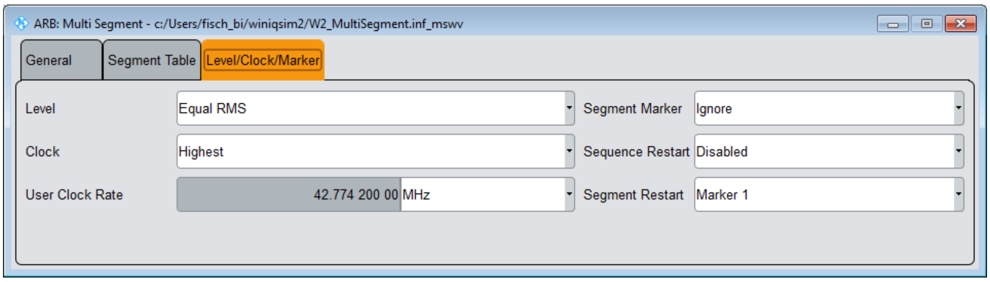
"Append Blank" Adds the blank segment to the multi segment file.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:BLANk:APPend
on page 316

4.7.3.3 Level / Clock / Marker Settings

1. To access these settings, select "Baseband > Multi Segment".
2. In the "ARB: Multi Segment > General" dialog, select an existing list.
3. Select "Segment Table".
4. Configure the segments.

5. Select "Level/Marker/Clock".



The "Level/Marker/Clock" tab provides the parameters required to adjust the level, marker and clock settings for the selected multi segment waveform file.

Settings:

Level

Determines the signal level mode to be used for generating the multi segment signal.

"Unchanged" When generating the signal, the level meets the values defined in the individual segment files.

"Equal RMS" The signal is generated with the same RMS level value for all segments.

Remote command:
`[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:LEVel[:MODE]`
on page 318

Clock

Determines the clock rate mode to be used for generating the multi segment signal.

"Unchanged" Uses the clock rate that is defined in the corresponding waveform file of each segment.

"Highest" Uses the highest available clock rate for all segments.

Note: Trade-off between fast switch over and computing time. This mode provides short switch over times between segments.
However, the computing time increases because the individual segments have to be resampled.

"User" Uses the clock rate defined by the parameter "User Clock" for all segments.
This mode is a trade-off between fast switch over and computing time, too.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CLOCk:MODE
on page 317

User Clock Rate
Determines a user-specific clock rate for all segments of the multi segment waveform.

This value applies to "Clock Mode > User".

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CLOCk on page 317

Segment Marker
Determines the marker signal mode to be used for generating the multi segment waveform signal, see also Chapter 4.7.2.3, "Impact of the Marker Settings", on page 127.

"Ignore" The marker information in the individual segment waveform files is not considered.

"Take Over" Uses the marker information of the individual segment waveform files.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:MODE
on page 320

Sequence Restart
Activates the generation of an additional sequence restart marker in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered, see also Chapter 4.7.2.3, "Impact of the Marker Settings", on page 127.

"Disable" No additional marker is generated.

"Marker 1, 2, 3"
Generates a restart marker signal at the beginning of the first segment of the complete multi segment sequence.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:FSEGment
on page 319

Segment Restart
Activates the generation of an additional segment restart marker in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered, see also Chapter 4.7.2.3, "Impact of the Marker Settings", on page 127.

"Disable" No additional marker is generated.

"Marker 1, 2, 3"
Generates a restart marker signal at the beginning of each segment. The segment start is defined by the low-high slope of the marker signal. It applies for switching between two segments as well as in case of segment replay.

Remote command:
[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:ESEGment
on page 319

4.7.4 How to Create and Work with Multi Segment Waveform Files

This section provides step-by-step instructions on how to configure and use the multi segment settings. The basic workflow in Figure 4-10 shows the essential stages of multi segment waveform configuration, together with the corresponding user interface dialogs and if applicable, the corresponding file extensions.

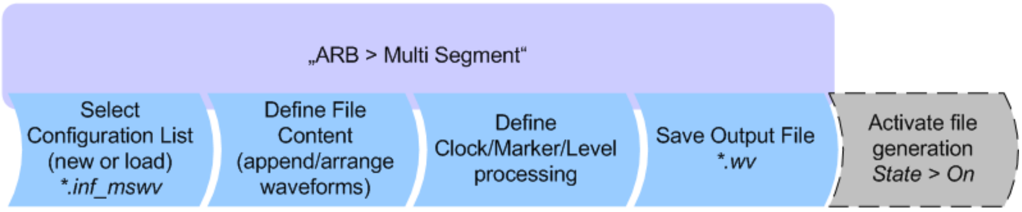


Figure 4-10: Basic workflow for generation of multi segment waveforms

To generate a multi segment waveform file (basic workflow)

1. Select "Baseband > Multisegment".
2. In the "General" tab, select "New List" to create an empty list.
3. Use the "ARB: Multi Segment > Segment Table > Append" function to add two or more waveform files.
4. In the "Level/Clock/Marker" tab, adjust the "Level", "Clock" and "Segment Marker" settings.
5. In the "General" tab, select "Output File".
6. Enter a file name.
7. Confirm with "Save".
8. Select "Save List" to store the configuration in the file.
9. Select "State > On".

To add additional marker signals

To add marker signals at the beginning of each segment and at the beginning of the sequence, additional marker signals are enabled and inserted into the multi segment waveform.

1. In the "Multi Segment > Level/Clock/Marker" dialog, select "Segment Marker = Ignore"
2. Enable "Level/Clock/Marker > Sequence Restart = Marker 1"

3. Enable "Level/Clock/Marker > Segment Restart = Marker 2"

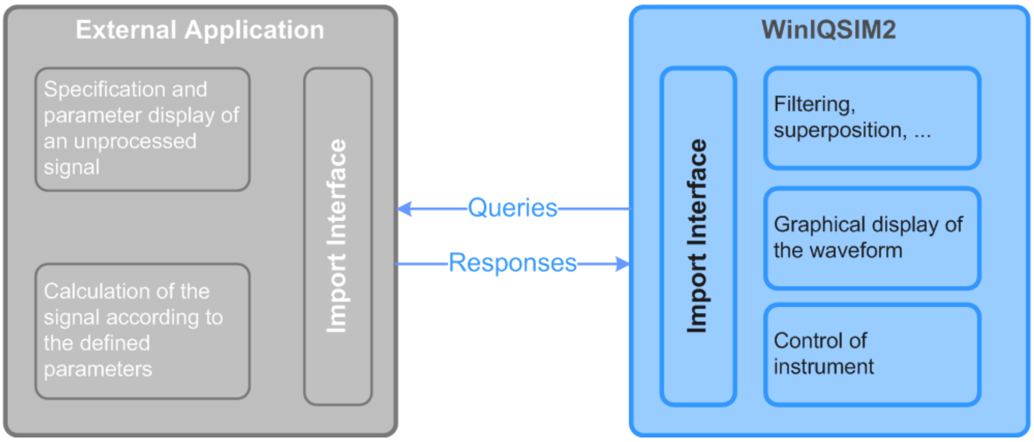
   An additional restart marker signal is generated at the beginning of each segment and the beginning of the waveform. These restart markers *overwrite* the existing Marker 1 and Marker 2 trace signals defined in the waveforms of the individual segments.

4.8 Import IQ Data

The import function allows you to load externally created I/Q samples via a server connection into R&S WinIQSIM2.

Created on a user-specific system, you can transfer unprocessed external signals directly to R&S WinIQSIM2. Once imported, R&S WinIQSIM2 provides all processing stages like filtering, graphics display or adding noise, and even directly transmitting the waveform signal to a connected instrument.

R&S WinIQSIM2 and the user-specific application communicate over LAN and the TCP/IP network protocol, provided the TCP/IP driver is installed on the corresponding computers. Using this link, programs can always communicate even if they run on different PCs and different operating systems. For information on the LAN interface and the structure of the associated IP address, see Chapter 11.1.1, "LAN Interface", on page 230.



The external application acts as a server for the data. In the following description, it is referred to as *import server*. For communication R&S WinIQSIM2 sends queries to the server, and receives the corresponding replies.

The applications exchange the following information:

- Server identity
- Parameter status
- Sequence length
- Waveform

4.8.1 About the Import Interface

This section explains the programming of the import interface using the example of the development environment *National Instruments LabWindows CVI xxxx Full Development System*. In principle, other development environments use similar TCP/IP commands, therefore see the manuals of the respective development systems for details.

Network communication using TCP/IP consists of a server and one or more several clients. The server provides the data and transfers it on request or spontaneously to the client. The client receives the incoming data.

Thus, when importing waveform files, R&S WinIQSIM2 acts as client and the external application as server. Client and server exchange the user data in ASCII format, and the messages as binary blocks.

4.8.1.1 Server Name, Port Number and Items

To build a link to the server, the client must know the name of the server (e.g. 12.34.56.78). In addition, both participants need access to a common channel, the TCP/IP port.

The default port address is 1000, but you can select any other value.

Note that certain ports are pre-allocated, such as 80 for browsers. Assign therefore only a freely available port address.

Table 4-14: TCP/IP items from client to server

Item	Meaning
`GetServerID`	ID query
`GetNewState`	Query if new data are available
`GetSequenceLength`	Query of sequence length
`GetSymbolRate`	Query of symbol/reference point rate in Hz
`GetSymbol`	Query of the symbols/reference points

4.8.1.2 Format of Query Commands of the Import Client

Query commands are sent with `ClientTCPWrite (gHandle, request, strlen(request), 1000);` the last parameter timeout is given in ms.

4.8.1.3 Format of Reply Commands (Messages) of the Import Server

Replies are sent with `ServerTCPWrite ().`

R&S WinIQSIM2 must receive the reply within 10 seconds, otherwise the timeout takes effect.

The calculation and the transmission of the signal typically require a certain time. Therefore, it is important that the calculation is performed in advance, and not triggered by renewed requests. R&S WinIQSIM2 queries regularly for data with the command `GetSymbols`, and the server provides the calculated signal from a buffer.

To distinguish between replies to different queries, a header is placed in front of the data section of the reply:

```
typedef enum { SYMBOLRATE, SEQUENCE_LENGTH, SYMBOLS, NEWSTATE, SERVER_ID }
SERVER_MESSAGE_HEADER_TYPE;
typedef struct
{
SERVER_MESSAGE_HEADER_TYPE type;
float data;
}
SERVER_MESSAGE_TYPE;
SERVER_MESSAGE_TYPE Reply;
Reply.type = SEQUENCE_LENGTH;
Reply.data = (float)SequenceLength;
ServerTCPWrite (gHandle, (void *)&Reply, sizeof(Reply), 1000);
```

ServerID, NewState and the symbol rate are handled the same way as the sequence length.

The format above cannot be used if the symbol value is to be sent to R&S WinIQSIM2 because the message is longer than the size of element data in structure `SERVER_MESSAGE_TYPE`.

A free memory block must be filled then with data as follows:

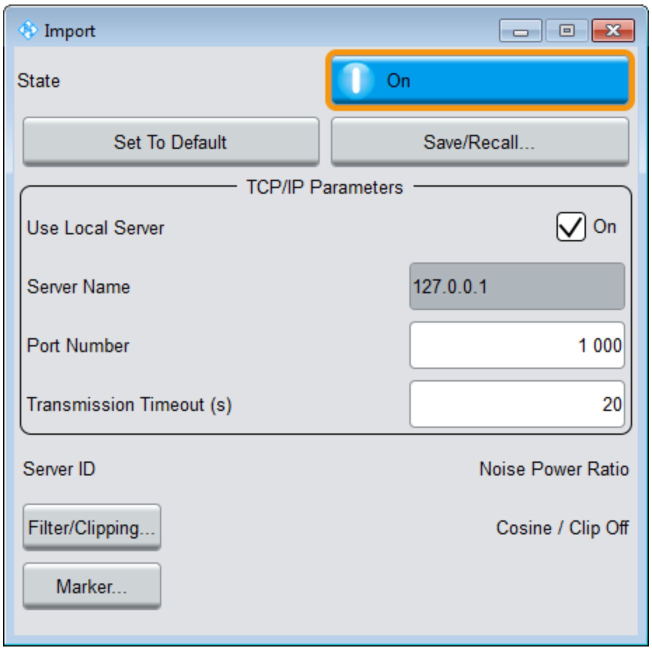
```
Pos 1 : SYMBOLS (enum, = 2)
Pos 2 : I(0)
Pos 3 : Q(0)
Pos 4 : I(1)
Pos 5 : Q(1)
....
Pos 2*N : I(-1)
Pos 2*N+1 : Q(-1)
```

A pointer on this block is then transferred to the function `ServerTCPWrite()`.

Alternatively to transmitting the waveform in one block, the samples can be divided into several blocks. In the latter case, only the first block is sent with a header, the following blocks are without header.

4.8.2 Import Settings

► To access the import settings, select "Baseband > Import...".



The "Import" dialog contains the parameters for configuring the TCP/IP connection to the server, and provides access to the filter, clipping and marker settings.

The remote commands required to define these settings are described in Chapter 12.10, "SOURce:BB:IMPort Subsystem", on page 352.

State

Activates the import of IQ data.

Remote command:

`[:SOURce<hw>]:BB:IMPort:STATe` on page 356

Set to Default

Sets all relevant parameters to default, see Table 4-15

Remote command:
[:SOURce<hw>]:BB:IMPort:PRESet on page 353

Save/Recall
Accesses the "Save/Recall" dialog, i.e. the standard function for storing and recalling the complete dialog-related settings in a file. The provided navigation possibilities in the dialog are self-explanatory.

The file name and the directory are user-definable; the file extension is however predefined (`*.import`).

See also Chapter 9, "File and Data Management", on page 206.

Remote command:
[:SOURce<hw>]:BB:IMPort:SETTing:CATalog? on page 354
[:SOURce<hw>]:BB:IMPort:SETTing:LOAD on page 355
[:SOURce<hw>]:BB:IMPort:SETTing:STORe on page 355
[:SOURce<hw>]:BB:IMPort:SETTing:DELete on page 355

Use Local Server
Determines the computer on which R&S WinIQSIM2 is running as the server from which the IQ data streams are downloaded. If activated, the field for entering a remote host is filled with the entry "127.0.0.1" automatically and is read-only.

Remote command:
[:SOURce<hw>]:BB:IMPort:SERVer:LOCal[:STATe] on page 353

Server Name
Sets the name (IP address) of the connected server.

This parameter is editable when "Use Local Server" is unchecked. Otherwise, "127.0.0.1" is used.

Remote command:
[:SOURce<hw>]:BB:IMPort:SERVer:NAME on page 353

Port Number
Defines the port number for the server connection.

See also Chapter 4.8.1.1, "Server Name, Port Number and Items", on page 136.

Remote command:
[:SOURce<hw>]:BB:IMPort:SERVer:PORT on page 354

Transmission Timeout (s)
Sets the wait time for aborting the data transfer in case of a server timeout.

Remote command:
[:SOURce<hw>]:BB:IMPort:SERVer:TTOut on page 354

Server ID
Indicates the connected import server.

If no import server is connected, "not connected" is displayed.

Remote command:
[:SOURce<hw>]:BB:IMPort:SERVer:ID? on page 353

Filter/Clipping
Accesses the dialog for configuring the baseband filter and clipping, see Chapter 4.8.3, "Filter / Clipping Settings", on page 140.

The current filter and the clipping state are displayed next to the button.

Marker
Accesses the dialog for configuring the marker signals, see Chapter 4.8.4, "Marker Settings", on page 144.

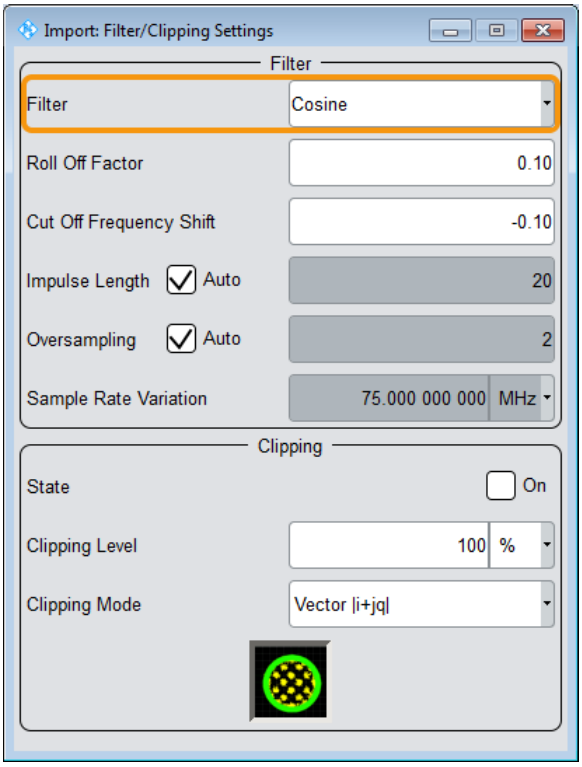
4.8.3 Filter / Clipping Settings

Access:

► Select "Import > Filter/Clipping".

The dialog comprises the settings, necessary to configure the baseband filter, sample rate variation and clipping.

4.8.3.1 Filter Settings



The upper section in the dialog contains the parameters necessary for configuring the baseband filter.

Settings

Filter
Selects the baseband filter.

Remote command:
`[:SOURce<hw>]:BB:IMPort:FILTer:TYPE` on page 359

Roll Off Factor or BxT
Sets the filter parameter.

The filter parameter ("Roll off Factor" or "BxT") depends on the currently selected filter type. This parameter is preset to the default for each of the predefined filters.

Remote command:
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine` on page 359
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:RCOSine` on page 360
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:PGAuss` on page 359
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:GAUSs` on page 359
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:SPHase` on page 360
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:APCO25` on page 360

Cut Off Frequency Shift
Sets the value for the cut off frequency shift.

The cut off frequency of the filter can be adjusted to meet spectrum mask requirements. This parameter applies to "Cosine" and "EUTRA/LTE" filters.

Remote command:
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine:COFS` on page 361

Cut Off Frequency Factor
Sets the value for the cutoff frequency factor. The cutoff frequency of the filter can be adjusted to reach spectrum mask requirements.

This parameter applies to "Lowpass" and "EUTRA/LTE" filters.

Remote command:
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASs` on page 359
`[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASSEVM` on page 359

Sample Rate Variation
Displays the sample rate of the imported signal.

If you change the sampling rate in the external application, R&S WinIQSIM2 updates the parameter immediately, even if you have not recalculated the signal.

A variation of this parameter only affects the ARB clock rate; all other signal parameters remain unchanged.

Remote command:
[:SOURce<hw>]:BB:IMPort:SRATe:VARiation? on page 356

Impulse Length
Displays the number of filter tabs.

If enabled, the most sensible parameter values are selected. The value depends on the coherence check.

Disable it to set the values manually.

Remote command:
[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth:AUTO on page 358
[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth on page 358

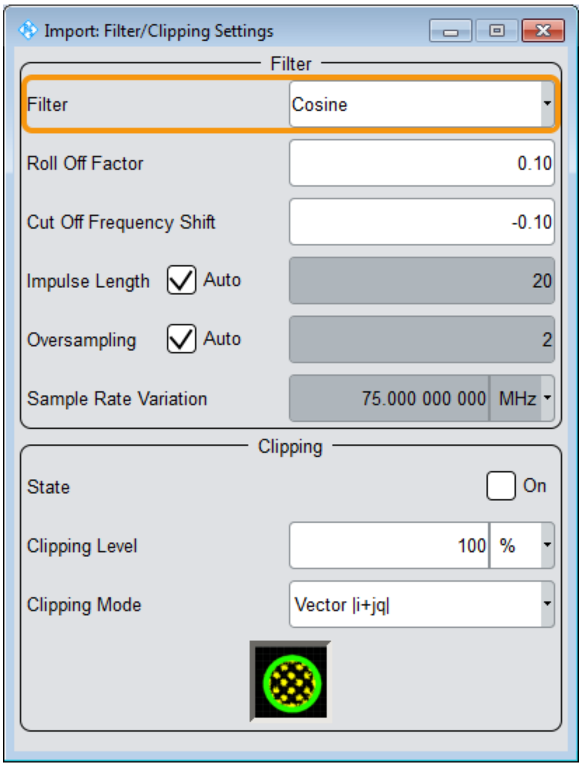
Oversampling
Sets the upsampling factor.

If enabled, the most sensible parameter values are selected. The value depends on the coherence check.

Disable it to change the value manually.

Remote command:
[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling:AUTO on page 359
[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling on page 359

4.8.3.2 Clipping Settings



The lower section in the dialog contains the parameters necessary for configuring the clipping.

Settings:

Clipping State

Switches baseband clipping on and off.

Baseband clipping is a simple and effective way of reducing the crest factor of the signal. Since clipping is done before to filtering, the procedure does not influence the spectrum. The EVM however increases.

Remote command:
`[:SOURce<hw>]:BB:IMPort:CLIPping:STATe` on page 358

Clipping Level

Sets the limit for clipping.

This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Remote command:
`[:SOURce<hw>]:BB:IMPort:CLIPping:LEVel` on page 357

Clipping Mode
Selects the clipping method. A graphic illustration of the way in which this two methods work is given in the dialog.

- "Vector | i + jq |"
  The limit is related to the amplitude | i + q |. The I and Q components are mapped together, the angle is retained.
- "Scalar | i | , | q |"
  The limit is related to the absolute maximum of all the I and Q values | i | + | q |. The I and Q components are mapped separately, the angle changes.

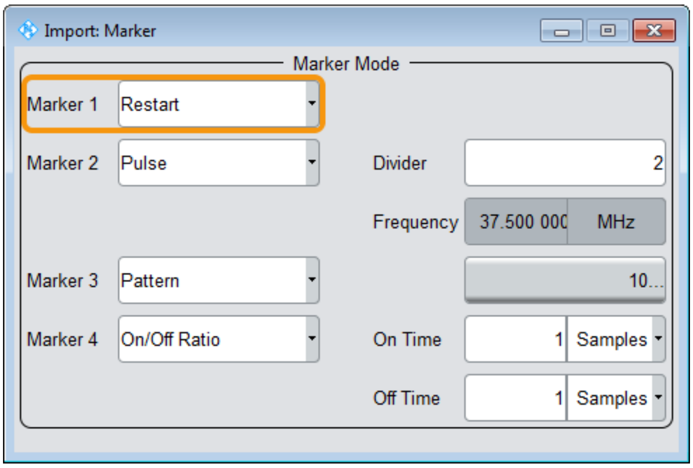
Remote command:
[:SOURce<hw>]:BB:IMPort:CLIPping:MODE on page 357

4.8.4 Marker Settings

This dialog provides access to the settings necessary to select and configure a marker output signal, like the marker mode.

For information on how these settings affect the signal, refer to Chapter 4.3.1.2, "Marker Signals", on page 59.

► To access the marker settings, select "Import > Marker...".



The dialog contains the parameters required for configuring marker output signals.

Settings:

Marker Mode
Marker configuration for up to four marker channels. The settings are used to select the marker mode defining the shape and periodicity of the markers. The contents of the dialog change with the selected marker mode; the settings are self-explanatory.

All regular marker signals are described in "Marker Modes" on page 59.

Remote command:
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:MODE on page 361
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:DIVider on page 362
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:FREQuency? on page 363
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PATTern on page 362
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:OFFTime on page 362
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:ONTime on page 362

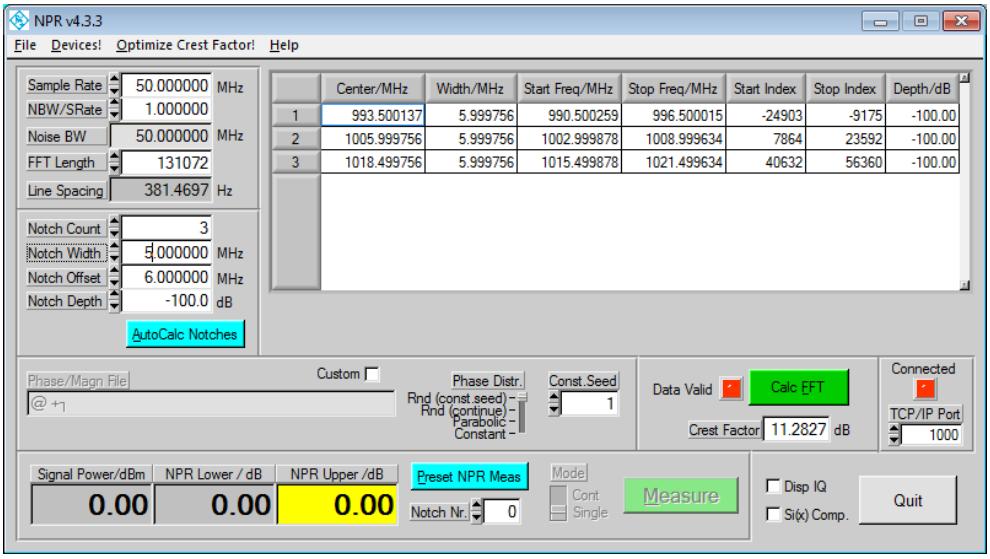
4.8.5 How to Import an External Unprocessed Signal

This section gives an example on how to import an externally generated signal to R&S WinIQSIM2. Using *R&S NPR Noise Power Ratio Signal Generation and Measurement*, an add-on tool for R&S WinIQSIM2 to generate noise power ratio signals, we create a stimulus signal for import to R&S WinIQSIM2. In the example, both applications are started on the same computer.

To establish the client to server connection

It is assumed that you have installed both, R&S WinIQSIM2 and the R&S NPR application on your computer.

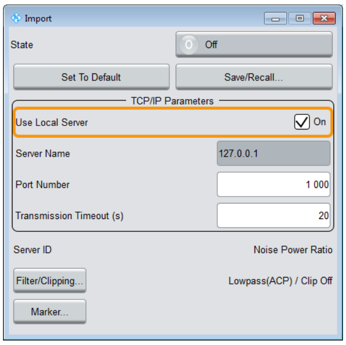
1. Start R&S WinIQSIM2.
2. To start R&S NPR, select "Windows > All Programs > R&S NPR".

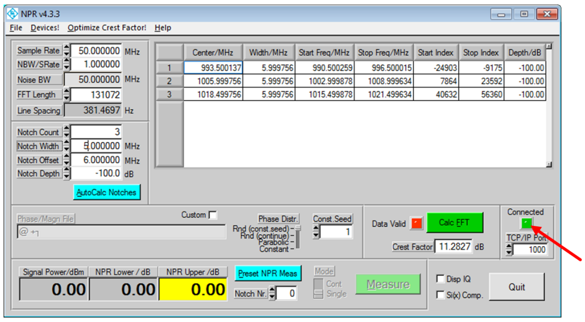


3. In R&S WinIQSIM2, select "File > New" to start from an initial state.
4. Select "Baseband > Import".
5. Check "Use Local Server".

   R&S WinIQSIM2 assigns the address of the local host automatically and displays it under "Server Name".

The connection to the server is established immediately and R&S WinIQSIM2 displays the designation of R&S NPR under "Server ID".
In R&S NPR, the "Connected" LED also indicates that the connection between the two applications is established.

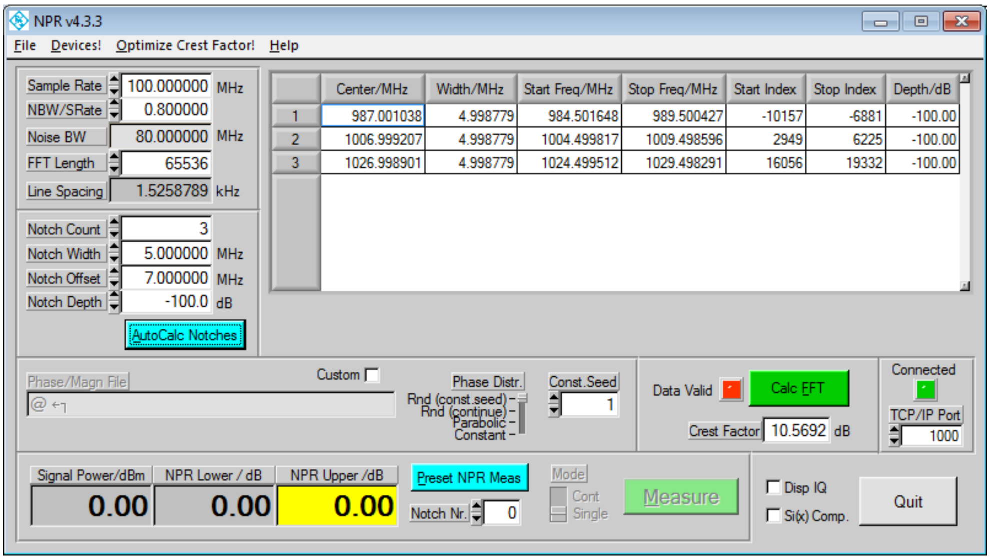




To configure a custom signal in NPR

The R&S NPR application provides all parameters required for configuring a noise ratio stimulus signal.

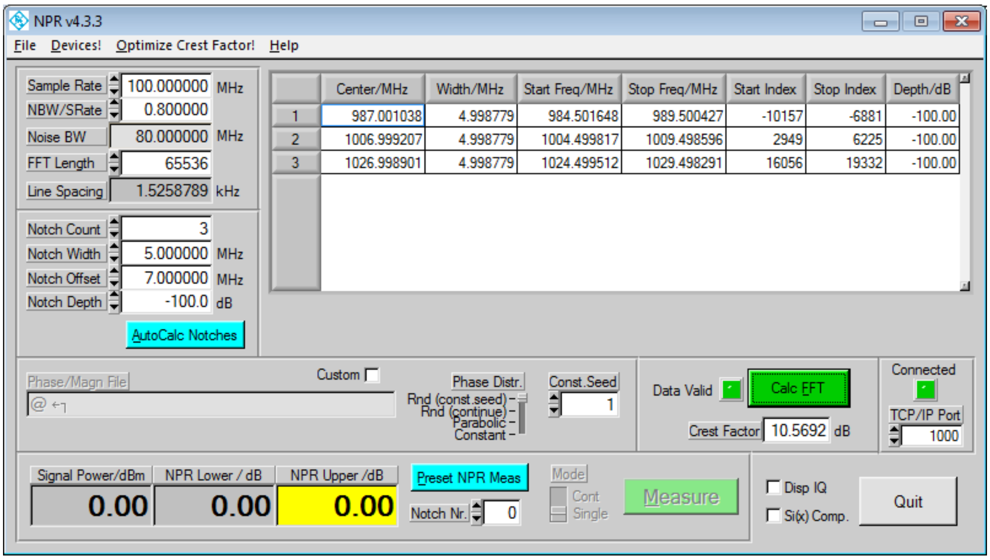
1. Configure the signal, e.g. ...



a) Set "Sample rate = 100 MHz", and "NBW/SRate = 0.8".
b) Select "FFT Length = 65536".
c) Set "Notch Count =3", "Notch width = 5 MHz", "Notch Offset = 7 MHz" and "Notch Depth = -100 dB".

2. Select "AutoCalc Notches" to update the signal parameters, displayed in the upper panel on the right.

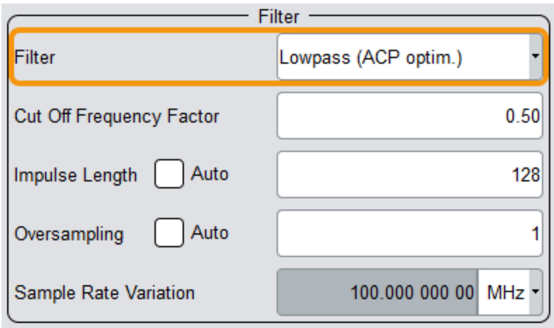
3. Select "Calc FFT" to prepare the IQ data for transmission to R&S WinIQSIM2.



The "Data Valid" LED indicates that the data is ready for transfer.

To process the signal in R&S WinIQSIM2 and validate it graphically

1. In the "Import" dialog of R&S WinIQSIM2, select "Filter/Clipping", to configure the filter, e.g. a lowpass:



   a) In the "Filter" tab, select "Filter > Lowpass (ACP optim.)".
   b) Uncheck "Impulse Length > Auto".
   c) Enter "Impulse Length = 128".
   d) Uncheck "Oversampling > Auto".
   e) Enter "Oversampling = 1".

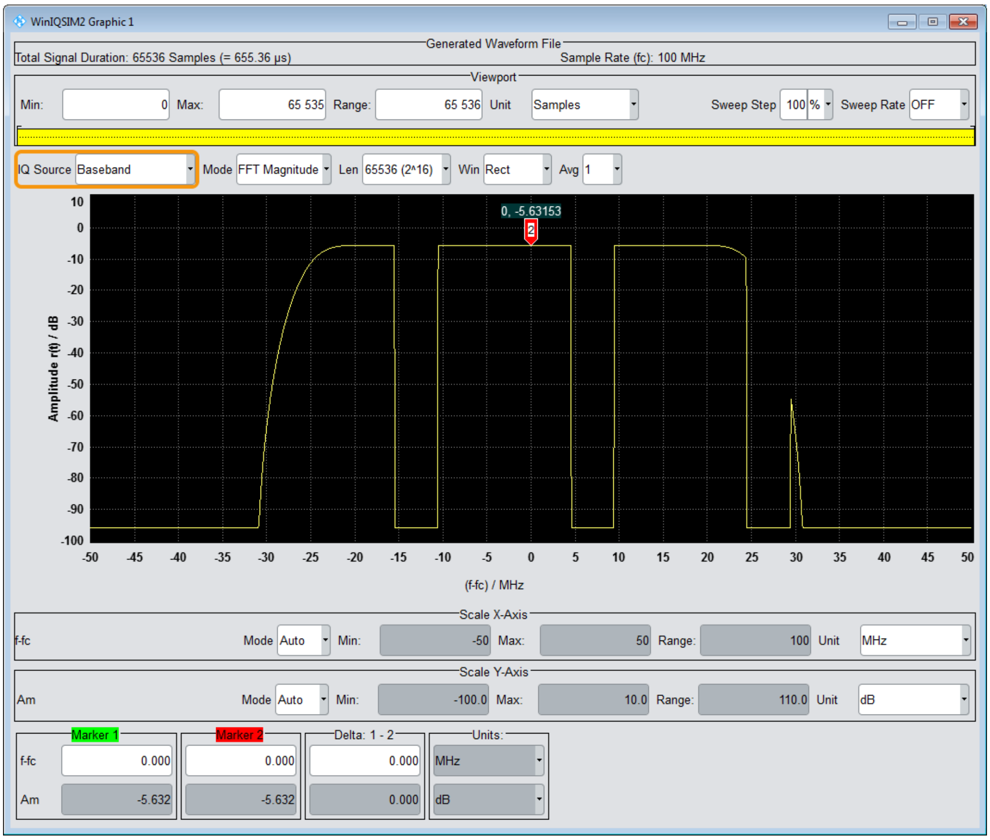
2. Close the dialog.

3. In the "Import" dialog of R&S WinIQSIM2, select "State > On" to enable data transfer.

   The IQ signal data is transmitted immediately.

To verify the signal in R&S WinIQSIM2 graphically

1. In R&S WinIQSIM2 application window, select "Graphics > Graphic 1 > Graphic1 (Complete)".
2. Configure the graphical display:
   a) Select "IQ Source > Baseband".
   b) Set "Mode > FFT Magnitude", FFT window function "Win > Rect", and FFT average mode "Avg. > 1".
   c) Select segment length "Len > 65536 (2^16)".



The graph displays the power spectrum of the imported I/Q signal calculated by FFT (Fast Fourier Transform).

Now you can further process the waveform, like e.g.:
- Verify the signal directly in the graph using markers and the zoom function, see "Using markers to analyze the signal" on page 33, or Chapter 6.5, "Scaling and Marker Settings", on page 175.
- Superimpose the signal with additional noise, see Chapter 3.3.4, "Adding Noise (AWGN) to the Signal", on page 29, or Chapter 5.1, "About the AWGN Generator", on page 150.
- Store the resulting waveform in a file, see Chapter 3.3.7, "Transmitting the Generated Waveform to a File", on page 39, or Chapter 9.7, "Transferring a File to an Instrument", on page 219.
- Directly transmit the waveform to an ARB instrument and activate signal generation, see Chapter 3.3.6, "Transmitting the Generated Waveform to the

R&S SMW200A", on page 37, and Chapter 8, "Transferring Data", on page 201.

4.8.6 References

Table 4-15: Import default settings

Parameter	Value
Use Local Server	On
Server Name	localhost
Port Number	1000
Transmission Timeout (s)	20
Filter	Cosine
Roll off factor (cosine filter)	0.10
Impulse Length	Auto
Oversampling	Auto
Clipping State	On

5 Adding Noise to the Signal

This section introduces the concept of the AWGN generator (Additive White Gaussian Noise). It describes the settings for generation of noise, sine wave interferer and adding noise to the generated signal.

5.1 About the AWGN Generator

R&S WinIQSIM2 allows you to superimpose the generated signal with noise. The noise generator generates an AWGN signal (Additive White Gaussian Noise) with selectable bandwidth and adds it to the digital baseband signal. The main characteristic of this kind of noise signal is the Gaussian distribution of the noise power density and uniform frequency distribution.

Multi carrier and multi segment signals cannot be superimposed with a noise signal.

This noise signal superimposes the (interference-free) useful signal ("Additive Noise" mode).

Typical applications for signals superimposed with noise, are bit or block error measurements, depending on the set S / N or C / N ratio, or measurements of mobile base stations.

Generation of the AWGN signal

The generated Gaussian noise achieves ideal statistical characteristics:

- I and Q paths are decorrelated from each other.
- Small probabilities are achieved due to the crest factor of 18 dB.
- The period of the noise signal depends on the selected system bandwidth. The relationship of period P to system bandwidth B_{sys} is approximated:
  $P \approx (2^{800}-1) / B_{sys}$
  – The results are periods between 10^{225} years with minimum bandwidth and approximately 2 days with maximum bandwidth.
  – For comparison: the result corresponds to a period of approximately one month for 3GPP FDD with a bandwidth of 3.84 Mcps. Or, a period of 427 days for GSM with 270.833 ksps.

AWGN modes

The AWGN generator generates signals in one of the following different modes:

- **"Additive Noise":** the generated noise signal superimposes the interference-free useful signal
  An additive white noise is required for measurements of mobile radio base stations.

- **"Noise Only":** a pure noise signal is generated and modulated on the carrier; the connection to the baseband is interrupted.
- **"CW Interferer mode":** a sinusoidal signal with an adjustable frequency offset and carrier-to-interferer (C/I) power ratio is added to the baseband signal by means of a counter instead of a shift register.

Signal and noise parameters

The Figure 5-1 illustrates the relation between the signal and noise parameters.

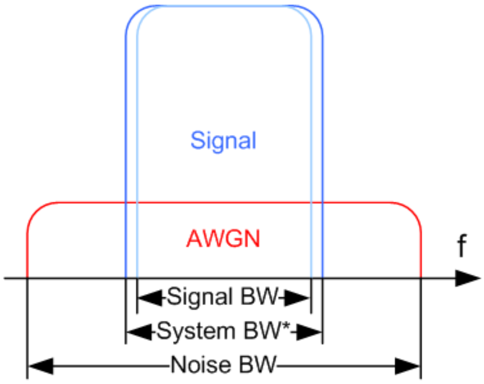


Figure 5-1: Graphical representation of the relation between system bandwidth and noise bandwidth (minimum Noise/System BW = 2)

System BW* = Occupied BW

The **system bandwidth** is a measure for the transmitted RF bandwidth. The selected value is usually the occupied bandwidth and can therefore be a value greater than the pure signal bandwidth.

In the most test cases, the signal and the noise power are not defined directly but by means of the target's signal-to-noise ratio (SNR) or **carrier/noise ratio**.

The **carrier power** is a measure for the *signal without the noise distribution*.

To achieve flat noise within the selected system bandwidth, the **noise bandwidth** must be larger than the system bandwidth. The minimum noise bandwidth is sometimes part of the specifications and is calculated as follows:

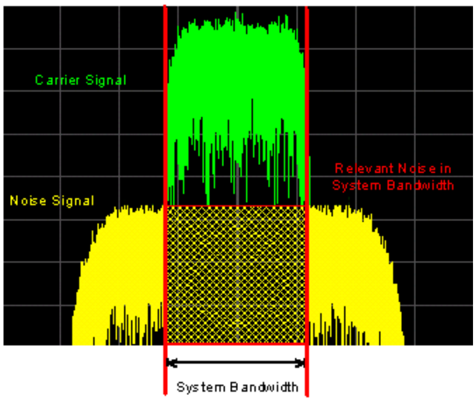
"Noise Bandwidth" = "System Bandwidth" x "Minimum Noise/System Bandwidth Ratio"

The achievable noise bandwidth must not exceed the sampling rate of the baseband.

By default, the **noise power** within the system bandwidth is not defined directly but calculated depending on the selected system bandwidth and the desired SNR. The noise power over the noise bandwidth is calculated respectively.

Measured noise power can exceed the displayed value

The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth. I.e. the total measurable noise power usually exceeds the displayed value "Noise Power".



For a correct measurement of the noise power within the system bandwidth, it is recommended that you measure the channel power with a signal analyzer.

In the "Additive Noise" mode, the output signal is the *signal with the noise distribution*. Hence, the power level at the RF output corresponds to the **carrier+noise power**.

The noise power of a digitally modulated signal is characterized by the parameter $\mathbf{E_b/N_0}$ indicating the ratio of bit energy to noise power density. The correlation to the SNR is as follows:

C/N or S/N = $(E_b/N_0)*(f_{bit}/B_{sys})$, where B_{sys} is the system bandwidth and

Bit Rate f_{bit} = *"Symbol Rate" * Modulation Value*

Where is the bit rate value retrieved from?

The parameter "AWGN > Bit Rate" indicates the value used by the C/N or E_b/N_0 calculation. The value is retrieved automatically depending on the configured baseband signal:

- For signals generated by the "Custom Dig Mod", the bit rate is determined by the selected standard.
  (see parameter "Custom Dig Mod > General" > ""Symbol Rate" on page 69").
- For signals generated in accordance to a digital standard, the bit rate is often a stand-alone parameter.
  Some test cases however, for example the 3GPP base station tests (TS 25.141), specify the E/N settings that apply to channel-coded data or block segments. Set the parameter "Bit Rate" to the required value, e.g. the bit rate before or after channel coding.

Application fields

Typically, the noise generator is required for the following tests:

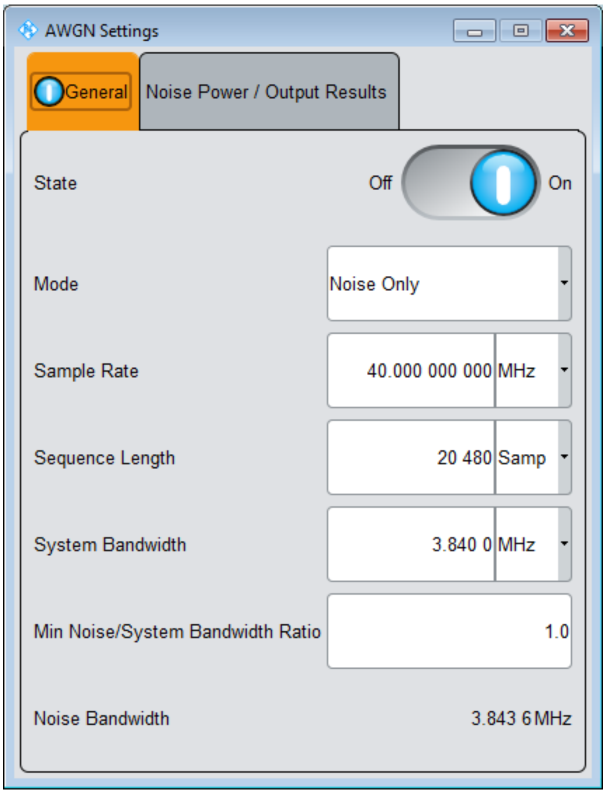
- In receiver sensibility tests with predefined SNR of the receiver
- In bit-error or block-error measurements, depending on the set signal-to-noise ratio (SNR)
- Whenever a pure noise signal is required

5.2 AWGN Settings

The "AWGN" block provides access to the settings for the AWGN generator.

How to access and configure the "AWGN Settings"

1. To access the "AWGN Settings" dialog, select "Block Diagram > AWGN".



   The dialog contains the parameters for configuration of noise and CW interfering signals.

2. To activate the AWGN generator, perform one of the following:
   a) In the AWGN Settings dialog, set "State > On".
   b) In the block diagram , select "AWGN > On".

The remote commands required to define these settings are described in Chapter 12.6, "SOURce:AWGN Subsystem", on page 293.

5.2.1 General Settings

State

Activates the generation of an AWGN signal. The interferer (AWGN or CW interferer, depending on the selected mode) is generated after the generator is activated.

Remote command:
`[:SOURce<hw>]:AWGN:STATe` on page 300

Mode

Selects the mode for generating the interfering signal.

"Additive Noise"
The AWGN noise signal with selectable system bandwidth is added to the baseband signal.

"Noise Only" The pure AWGN noise signal with selectable system bandwidth is modulated to the carrier. The connection to the baseband is interrupted.

"CW Interferer" A sine with a defined frequency offset is added to the baseband signal. The calculation of E_b/N_0 ratio is omitted.

Remote command:
`[:SOURce<hw>]:AWGN:MODE` on page 297

Sample Rate

Sets the sample rate of the AWGN signal in "Noise Only" mode.

The sample rate depends on the selected instrument. If you select a different device, the value range of the sampling rate changes accordingly. In addition, the maximum sample rate must not exceed the maximum clock frequency of the instrument.

Remote command:
`[:SOURce<hw>]:AWGN:SRATe` on page 300

Sequence Length

Sets the sequence length of the "Noise Only" signal in samples.

This parameter depends on the selected instrument. The value range changes accordingly, if you select a different instrument.

Remote command:
`[:SOURce<hw>]:AWGN:SLENgth` on page 300

System Bandwidth

Sets the RF bandwidth for the corresponding carrier/noise ratio of "Additive Noise " and "Noise Only" signals.

Within this frequency range, the signal is superimposed with a noise signal which level corresponds exactly to the set C/N or S/N ratio.

For more information, refer to the description of Figure 5-1, and "Measured noise power can exceed the displayed value" on page 152.

Remote command:
`[:SOURce<hw>]:AWGN:BWIDth` on page 294

Min. Noise/System Bandwidth Ratio
Sets the ratio of minimum noise bandwidth to system bandwidth of "Additive Noise " and "Noise Only" signals. Using this parameter, you can configure the minimum real noise bandwidth, as required by some standards.

R&S WinIQSIM2 automatically adjusts the value range corresponding to the selected instrument. The maximum value of "Min. Noise/System Bandwidth Ratio" is derived from the maximum sample clock of the instrument, according to:

Noise Bandwidth = System BW x Minimum Noise/System BW Ratio

The parameter Noise Bandwidth displays the resulting noise bandwidth. It must not exceed the maximum sampling rate of the instrument.

The calculation of level from the selected C/N or S/N ratio in relation to system bandwidth is not affected

Remote command:
`[:SOURce<hw>]:AWGN:BWIDth:RATio` on page 295

Target CW Frequency Offset
Sets the frequency offset of the sine wave for "CW Interferer" signals.

The range is limited to *± (Symbol Rate + Sample rate) /4*.

Remote command:
`[:SOURce<hw>]:AWGN:FREQuency:TARGet` on page 297

Resulting CW Frequency Offset
Indicates the resulting frequency offset of the sine wave in "CW Interferer" mode.

The resulting "CW Frequency Offset" is the correction of the desired value based on generating an integer multiple of periods out of the output sequence length.

Remote command:
`[:SOURce<hw>]:AWGN:FREQuency:RESult?` on page 296

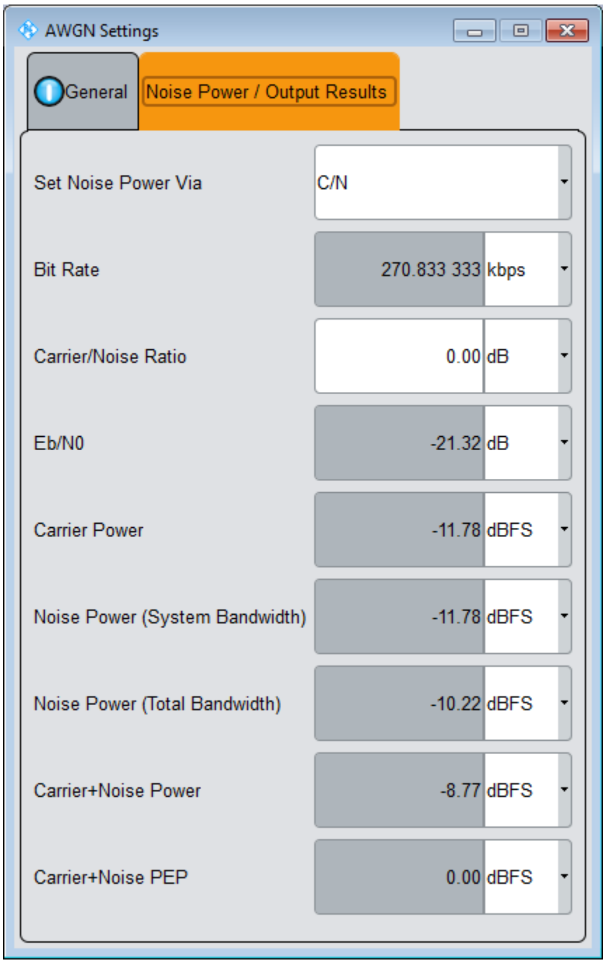
Noise Bandwidth
Indicates the real noise bandwidth of the "Additive Noise" and "Noise Only" signals.

See also Figure 5-1.

Remote command:
`[:SOURce<hw>]:AWGN:BWIDth:NOISe?` on page 295

5.2.2 Noise Power/Output Results Settings

► To access these settings, select "AWGN > Noise Power / Output Results".



The dialog contains the parameters for noise power configuration. The available settings vary according to the selected Mode.

Set Noise Power Via

Selects the mode for setting the noise power of an "Additive Noise" signal.

The following correlation applies:

"C/N or S/N"= ("E_b/N_0")(f_{bit}/B_{sys})*, where:

- "C/N or S/N" is the carrier/noise ratio
- "E_b/N_0" is the ratio of bit energy to noise power density
- *"Bit Rate "f_{bit}= "Symbol Rate" x Modulation Value*
- B_{sys} is the system bandwidth.

See also Chapter 5.1, "About the AWGN Generator", on page 150.

Remote command:

`[:SOURce<hw>]:AWGN:POWer:MODE` on page 298

Bit Rate
Indicates the bit rate used for converting C/N or S/N to E_b/N_0 in "Additive Noise" mode.

For input, set > C/N, or S/N. In "E_b/N_0" mode, the parameter is only displayed.

Tip: For digitally modulated signals, select the bit rate before or after channel coding, as required.

See also "Signal and noise parameters" on page 151.

Remote command:
`[:SOURce<hw>]:AWGN:BRATe` on page 294

Carrier/Noise Ratio, Carrier/Interferer Ratio
Sets the ratio of the carrier signal to the respective AWGN signal:
- "Carrier/Noise Ratio" in "Additive Noise" mode
- "Signal/Noise Ratio" in "CW Interferer" mode

- The power of the noise signal is derived from the entered C/N or S/N value and displayed with the parameter Noise Power (System Bandwidth) / Interferer Power in the units of the useful signal.
- The power of the useful signal is displayed with the parameter Carrier Power and can also be modified there.
- The power of the output signal is displayed under Carrier + Noise Power, Carrier + Interferer Power.

Remote command:
`[:SOURce<hw>]:AWGN:CNRatio` on page 296

E_b/N_0
Specifies the value for the ratio of bit energy to noise power density in "Additive Noise" mode.

For input, set > Eb/N0. In "C/N, S/N" mode, the parameter is only displayed.

- The power of the noise signal is derived from the selected E_b/N_0 and displayed with the parameter Noise Power (System Bandwidth) / Interferer Power in the units of the useful signal.
- The power of the useful signal is displayed with the parameter Carrier Power"".
- The power of the output signal is displayed under Carrier + Noise Power, Carrier + Interferer Power.

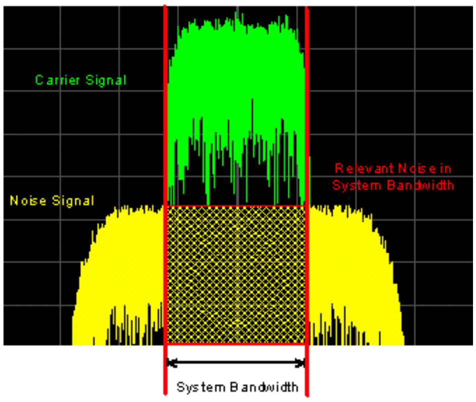
Remote command:
`[:SOURce<hw>]:AWGN:ENRatio` on page 296

Carrier Power
Displays the power of the useful signal in "Additive Noise" or "CW Interferer" mode.

Remote command:
`[:SOURce<hw>]:AWGN:POWer:CARRier` on page 297

Noise Power (System Bandwidth) / Interferer Power
Displays the power of the noise signal in the system bandwidth.

Note: The noise signal is not generated arbitrarily for a particular bandwidth, but instead it is generated in steps. Noise therefore also occurs outside the set system bandwidth. I.e. the total measurable noise power (see below) usually exceeds the value displayed here. For correct measurement of the noise power within the system bandwidth, it is recommended that you measure the channel power with a signal analyzer.



Remote command:
[:SOURce<hw>]:AWGN:POWer:NOISe on page 298

Noise Power (Total Bandwidth)
Displays the power of the noise signal in the total bandwidth in "Noise Only" and "Additive Noise" mode.

Remote command:
[:SOURce<hw>]:AWGN:POWer:NOISe:TOTal? on page 299

Carrier + Noise Power, Carrier + Interferer Power
Displays the power of the noise/interferer signal plus useful signal in "Additive Noise" and "CW Interferer" mode.

Remote command:
[:SOURce<hw>]:AWGN:POWer:SUM? on page 299

Carrier+Noise PEP, Carrier + Interferer PEP
Displays the peak envelope power (PEP) of the signal comprised of noise signal plus useful signal in "Additive Noise" and "CW Interferer" mode.

This parameter is set to 0 dBFS whereas all other display power values show the values relative to the PEP after noise generation.

Note: The peak envelope power ("PEP") displayed in the "Status bar" corresponds to the PEP value of the carrier.

Remote command:
[:SOURce<hw>]:AWGN:POWer:SUM:PEP? on page 299

5.3 How to Configure the Noise Generator for Receiver Tests

This section shows you how to configure the noise generator for receiver tests, for example the tests specified in the 3GPP test specification TS 36.141, chapter 7.

For the following example, we assume that R&S WinIQSIM2 generates an uplink EUTRA/LTE signal with following characteristics:

- "EUTRA/LTE > Duplexing = FDD", "Link Direction = Uplink"
- "EUTRA/LTE > General Uplink Setting > Channel Bandwidth = 5 MHz" and "Occupied Bandwidth = 4.5 MHz"
- "EUTRA/LTE > Frame Configuration > UE1 > FRC > FRC State = On" and "FRC = TS 36.141: A2-3"
- "EUTRA/LTE > State = On"

Settings in the used signal generator:

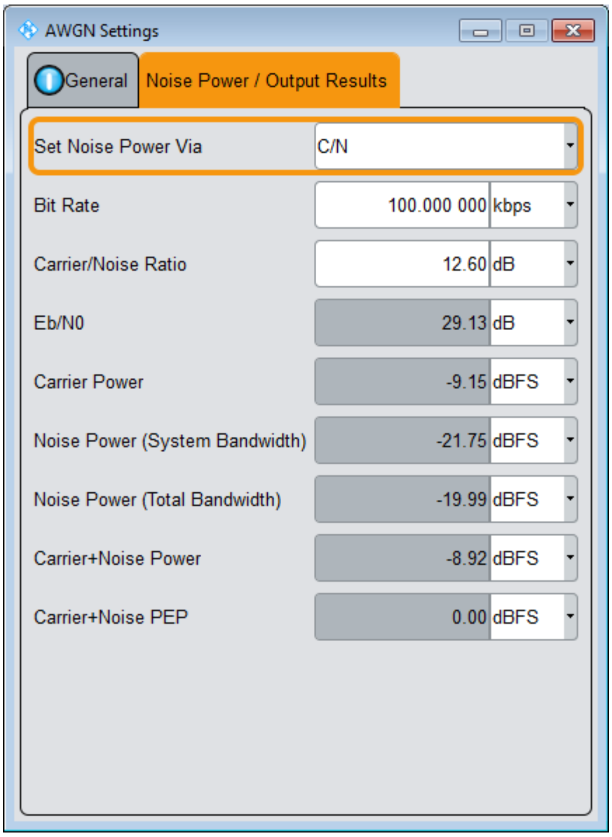
- "Status bar > Frequency = 1.95 GHz" and "Level = -69.9 dBm"
- "RF State > On"

To configure the AWGN for receiver sensibility tests with predefined SNR of the receiver

We assume that an AWGN interfering signal with the following characteristics is required:

Interfering signal mean power = -21.75 dBm/BW and C/N = 12.6 dB

1. Select "AWGN".
2. Set the following parameters:
   - "Mode > Additive Noise"
   - "System Bandwidth > 4.5 MHz"
     The unit of the interfering mean power is dBm/*BW*, where BW is the system bandwidth. The system bandwidth of the LTE signal is the occupied bandwidth, see also Figure 5-1.
   - "Min Noise/System Bandwidth Ratio = 1.5"
3. Select "Noise Power/Output Results".
4. Set the following parameters:
   - "Set Noise Power Via > C/N"
   - "Carrier/Noise Ratio > 12.6 dB"



The dialog confirms the required interfering signal mean power "Noise Power (System Bandwidth) = -21.75 dBm". If you add this value to the level of the generator (-69.9 dBm), the resulting interfering power is 82.5 dBm, as required for testing.

5. Select "AWGN > General > State > On".

   R&S WinIQSIM2 generates the noise signal, and adds it to the baseband signal. Now you can store the waveform, or directly transfer it to a connected instrument, see Chapter 8, "Transferring Data", on page 201.

To generate a CW interfering signal

In the following example, we assume that two interfering signals are required, a 5 MHz uplink LTE signal and a CW signal. The signals are transmitted at -52 dBm in the signal generator; the LTE signal at 1.96 GHz and the CW signal 10 MHz below it.

To configure the signal:

1. To enable the required LTE signal, select "EUTRA/LTE > General Uplink Setting > Channel Bandwidth = 5 MHz" and "EUTRA/LTE > State = On".
2. Enable the required LTE signal:
   a) Select "EUTRA/LTE > General Uplink Setting > Channel Bandwidth = 5 MHz" and "EUTRA/LTE > State = On".
   b) Select "Status bar > Frequency = 1.96 GHz" and "Level = -52 dBm".
3. Enable the required interfering signal:

a) Select "AWGN > General > Mode > CW Interferer".
b) Select "Target CW Frequency Offset = 10 MHz".
c) In the "Interferer Power/ Output Results" tab, select "Carrier/Noise Ratio = 12.6 dB".
d) In the "General" tab, select "State > On".

R&S WinIQSIM2 generates the noise signal, and adds it to the baseband signal. Now you can store the waveform, or directly transfer it to a connected instrument, see Chapter 8, "Transferring Data", on page 201.

6 Displaying Simulated Waveforms Graphically

The built-in graphical signal display function of R&S WinIQSIM2 visualizes a generated signal in various graphical views. Shown graphically, you can quickly check signal characteristics or evaluate the signal in detail.

6.1 About the Graphical Signal Display

R&S WinIQSIM2 displays a signal either as I/Q, amplitude/phase, vector, eye, or constellation diagram, but also the power spectrum of the signal or the statistical evaluation of the signal power, see Diagram Modes. General display functions, like zooming or the use of markers, assist you in the in-depth evaluation of certain measuring points, see Display Functions.

6.1.1 Diagram Modes

This section focuses on graphical signal displays, their content and application. See Table 6-1 for an overview of the signal characteristics you can monitor.

Table 6-1: Graphics modes overview

Graphics modes	Domain	Measurement is known from this instrument
I/Q Diagram	Time	Oscilloscope (standard mode)
R/Phi Diagram	Time	Oscilloscope (standard mode)
Vector Diagram	Time	Oscilloscope (XY mode)
Constellation Diagram	Time	Oscilloscope (XY mode)
Eye Diagram	Time	Oscilloscope (triggered to symbol clock and showing repetitive traces)
CCDF Display	Statistical representation of peaks	Peak Power Analyzer
Power Spectrum	Frequency	Spectrum Analyzer

6.1.1.1 I/Q Diagram

The I/Q diagram displays the inphase component (i[t]) and quadrature component (q[t]) of the I/Q signal over time.

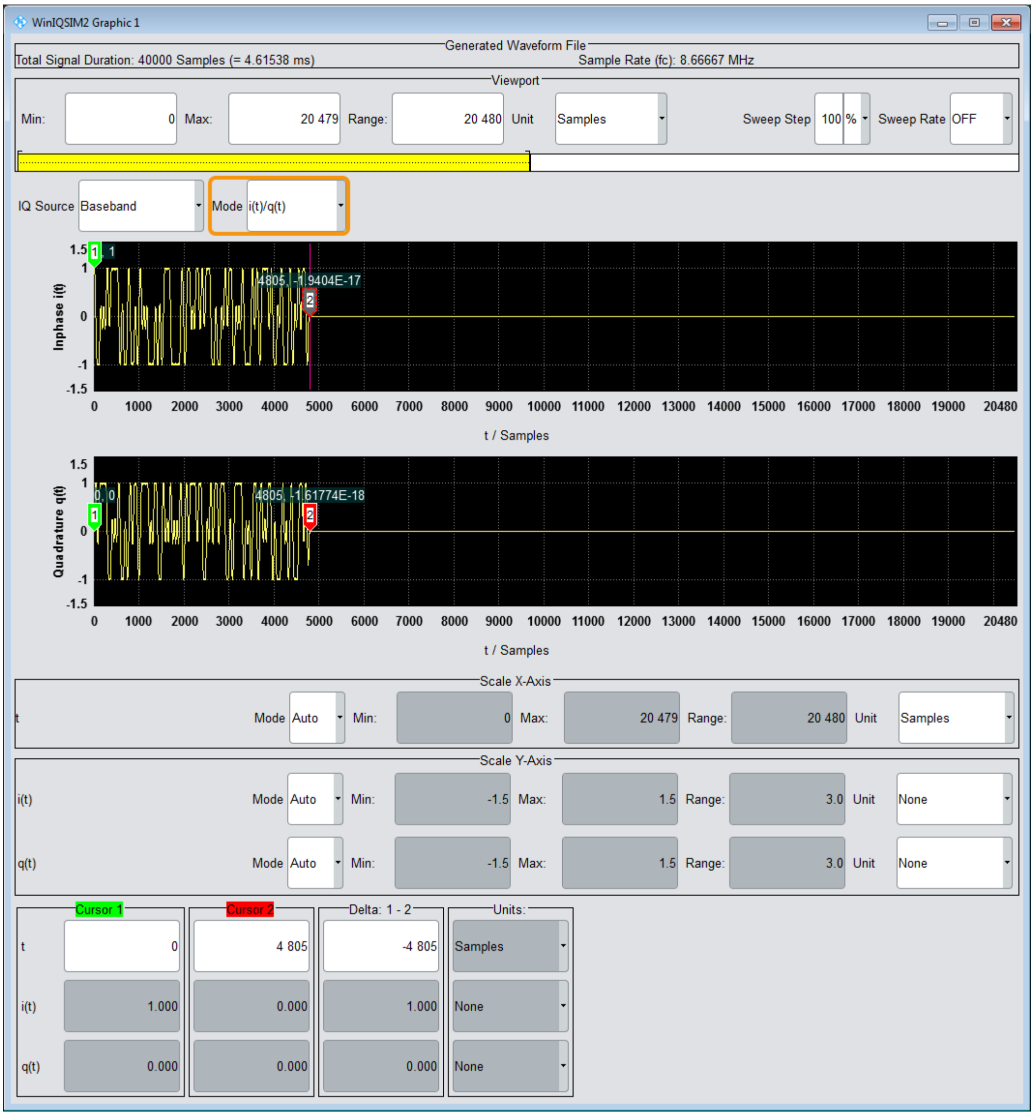


Figure 6-1: I/Q diagram of a GSM/EDGE signal with markers

Inphase (t), quadrature q[t]) = graphical signal display in two separate coordinate systems with identical X and Y axes

X-axis = time represented as number of symbols, chips or samples depending on the signal

Y-axes = amplitude, scaled to the peak envelope power; where minimum scaled amplitude = -1, maximum scaled amplitude = +1

Application: To analyze time slot signals for example with added noise from the AWGN in the time domain. This mode shows the signal like an oscilloscope.

6.1.1.2 R/Phi Diagram

The R/Phi diagram displays the amplitude (r[t]) and phase (phi[t]) of the waveform over time.

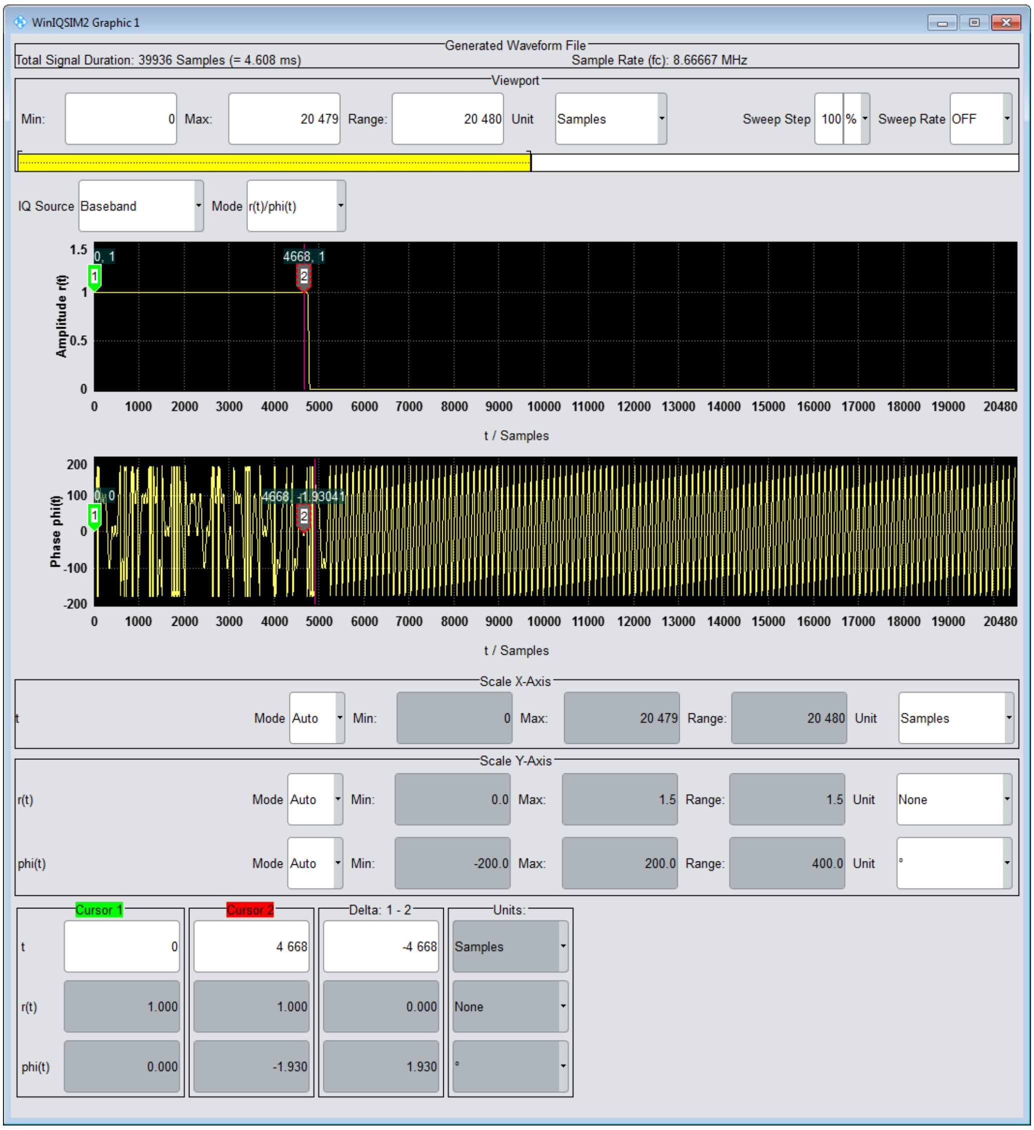


Figure 6-2: R/Phi diagram with markers

Amplitude (r[t]), Phase (phi[t]) = graphical signal display in two separate coordinate systems with identical X axes and varying Y axes

X-axis = time represented as number of symbols, chips or samples depending on the signal

Amplitude (r(t) = absolute amplitude value, scaled to the peak envelope power; where minimum scaled amplitude = -1, maximum scaled amplitude = +1

Phase (phi(t) = phase, scaled in rad or degree; where minimum scaled phase = -180 °, maximum scaled amplitude = +180 °

Application: To analyze the amplitude or phase modulated signals in the time domain.

6.1.1.3 Vector Diagram

The vector diagram displays the Q component over the I component. All points in the complex domain are connected by lines.

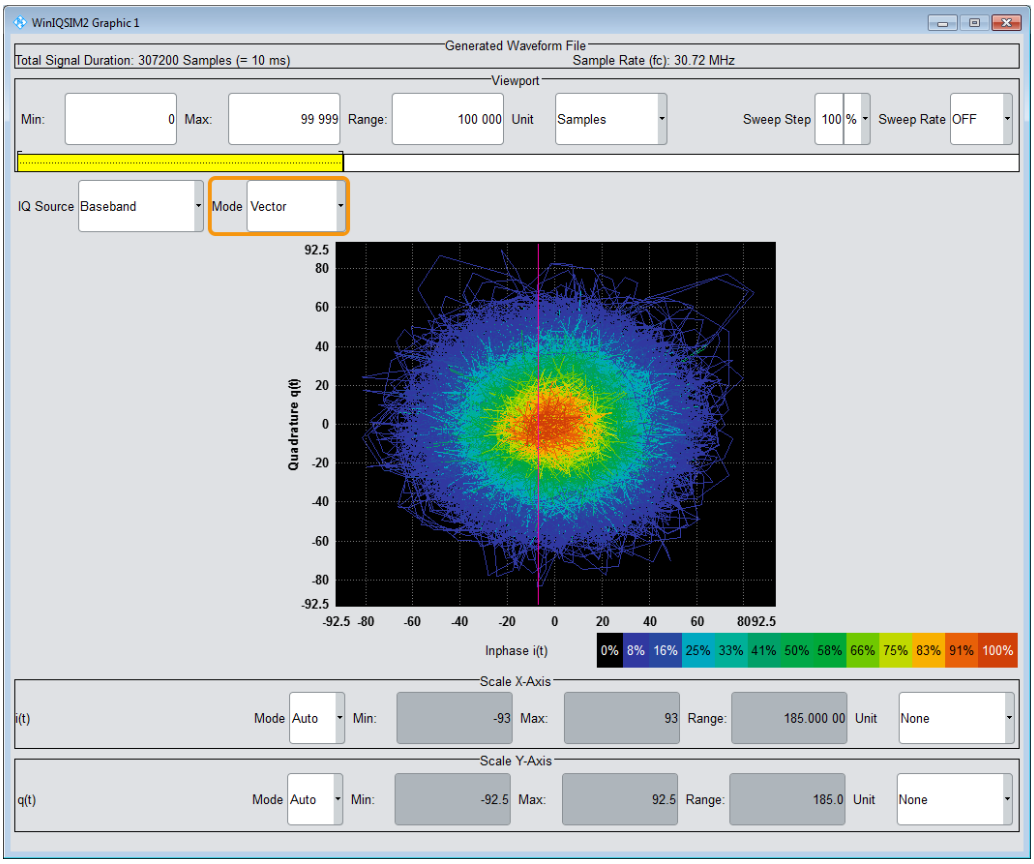


Figure 6-3: Vector diagram

X-axis, Y-axis = amplitudes of the signal components, scaled to the peak envelope power; where minimum scaled amplitude = -1, maximum scaled amplitude = +1

Application: To analyze the transition between the various states of modulation mapping, especially with linear modulations like MSK, QPSK or QAM. This mode shows the signal like an oscilloscope (X/Y-mode).

6.1.1.4 Constellation Diagram

Constellation diagrams display the modulation symbols as discrete points in the I/Q plane. Unlike the vector diagram, the constellation diagram displays only one sample per symbol. This sample represents the symbol.

Constellation diagrams are helpful when generating signals using the "Custom Digital Modulation" settings. Compare the displayed constellation diagram with the diagram displayed in the "Custom Digital Modulation > Modulation" dialog.

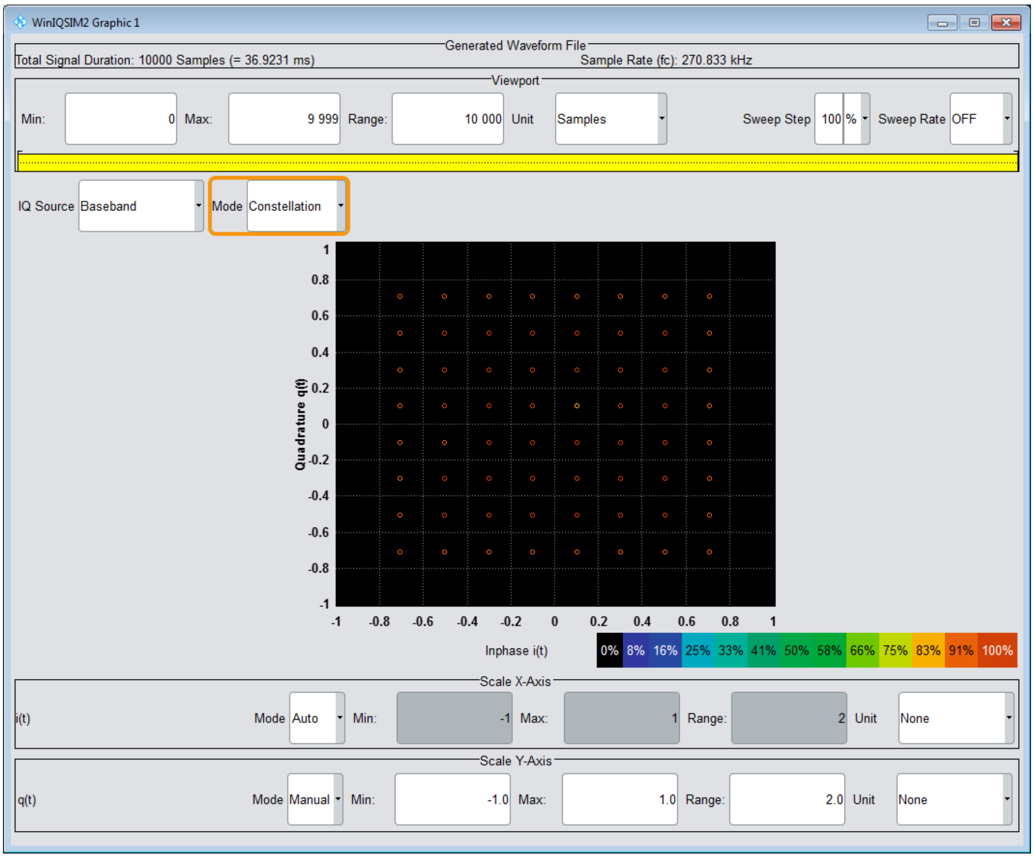


Figure 6-4: Constellation diagram of a WCDMA-3GPP (QPSK 45° Offset) signal

X-axis, Y-axis = amplitudes of the signal components, scaled to the peak envelope power; where minimum scaled amplitude = -1, maximum scaled amplitude = +1

Application: To classify modulation.

6.1.1.5 Eye Diagram

The eye diagram displays synchronized and superimposed sections of either the inphase or the quadrature components of the signal.

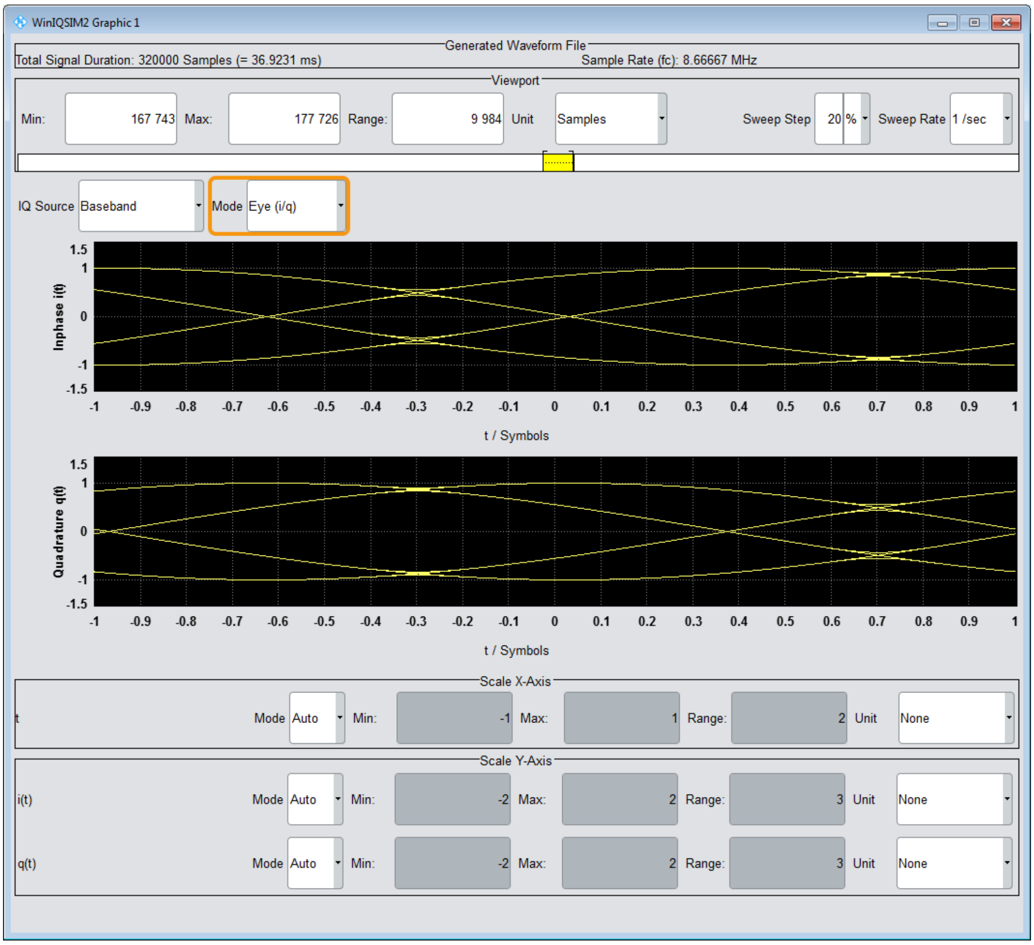


Figure 6-5: Eye diagram with partially closed eye

X-axis = time in the range of -/+ 1 symbol
Y-axis = amplitude, scaled to the peak envelope power; where minimum scaled amplitude = -1, maximum scaled amplitude = +1

The display width is normalized for two symbols:

- The one-symbol "eye opening" in the center of the display.
- 1/2-symbols to the left and right of the center eye for capturing time transitions.

Several 100 curve segments are superimposed. The beginning of the recording is synchronous to the symbol and chip clock pulse.

Application: To analyze amplitude and time distortion elements (e.g. jitter) at high-speed digital data systems. A high quality, unimpaired signal shows a clearly open eye (horizontally and vertically).

6.1.1.6 CCDF Display

The Complementary Cumulative Distribution Function (CCDF) displays the probability the waveform exceeds the average power.

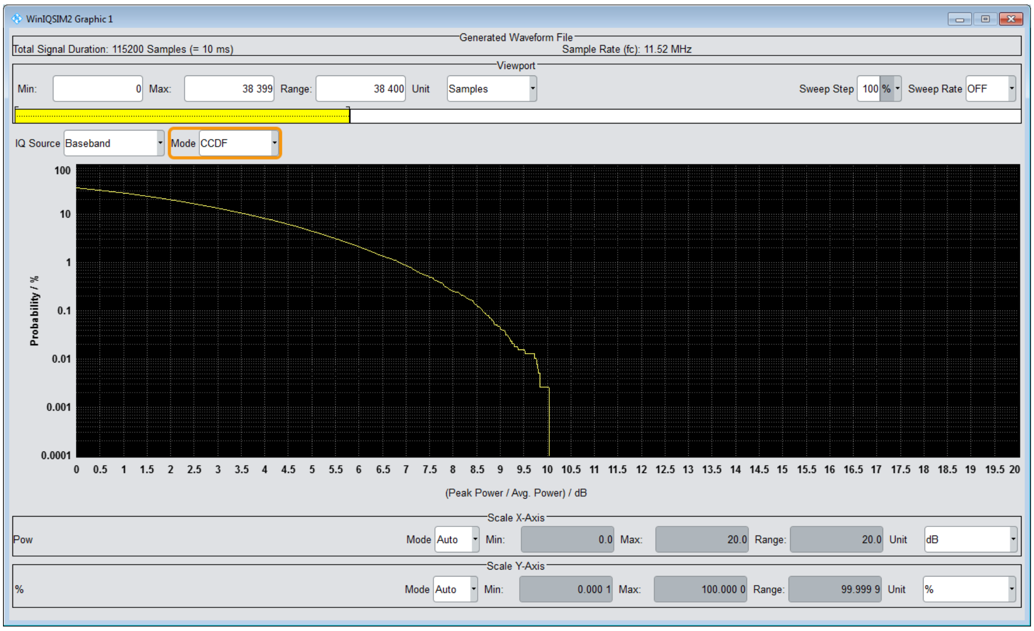


Figure 6-6: CCDF diagram of a 3GPP Signal

X-axis = level over the average signal power, where the average power (RMS) corresponds to the origin
Y-axis = probability of exceeding the average power, that is the percentage of time the signal spends at or above the power level specified by the X-axis

6.1.1.7 Power Spectrum

The power spectrum displays the signal spectrum which is calculated from the I/Q signal by means of Fast Fourier Transform (FFT). The power density over frequency is displayed.

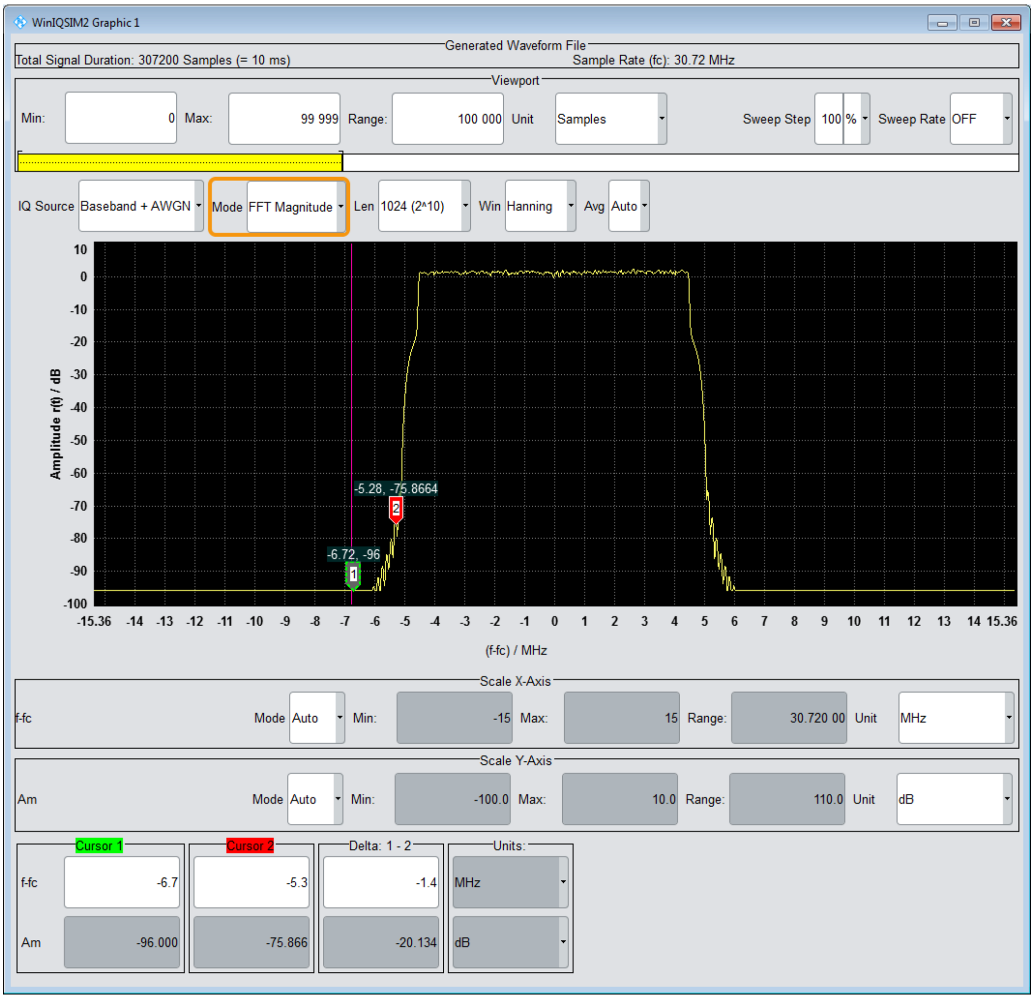


Figure 6-7: Power spectrum diagram: Example with two signals

"Len" = indicates the number of samples used for calculating the Fast Fourier Transform (FFT)
"AVG" = indicates the number of subspectra used for averaging
"Win" = indicates the window FFT function
X-axis = frequency plotted symmetrically (- Sampling Rate/2 to +Sampling Rate/2)
Y-axis = power density

6.1.2 Display Functions

The graphical display provides general display functions known from other measurement instruments. It enables you to focus on a certain area of the diagram or to use markers for precise evaluation. You can find out how to optimize the display for your monitoring task.

Zooming

You can zoom into the diagram to visualize the measurement in more detail. Using the mouse pointer or by value entries you can define the area to be zoomed.

Zooming is merely a visual tool, it does not change the number of samples used for the calculation. Increase the number of samples before zooming, as otherwise the function has no real effect.

A zoom replaces the current diagram by a new diagram which displays an enlarged extract of the diagram. Repeat zooming until the required details are visible.

Using markers

Markers are tools for numerical readout of measured data in diagrams. They help you to analyze the displayed signal characteristics by determining particular values in the diagram. Thus you can extract numeric values from a graphical display in both the time and the frequency domain. For example, you can measure the frequency distance between two peaks of a signal.

See also .

You can use two markers (and) for your measurements. With the mouse pointer or on the basis value entries, you can determine the marker positions and the distance between "Marker 1" and "Marker 2". Directly above the icons, you can see the corresponding values of the marker in the diagram.

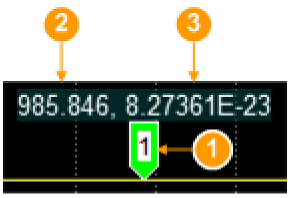


Figure 6-8:

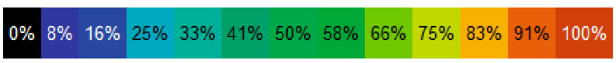
1 = marker number
2 = marker position on the x-axis in the current units
3 = marker value on the y-axis

If a marker is outside the range of the "Viewport", and thus not visible, you can shift it into the indicated area. Type a value within the displayed scale in the entry field of the corresponding marker.

Color maps

The vector, constellation and power spectrum diagrams assign colors to the graphs to visualize how often the signal components have achieved certain values. The color represents the number of occurrences in %, related to the maximum value.

The legend below the diagram explains the distribution the colors represent.



6.2 Graphical Signal Display Settings

This section focuses on the configuration and the graphical representation of a waveform.

If you have activated a graphical representation, R&S WinIQSIM2 provides the graphic as:

- Graphics icon
- Preview in the block diagram
- Full-size graphs in the configuration dialog

To show the small graphics preview in the block diagram, perform one of the following:

1. In the menu bar, select "Graphics > Graphic 1 > Graphic 1 (Preview Only)"
2. In the "Toolbar", select a graphic icon .

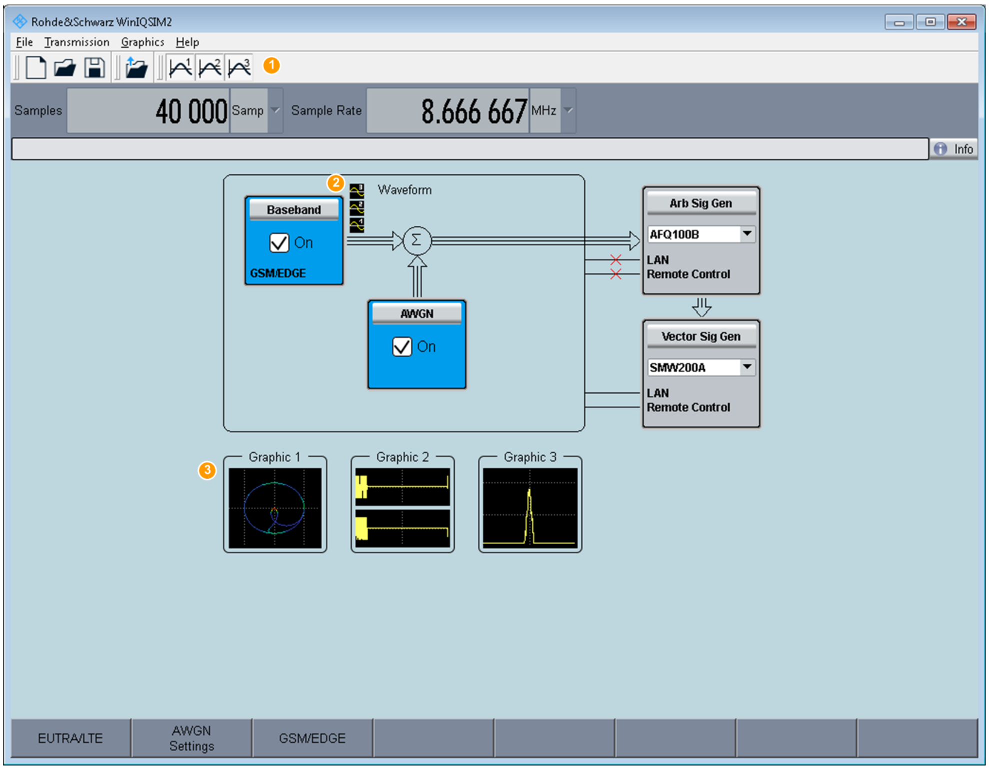


Figure 6-9: R&S WinIQSIM2 bock diagram with active graphics

1 = Toolbar icon
2 = Graphics icon in the block diagram
3 = Small graphics preview

R&S WinIQSIM2 displays the selected graphic as small graphic block diagram.

To access the graphic signal display settings, perform one of the following:

1. In the menu bar, select "Graphics > Graphic 1 > Graphic 1 (Complete)"

2. In the block diagram, double-click the small graphics preview.

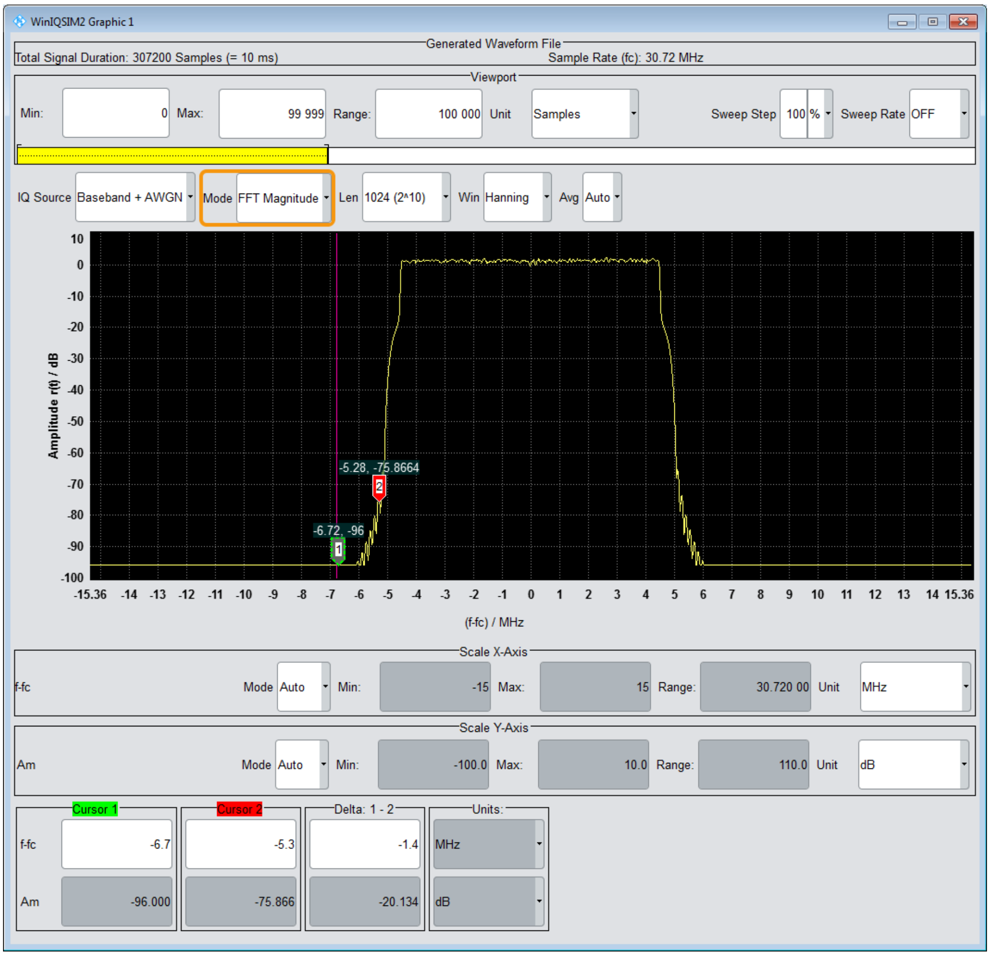


Figure 6-10: Graphics configuration dialog

The dialog contains all parameters required for graphical representation of the waveform:
- Generated Waveform File
- Viewport
- Chapter 6.5, "Scaling and Marker Settings", on page 175.

The remote commands required to define these settings are described in Chapter 12, "Remote Control Commands", on page 251.

6.3 Generated Waveform File

This section shows information on the generated waveform such as the total duration, the sample rate and the oversampling factor.

Total Signal Duration

Displays the signal duration in seconds of the available samples within the generated waveform file.

Remote command:
`:GENerate:WAVeform:DURation?` on page 287

Sample Rate (fc)

Displays the Sample Rate in [Hz]. It is calculated as follows:

Sampling Rate = Number of Symbols * Oversampling Factor.

Remote command:
`:GENerate:WAVeform:SAMPles?` on page 288
`:GENerate:WAVeform:SRATe?` on page 288

Oversampling Factor

Displays the baseband oversampling factor.

This parameter is determined by the content of the generated waveform file. To adjust the oversampling rate, set the corresponding parameters defined within the standards.

Remote command:
`:GENerate:WAVeform:OSAMpling?` on page 287

6.4 Viewport

The viewport indicates the currently displayed range of the waveform.

The yellow bar indicates the displayed part, and the white bar represents the total length of the waveform. The display is dynamic, and you can control it either manually or automatically.

To control the viewport, you have the various options:

- Manually by defining "Min" and "Max" or "Min"or "Max" and "Range"
- Manually with the mouse by dragging the yellow bar
- Using an automatic time control by defining the sweep step size and time

"Mode"	"Viewport"	Yellow Bar	White Bar
"I(t)/q(t)" "r(t)/q(t)"	Range, within the scaling of the X-axis is shifted in the display.	Currently displayed range.	Total signal
"FTT"	Maximum observed range.	Used range for FTT	Total signal
"Vector" "Constellation" "Eye" "CCDF"	Observed scope	Irrelevant	Total signal

In addition, the viewport section includes the selection of the graphics mode, and the possibility to display the signal with noise specified under AWGN (see Chapter 5.2, "AWGN Settings", on page 153.

Depending on the selected graphic mode, you can set additional parameters like filter, segment length.

Viewport
Note: There is no limitation to the length of the waveforms. Long waveform streams are divided in packets of a displayable size and are then displayed packet-wise.

Sets the operating range to display the waveform.

"Min"	Start value of the operating range.
"Max"	End value of the operating range.
"Range"	Operating range, the calculated delta between "Min" and "Max". **Tip:** If you define two of the three values, the third value is calculated automatically.
"Unit"	Unit of the operating range. If you change the unit, the settings are adjusted automatically.
"Sweep Step"	Step size of the sweep in per cent.
"Sweep Rate"	Sweep steps per second.

IQ Source
Selects the signal for the graphical display.

The graphics allow you to view the baseband signal and the additive noise separately, as well as the resulting superimposed signal.

Mode
Selects the graphics mode.

"I(t) / q(t)"	Displays the currently active waveform over time in two graphs: the top one shows the I component (inphase I (t)), the bottom one the quadrature samples (q(t)).
"r(t) / phi(t)"	Displays the waveform over time in two graphs: the top one shows the amplitude, the bottom one the phase.
"Vector"	Depicts the samples in the complex plane, that means the transitions are shown as vectors.
"CCDF"	Depicts the probability the waveform takes on a value higher than the maximum output power.
"FFT Magnitude"	Displays the power spectrum (magnitude over frequency) derived from a Fast Fourier Transform of the waveform.
"Constellation"	Depicts all samples in a two-dimensional diagram in the complex plane, sampled at the symbol rate.
"Eye (I/q)"	Displays the repetitively superposed I and Q samples. The symbol transitions are sampled at the symbol rate.

Len
Selects the number of samples of the Fast Fourier Transform.

Win
Selects the FFT window function.

Windowing supports you in FFT analysis to minimize the discontinuities in the measured signal interval. It reduces the effect of spectral leakage, and the frequency resolution increases.

Various window functions are provided for the FFT analysis of waveform signals. Each of these functions has certain characteristics with specific advantages and also trade-offs. Therefore consider these properties carefully, to find the optimal FFT function for your signal.

"Rect (Rectangular)"	Suitable for separate two tone signals with almost equal amplitudes and a small frequency distance. Resolution: high (frequency), low (amplitude)
"Hanning"	Used for frequency response measurements, sine waves, periodic signals and narrow-band noise. Resolution: high (frequency), low (amplitude)
"Blackman"	Mainly for signals with single frequencies to detect harmonics. Suitable for signals provided for accurate single-tone measurements. Resolution: low (frequency), high (amplitude)
"Welch"	Fast estimation of the power spectrum at different frequencies. Reduces noise in the estimated power spectrum, but also the frequency resolution. Resolution: low (frequency), high (amplitude)
"Flat Top"	Suitable for signals that are used for accurate single-tone measurements. Resolution: low (frequency), high (amplitude)

Avg
Selects the number of waveforms used for average calculation.

6.5 Scaling and Marker Settings

In this section, you can configure the graphics display and determine the marker positions manually.

The settings for configuring the x-axis and y-axis are the same. You can select the manual mode, which enables manual defining of the scope (min, max, range values) and the unit for the respective parameter

Mode
Selects the mode for setting the axis scaling.

"Auto"	Activates automatic scaling. R&S WinIQSIM2 scales the axis automatically, according to the parameters of the waveform. The value range is indicated as read only, but you can still determine the unit. Automatic scaling assumes the values of the viewport for the x-axis, and calculates the scaling for the y-axis on the basis of the maximum and minimum values of the samples.
"Manual"	Activates manual scaling. You can define the area that is to be zoomed with "Min" and "Max", or "Range" in combination with "Min" or "Max". The third value is calculated automatically. The unit can be set as well.

Remote command:
n.a.

Scale X-Axis /Y-Axis
Sets the scaling and the units for the x-axis and the y-axis.

Note: The scaling values can never be higher or lower than the scaling values set under "Viewport" on page 174.

"Min"	Start value of the operating range.
"Max"	End value of the operating range.
"Range"	Operating range, the calculated delta between "Min" and "Max". **Tip:** If you define two of the three values, the third value is calculated automatically.
"Unit"	Unit of the operating range. If you change the unit, the settings are adjusted automatically.

Marker 1 / 2, Delta 1-2, Units
Determines the marker position on the x-axis, and shows the corresponding y-value, as well as the distance between the two markers. See also "Using markers" on page 170.

Zoom Out
Resets a previous zoom, see also "Zooming" on page 169.

Note: The "Zoom Out" button always returns to the initial display size.

6.6 How to Verify the Generated Signal with the Graphics Display

This section shows how to use the various graphic modes to visualize the signal characteristics of the generated waveform. It also explains how to evaluate the effect of standard settings like added noise. The examples use a simple custom digital modulated signal.

Generating a simple WCDMA-3GPP (QPSK 45° Offset) waveform

► In the custom digital modulation dialog, enable a predefined WCDMA-3GPP signal (see "To generate a digitally modulated signal" on page 84).

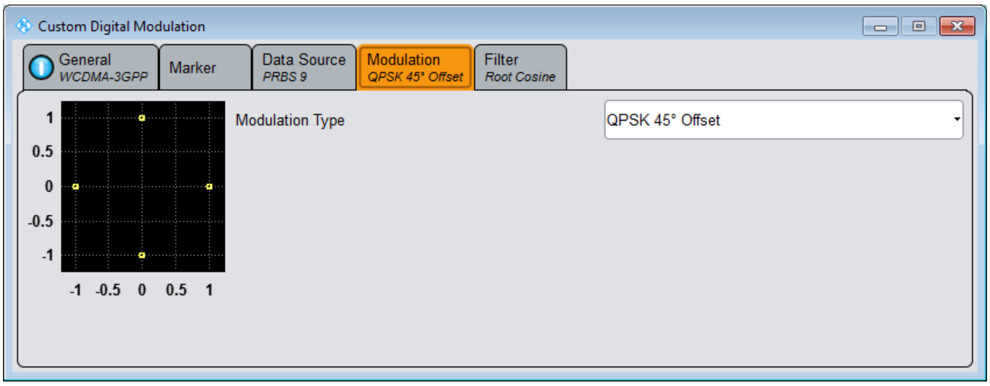
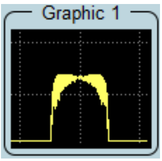


Figure 6-11: Used modulation type

To display the constellation diagram of the WCDMA-3GPP waveform graphically

1. To activate the graphical signal display, perform one of the following:
   a) In the menu bar, select "Graphics > Graphic 1 > Graphic 1 (Preview only)"

   R&S WinIQSIM2 indicates the graphics preview in the block diagram. To access the graphics settings, select the small graphics preview.

   b) In the menu bar, select "Graphics > Graphic 1 (Complete)"

Figure 6-12: Graphic dialog with WCDMA-3GPP signal, I(t)/q(t) mode

The "Graphics Configuration" dialog contains the parameters, required to configure the graphical display, and shows the waveform graphically.

2. Select "IQ Source > Baseband".
3. Select "Mode > Constellation".

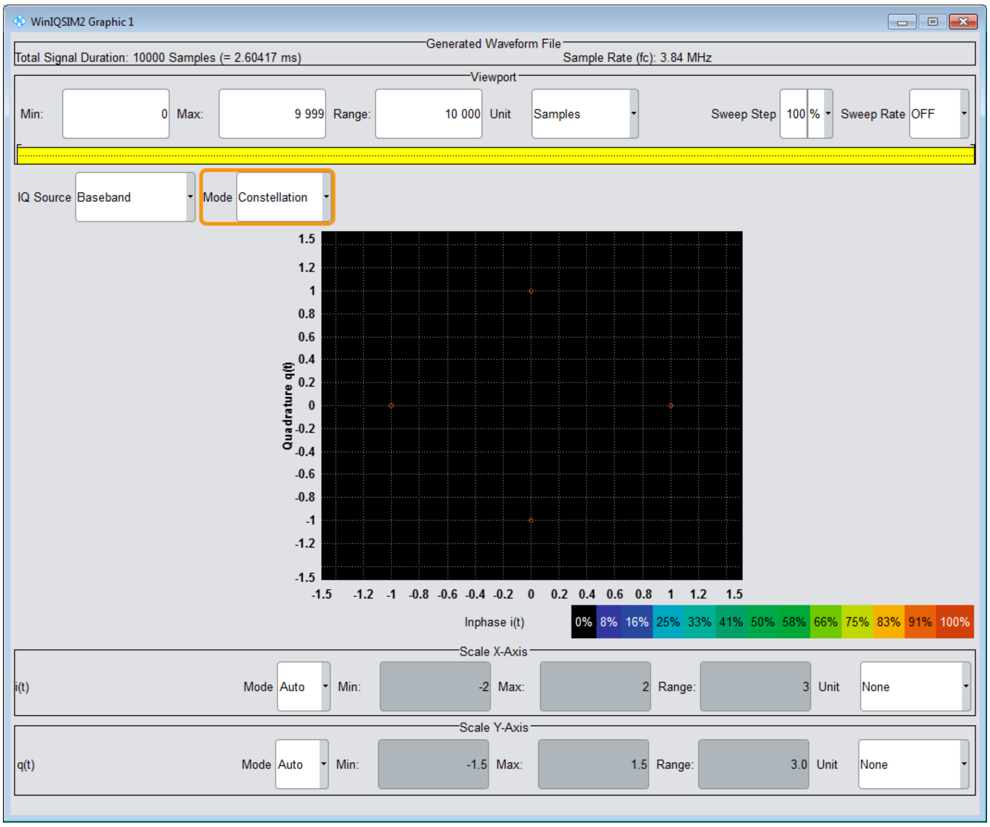


Figure 6-13: Constellation diagram of a WCDMA-3GPP (QPSK 45° Offset) waveform

The displayed constellation diagram confirms the used modulation type, see Figure 6-11.

To display the WCDMA-3GPP waveform in the vector diagram

1. Select "IQ Source > Baseband".
2. Select "Mode > Vector".

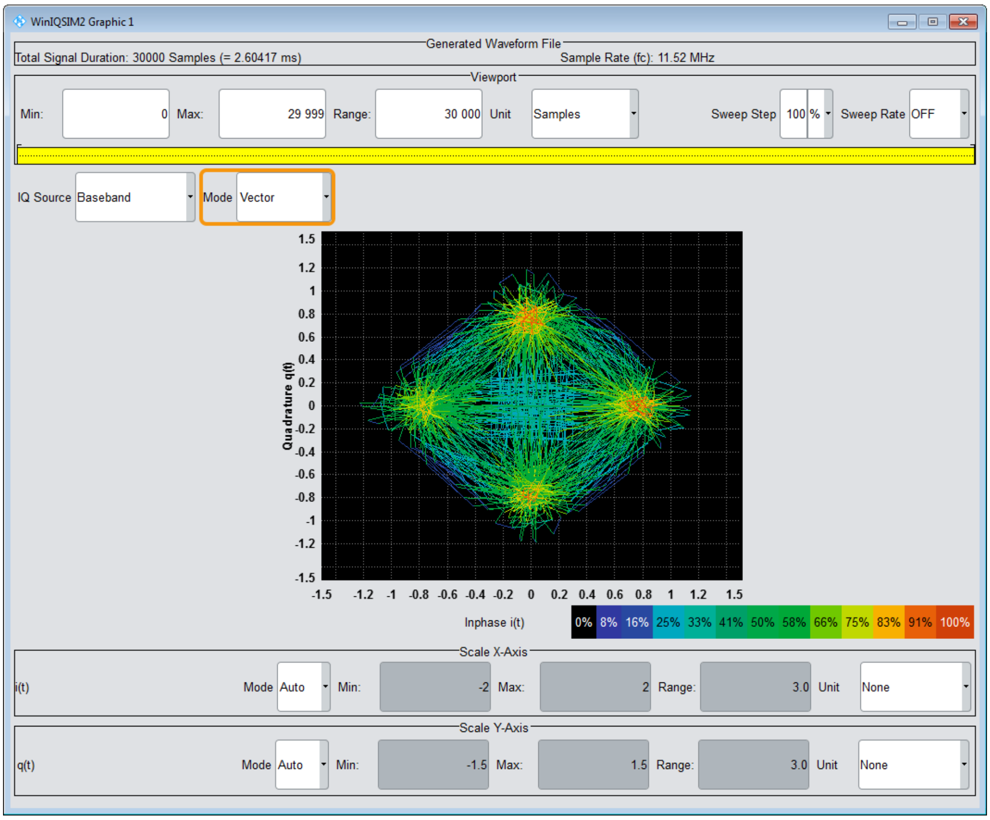
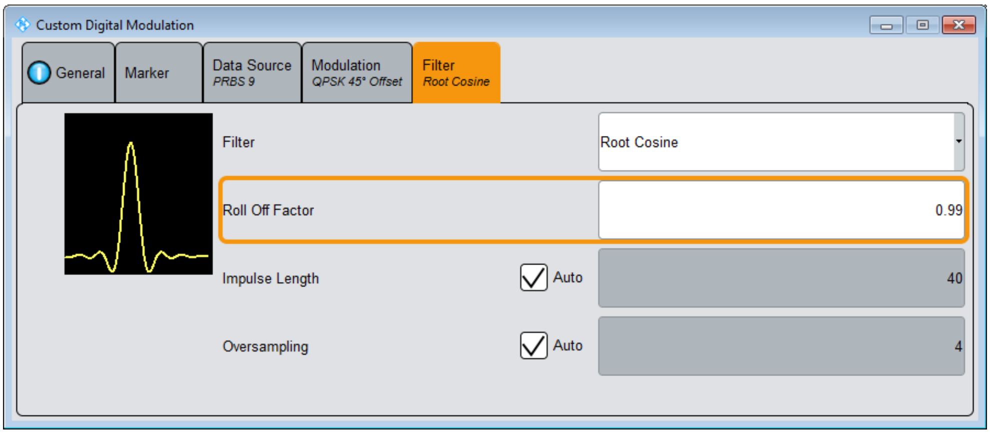
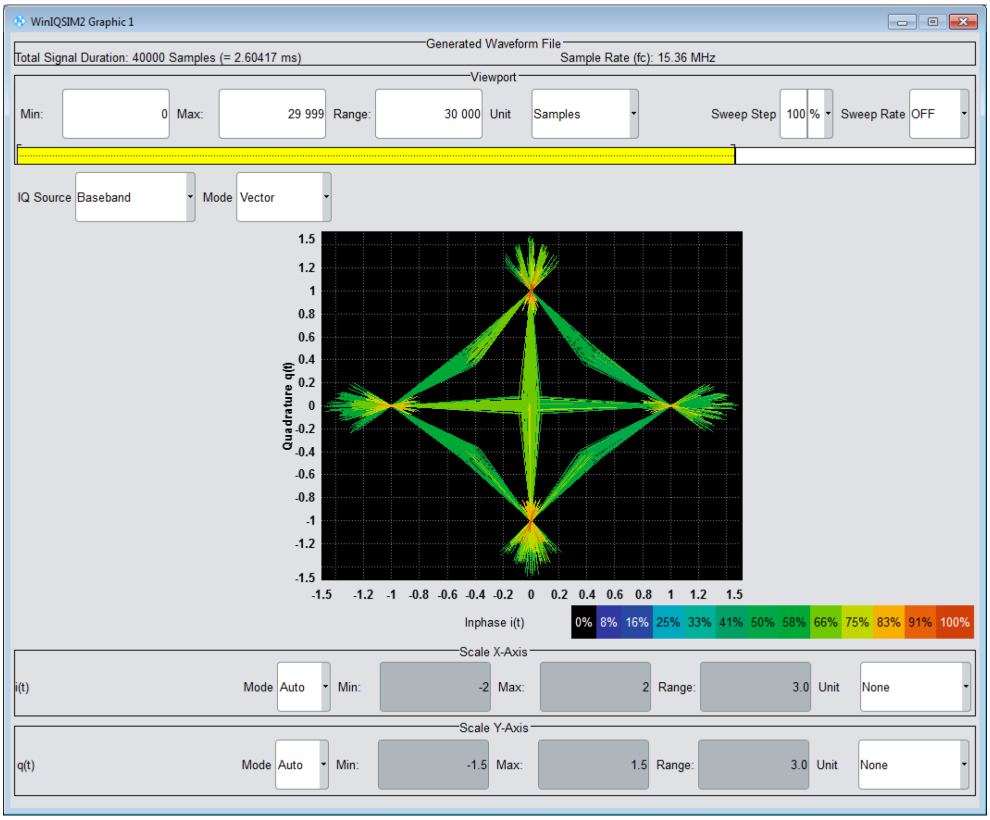


Figure 6-14: Vector diagram of the WCDMA-3GPP (QPSK 45° Offset) waveform

To visualize the effect of filter parameters on the generated waveform

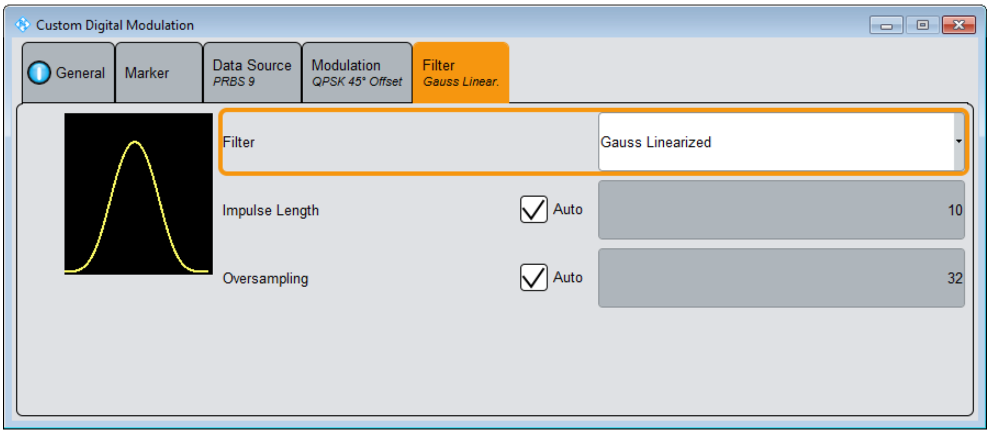
1. Select "Baseband > Custom Digital Mod... > Filter > Roll Off Factor = 0.99"

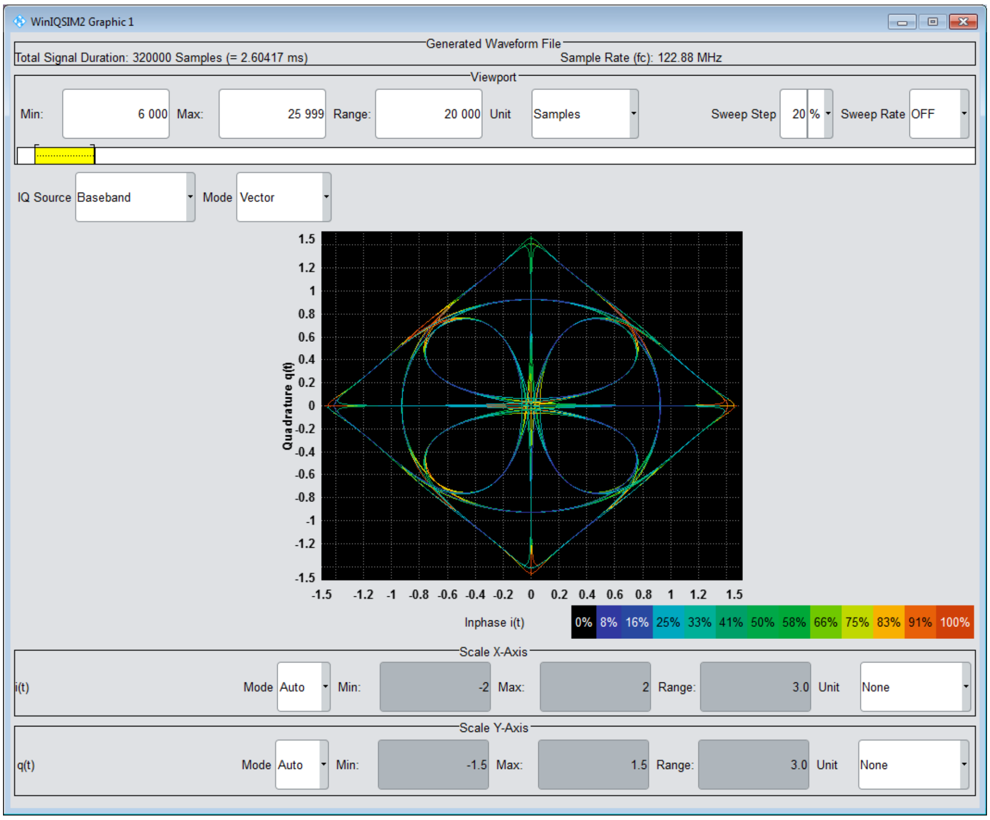




The vector diagram changes according to the modified filter parameter.

2. Select "Baseband > Custom Digital Mod... > Filter > GAUSS linearized"

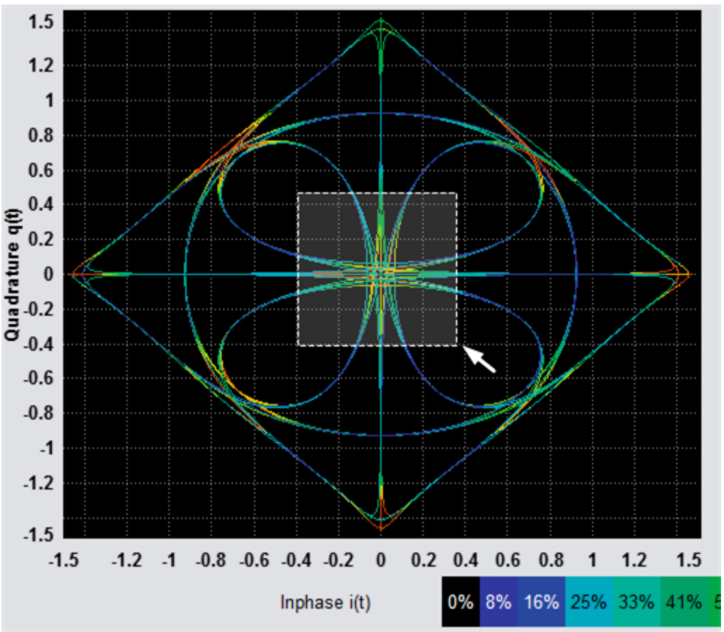




To zoom in a diagram with the mouse

To zoom a displayed part of a waveform directly in the graph, perform the following:

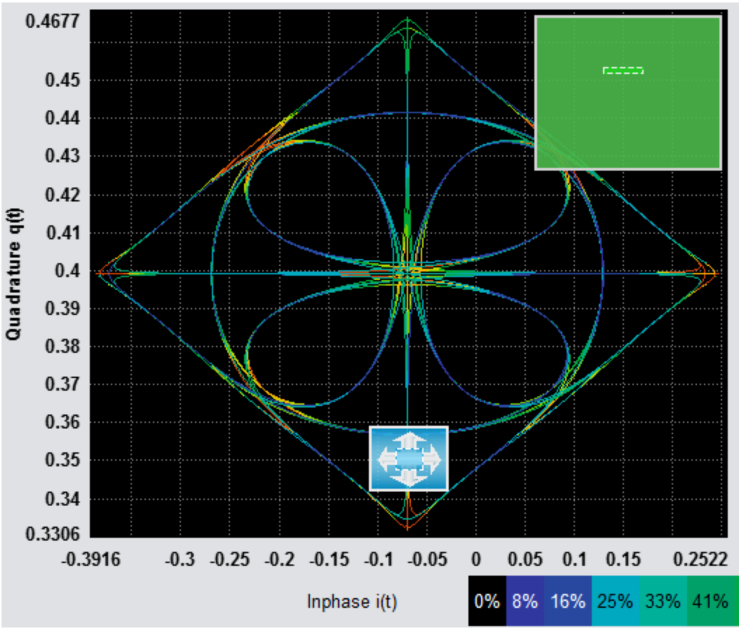
1. Left-click the upper left corner of the graph area you want to zoom.
2. Drag the zoom rectangle.



A dotted rectangular frame shows the marked area.

3. Click into the frame to confirm the selection.

The display expands and shows the selected area. A green frame with a miniaturized rectangle embedded appears, indicating the zoomed area.



Now you can move the zoomed area in the overview-window by dragging with the mouse.
To return to original size, right-click and select "Unzoom" in the context-sensitive menu.

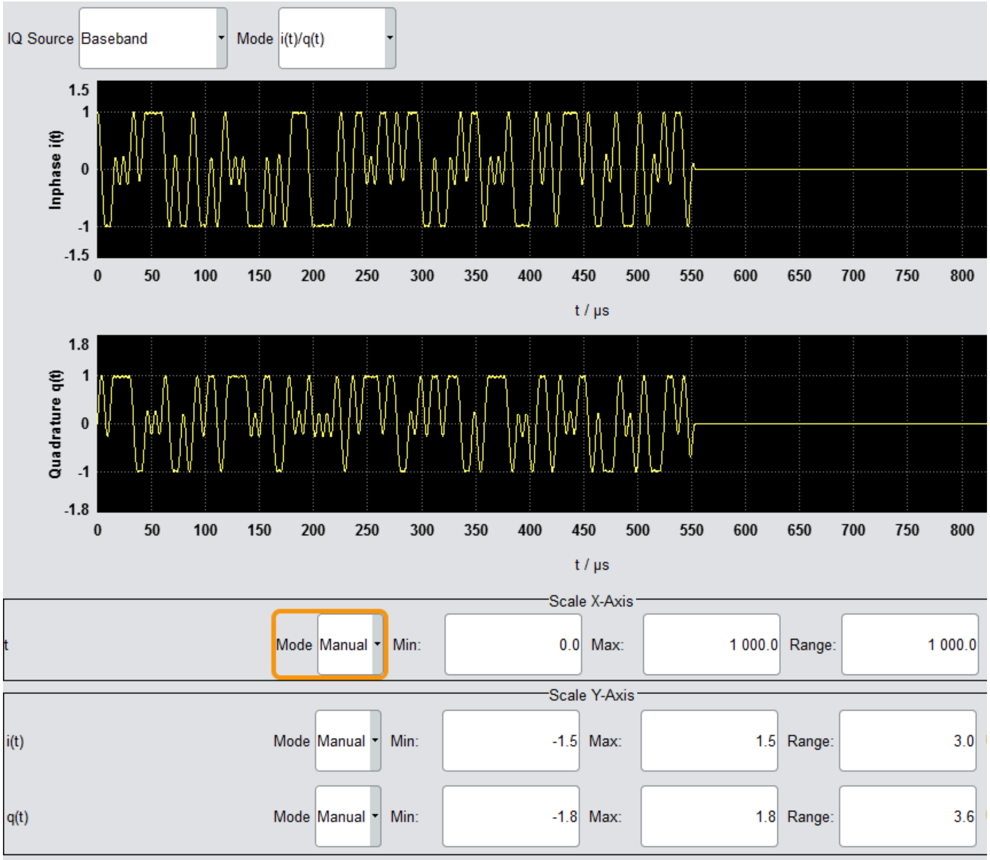
T zoom in a diagram by adjusting the axes

To expand the entire graph, you can also adjust the scaling of the axes. The method for adjusting the axis scale is the same for both the X and Y axis. Therefore, the function is explained only once.

Perform the following:

1. Select "Mode > Manual".

2. Set the "Min" and "Max" values to determine the zoom area.



R&S WinIQSIM2 adjusts the "Range" accordingly. Alternatively, you can determine "Min" and "Range" with an automatically calculated "Max" value.

Tip: The scaling range is limited by the scaling values of the "Viewport".

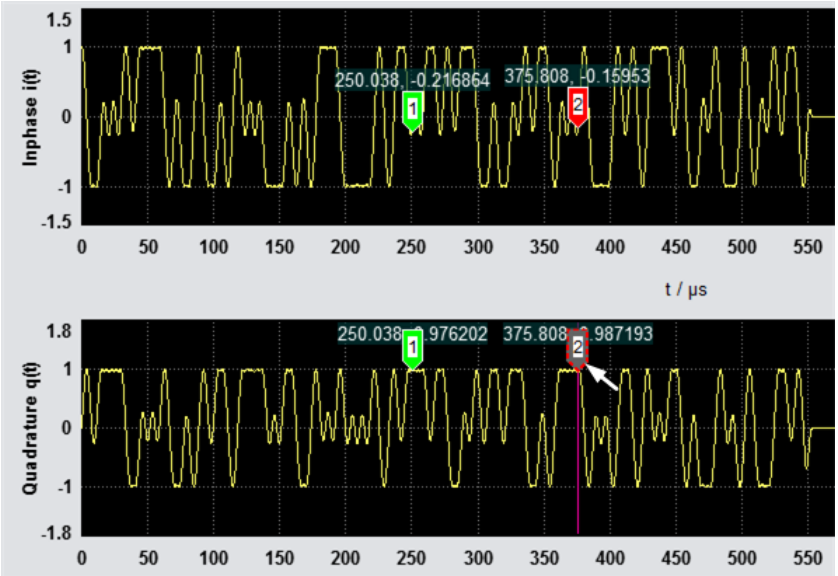
3. If necessary, set the "Unit".
   R&S WinIQSIM2 converts the values automatically.

   For reverse operation, set "Mode > Auto".

To set markers with the mouse

To place a marker directly in the graph, perform the following:

1. Left-click the first marker symbol.
2. Drag and drop the marker to the target position.



3. Left-click the second marker and repeat step 1 to 2.

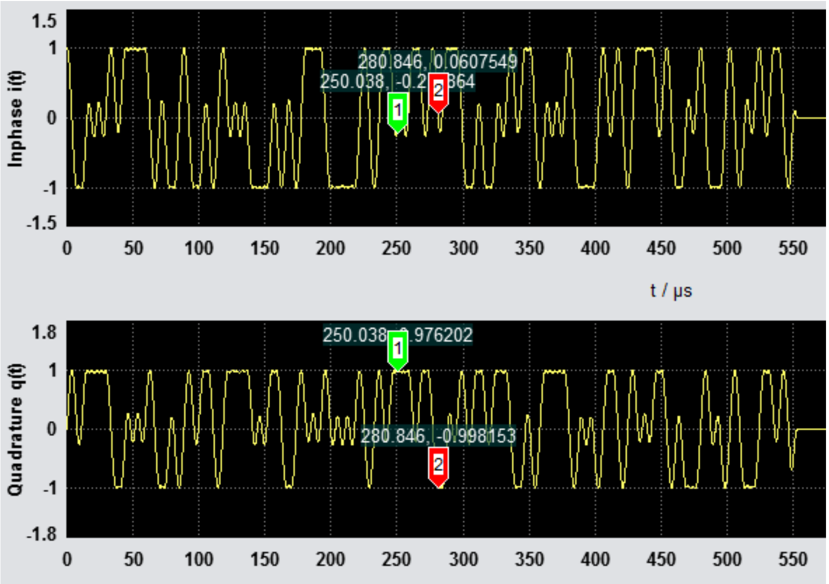
   Now you can analyze the displayed waveform.

Setting markers by entering the position

To place a marker precisely in the graph, perform the following:

► Enter the X-axis positions for the "Marker 1" and "Marker 2".

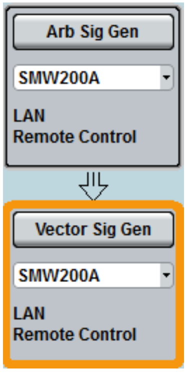
The marker moves to the defined position, giving the Y-axis values in the "I(t)" and "q(t)" fields. The values are not editable. In the "Delta: 0 - X" group box, the difference (delta) between the X-axis value and Y-axis value of the marker position is displayed.



Alternatively, you can determine "Delta: 1 to 2" value with an automatically adjusted "Marker 2" position. "Marker 1" remains fixed.
Now you can analyze the waveform.

7 Setting Up Instruments

The signal generator blocks in the main application window symbolize instruments that can generate signals from the waveforms of R&S WinIQSIM2.



- "Arb Sig Gen" represents instruments featuring an ARB modulation generator.
- "Vector Sig Gen" represents vector signal generators with an I/Q modulator and RF generator.

Both blocks provide access to dialogs for configuring the corresponding destination instrument, the connection between the instrument and R&S WinIQSIM2, and remote operation.

When connected, R&S WinIQSIM2 can send the generated waveform signal directly to a Rohde&Schwarz instrument, and even enable signal generation on the instrument remotely.

Required equipment for instruments

Processing waveform files generated by R&S WinIQSIM2, an instrument must be equipped with an ARB and the corresponding options for the digital standard, multi carrier signal generation and AWGN.

- See the list of instruments Chapter 1.2, "Related Rohde & Schwarz Instruments", on page 13 for information on instruments from Rohde & Schwarz.
- For information on the available options, see the data sheet of the instrument.
- The appropriate options are specified in the description of the respective standard.

Note also, that R&S WinIQSIM2 does not necessarily need a connected instrument for the signal calculation. You can generate the waveform for a manually configured dummy instrument (in offline mode), and subsequently select an instrument in the network for transmission (online mode). However, keep in mind, that the general instrument settings affect various functions, as for example the maximum size of the file the waveform is stored in.

Therefore, it is important that you consider the value ranges of the instrument, which will later generate the signal.

The following sections describe the basic steps to be taken and the necessary prerequisites.

This sections describes the basic steps to be taken and the associated requirements.

7.1 Available Instruments Settings

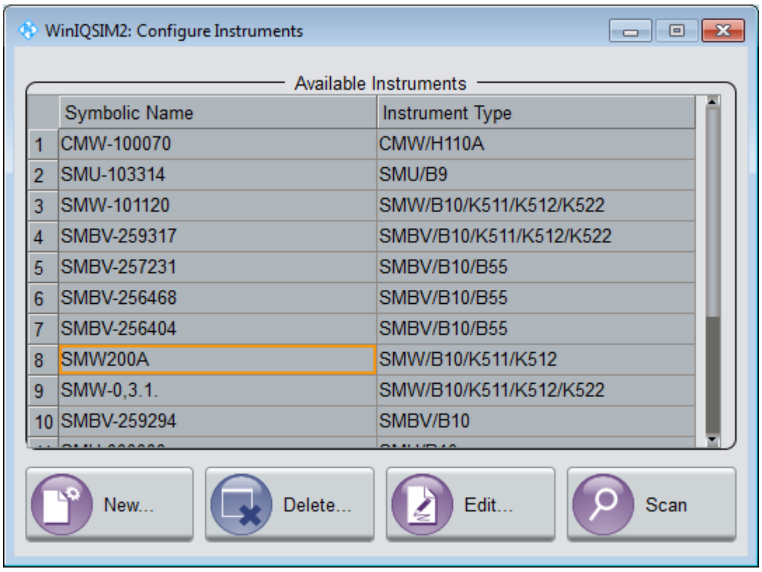
To access an instrument directly, you have the options to scan the network, or USB or GPIB interfaces for connected instruments, as described in "To scan and configure an instrument" on page 24. As an alternative, you can specify an instrument manually. It can be a dummy or an existing instrument, that has not been detected during a scan. To perform these tasks, see Chapter 7.2, "Configure Instruments Settings", on page 189.

The list of instruments shows all previously assigned entries. When you start a network search (scan), or add a new instrument, R&S WinIQSIM2 maintains the list. It adds any newly detected instruments to the next free list entry (instrument number).

► To access the list of available instruments, perform one of the following:

a) In the block diagram, select "ARB Sig Gen, Vector Sig Gen > Instruments"

b) In the menu bar, select "Transmission > Instruments".



The dialog lists all available instruments for selection and provides access to the parameters for configuring a selected instrument. You can also browse the network for instruments, or remove an entry from the list. With "New", you can create a dummy instrument.

Available Instruments
Displays a list of all instrument entries with information on the instrument type, and the user-definable symbolic name.

New, Edit
Accesses the corresponding "Configure Instrument - Edit" or "Configure Instrument - New" dialogs. Since the provided parameters are similar, these dialogs are described only once, see Configure Instruments Settings.

Scan
Searches for instruments on the network, the USB and GPIB interfaces.

Remote command:
`:INSTruments:SCAN` on page 289

Delete
Removes an instrument from the list.

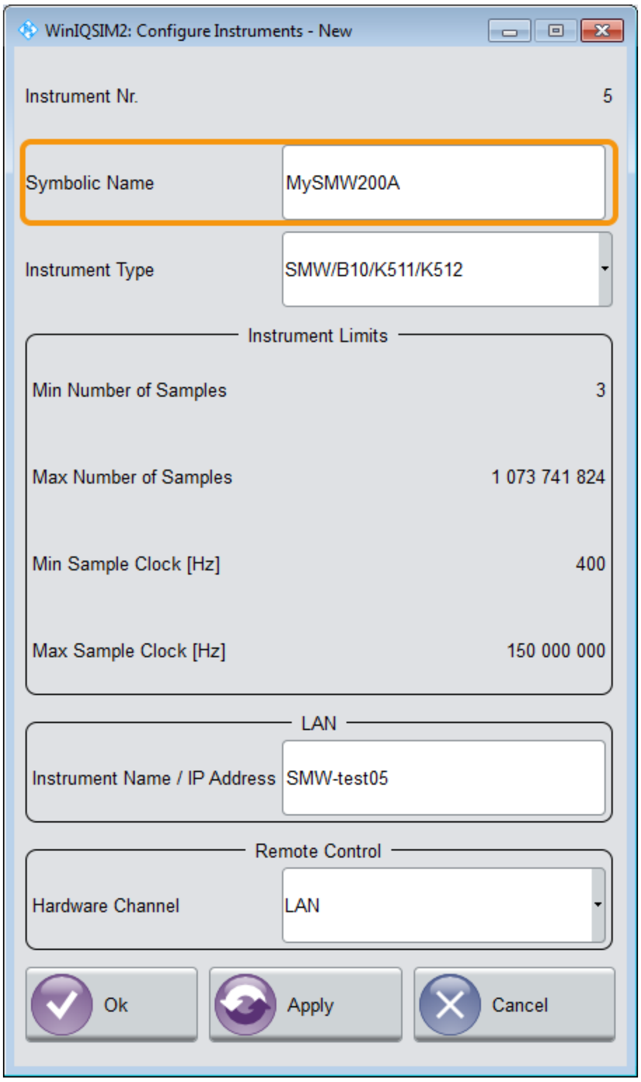
7.2 Configure Instruments Settings

To communicate with a connected instrument on the network or via other interfaces, it must be clearly identifiable. Therefore, each connection requires unique address information.

The "Configure Instruments - Edit/New" dialogs cover the same parameters, therefore the following description applies to both modes.

1. To access these settings, select "Arb Sig Gen/Vector Sig Gen > Instruments...".
2. To define a new instrument, or open an existing instrument configuration, perform one of the following:
   a) Select "New".

b) Select an instrument from the list of available instruments and then select "Edit".



Both, the "Configure Instruments > New/Edit" dialogs contain the parameters for configuring the instrument and the network connection. Additionally, they show the value ranges of the characteristical signal parameters of the currently selected instrument.

Instrument No.
Displays the line number of the instrument entry in the list of available instruments, see Available Instruments Settings.

Symbolic Name
Determines an alias name for the selected instrument.

Instrument Type
Selects a type from the instrument family according to the features of the selected instrument.

Instrument Limits
Displays the samples and clock value ranges of the selected instrument.

Instrument Name / IP Address
Determines the instrument IP address or instrument name in the LAN.

Note: In a LAN that uses a DNS server (Domain Name System server), each PC or instrument connected in the LAN can be accessed via an unambiguous instrument name (hostname) instead of the IP address.
Basically the default instrument name follows the syntax `<instrument>-<serial number>`, e.g. `SMW200A-100000` according to the conventions of Rohde & Schwarz. However, some instruments might be addressed by an instument name with slightly different syntax. In addition, the serial number might not always be located at the rear panel of the instrument.

Check the settings in the instrument setup or the user documentation to obtain the required information on the instrument name.

Hardware Channel
Selects the remote connection.

Each of the available channels requires a unique address information for identification. Therefore, the displayed parameters vary accordingly.

"LAN" Configures the connection in the LAN.
For communication in the LAN, the IP address or the instrument name is required, see Instrument Name / IP Address

"USB" Configures the connection via the USB interface. The USB resource string requires that you enter the "Serial Number".
"Tip:" You find the 6-digit serial number on the rear panel of the instrument.

Example:

100012

"GPIB" Configures the connection via the Interface GPIB (IEC/IEEE Bus Interface) interface. GPIB provides channel addresses from 0 to 30.

Example:

28

Apply, Ok
Applies the settings. "Ok" confirms the settings and closes the dialog.

Cancel
Discards the modifications and closes the dialog.

7.3 Configuring Remote Operating Modes

7.3.1 Remote Control (SCPI)

R&S WinIQSIM2 provides the option to remotely control the instrument for a number of basic parameters that are relevant for the signal generation. These parameters vary depending on the selected generator type (ARB or Vector), and instrument.

The "Remote Control" dialog displays the parameters relevant for exchanging and activating waveforms from R&S WinIQSIM2, and you can modify these settings via remote control.

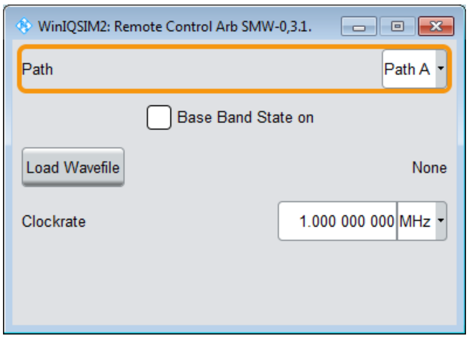
When the connection is active, both, R&S WinIQSIM2 and the instrument synchronize modified settings immediately, regardless of where you perform the changes.

7.3.2 Remote Control Settings of ARB Generators

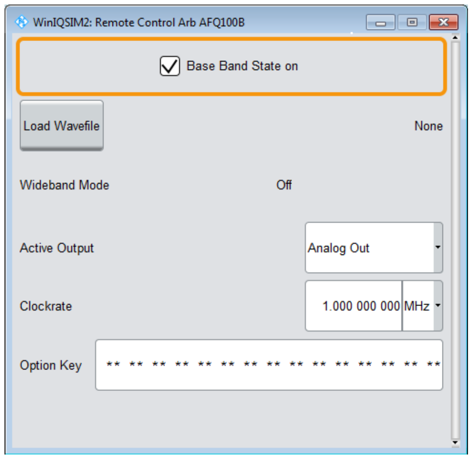
Accessing an ARB generator

► In the block diagram, select "Arb Sig Gen > Remote Control (SCPI)".

Table 7-1: Remote Control ARB <instr>, examples



R&S SMW200A



R&S AFQ100B

The dialog contains the parameters required for the ARB configuration of the appropriate instrument. Depending on the accessed instrument, the parameters in the dialog vary.

Path

Selects the signal path to be controlled.

Applies to instruments with more than one signal path.

Base Band State on
Activates the baseband signal.

Load Wavefile
Accesses the instruments directory, where waveform files are stored. When a waveform file is selected, the instrument loads it into the ARB. On the right of the button, R&S WinIQSIM2 displays the loaded file.

Wideband Mode
Displays the current setting of the wideband mode, when you are working with an R&S AFQ100B.

Active Output
Sets the active output of the instrument to "Analog out" or "Digital out", provided the instrument is equipped accordingly.

Output Type
Selects output type options for analog signal output of an R&S AFQ100B:

"Balanced" Provides symmetrical signals at the inverting and non-inverting outputs.

"Unbalanced" The inverted and non-inverted signals at the outputs are not symmetrical.

"No options for digital output type"
There are no options for the digital output.

Amplitude
Sets the amplitude of an R&S AFQ100B. The value depends on the settings of the "Active Output" and "Output Type". The analog output amplitudes are given in Volt, the digital output amplitudes in a value related to Full Scale (FS).

Clockrate
Sets the clock rate of the instrument.

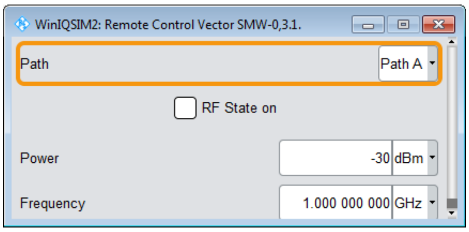
Option Key
Defines the option key for authentication of an R&S AFQ100.

7.3.3 Remote Control Settings of Vector Signal Generators

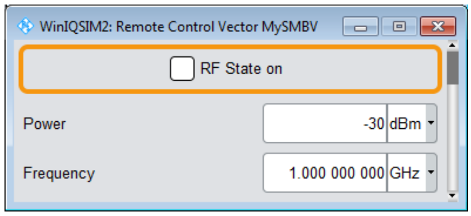
Accessing a Vector Signal Generator

► In the block diagram, select "Vector Sig Gen > Remote Control (SCPI)".

Table 7-2: Remote Control Vector <instr>, examples



R&S SMW



R&S SMBV

The dialog contains the parameters required for the configuration of the appropriate vector signal generator. Depending on the instrument, the parameters in the dialog vary.

Path
Selects the signal path to be controlled.

Applies to instruments with more than one signal path.

RF State on
Activates the RF signal.

Power
Sets the RF power.

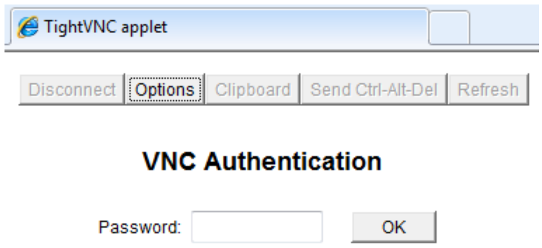
Frequency
Sets the RF frequency.

7.3.4 Remote Desktop

To access the instrument via remote desktop, perform the following steps:

Prerequisites
It is assumend that the instrument is connected to the LAN and configured in R&S WinIQSIM2. In addition, you need the security password of the instrument, and the remote desktop control must be enabled in the instrument (see the user manual of the instrument).

1. In the "ARB Sig Gen or Vector Sig Gen" block, select an instrument.
2. Select "ARB Sig Gen or Vector Sig Gen > Remote Desktop".



The "VNC Authentication" dialog prompts you to enter the instruments' security password.

3. Enter the password and confirm with "OK".

   When the connection is established, you have access to the instrument and you can perform settings directly from your computer.

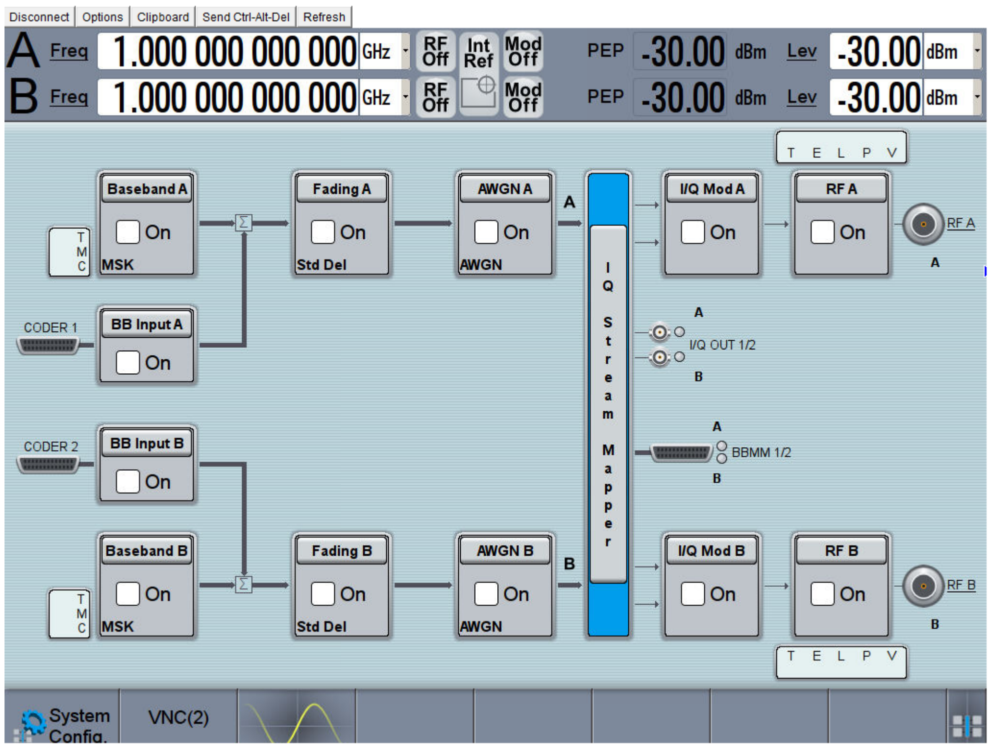


Figure 7-1: Remote desktop access to the R&S SMW

7.4 How To Access and Configure an Instrument

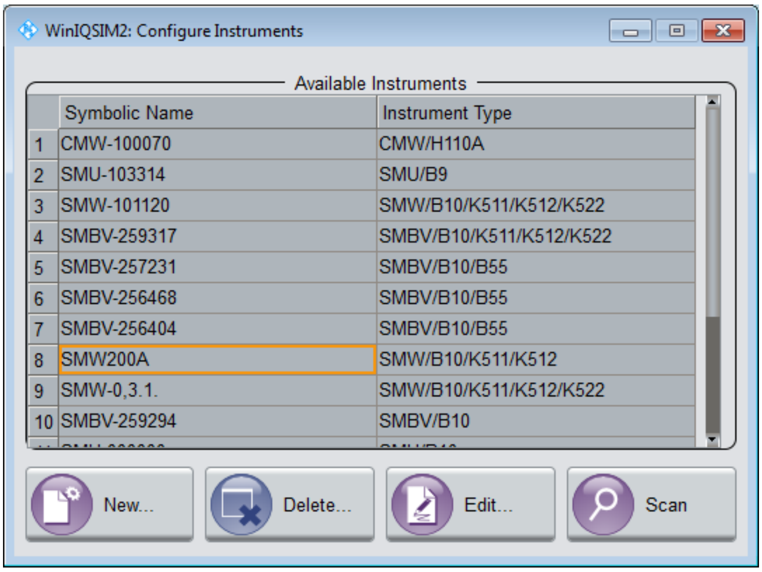
The following are examples that show step by step how to detect instruments in the network, and configure the associated settings in R&S WinIQSIM2.

To scan the network for connected instruments

1. In the block diagram, select "Arb Sig Gen/Vector Sig Gen > Instruments ...".

2. Select "Scan" to search for connected instruments.

   R&S WinIQSIM2 searches for available instruments. This process may take some time.

   When finished R&S WinIQSIM2 displays all instruments found in the your network domain as well as instruments connected via the GPIB or USB interfaces.
   In addition, R&S WinIQSIM2 retrieves information on the connection to the instrument automatically.

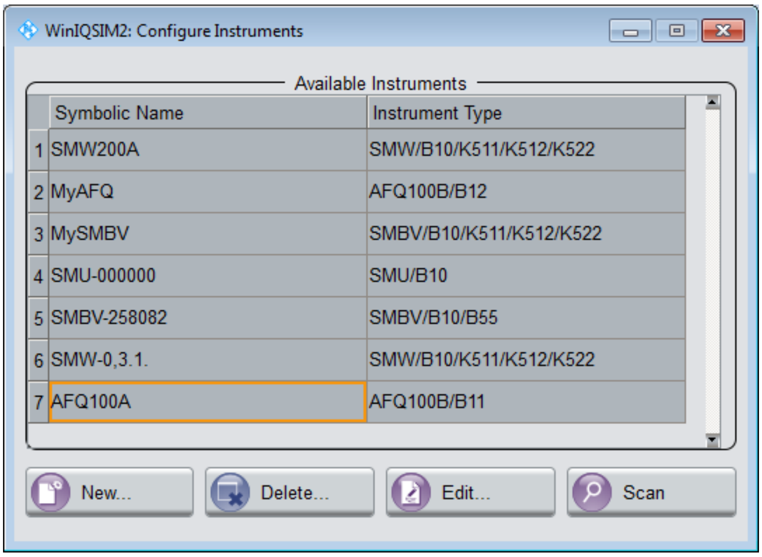
3. To check the network parameters, select the corresponding instrument in the list, and select "Edit".

   The "Configure Instruments" dialog opens, and you can take a look at the settings of the selected instrument, or modify the configuration. For details on the provided see Chapter 7.2, "Configure Instruments Settings", on page 189. For an example on how to set the parameters, see "To configure the instrument" on page 197

To create an instrument manually

1. In the menu bar of R&S WinIQSIM2, select "File > New" to start the software from an initial state.

2. In the block diagram, select "Arb Sig Gen/Vector Sig Gen > Instruments...".



R&S WinIQSIM2 shows the list of all previously assigned entries, even if the instruments are currently not connected.

3. Select "New" to create new instrument entry.

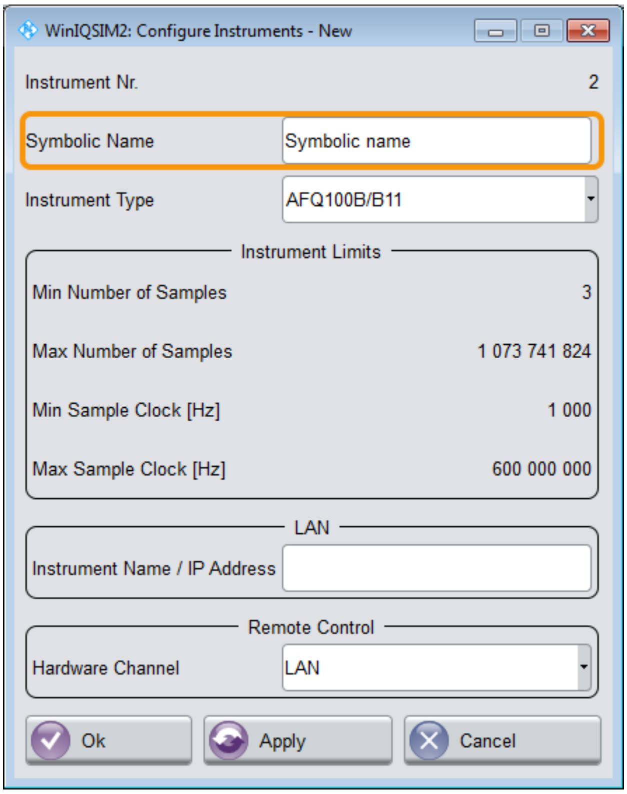
   The "Configure Instruments" dialog opens where you can configure the parameters of the dummy instrument, see "To configure the instrument" on page 197.
   For details on the provided settings, see Chapter 7.2, "Configure Instruments Settings", on page 189.

To configure the instrument

The example shows the steps to be taken for configuring an instrument manually. We assume an R&S SMW200A equipped with extended ARB and bandwidth options is connected via USB.

Note: If you have selected a connected instrument, no further configuration is required.

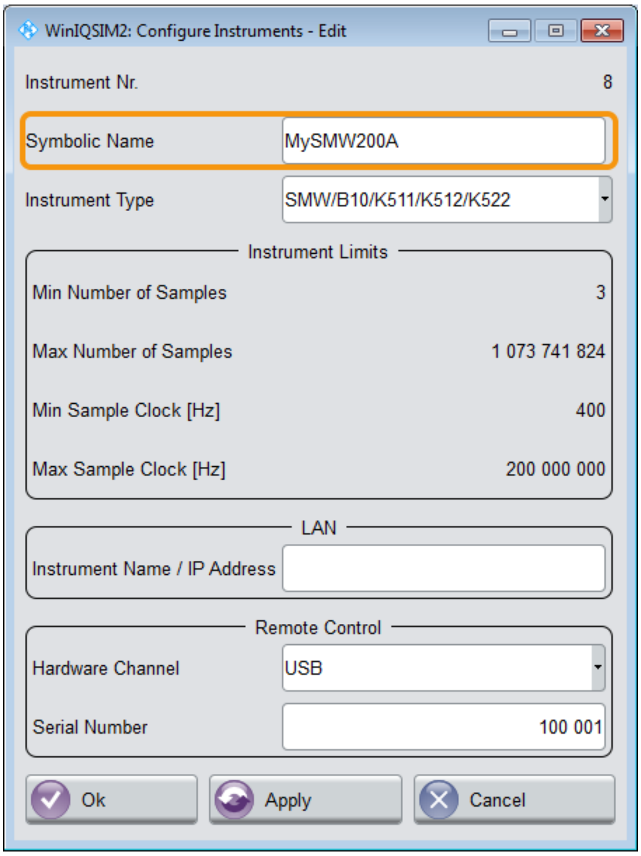
1. To access the instrument settings, select "Arb Sig Gen/Vector Sig Gen > Instruments ... > New/Edit"



First, the dialog box shows any default entries.

2. Enter "Symbolic Name > MySMW200A ".
3. Select "Instrument Type", e.g. "> SMW/B10/K511/K512/K522".

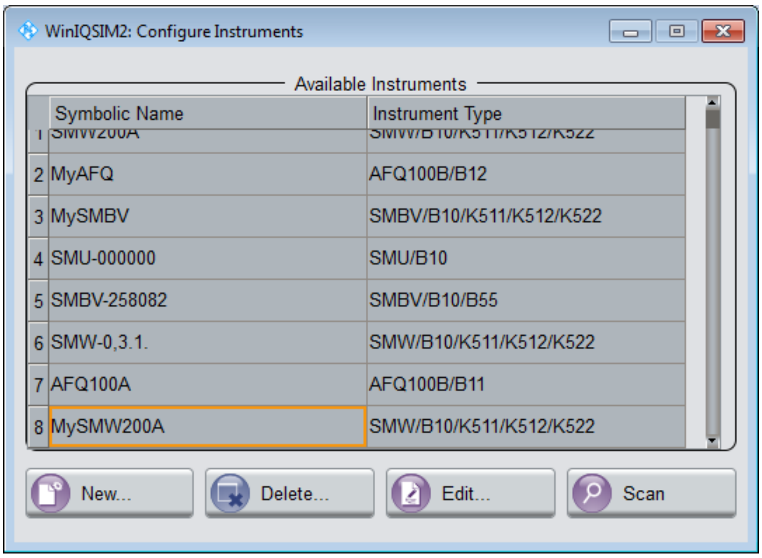
   According to the instrument and its options, R&S WinIQSIM2 indicates the relevant sample parameters of the instrument in the "Instrument Limits" section.
4. Select "Hardware Channel > USB".
5. Confirm with "Apply".



R&S WinIQSIM2 assigns your settings.

6. To close the dialog, select "Ok".

   You can see the new instrument *MySMW200A* added to the list.



To delete an instrument from the list

1. In the block diagram, select "Arb Sig Gen/Vector Sig Gen > Instruments...".

2. In the "Configure Instruments" dialog, select the instrument to be deleted and click the "Delete" button.

   R&S WinIQSIM2 removes the selected entry from the list of available instruments.

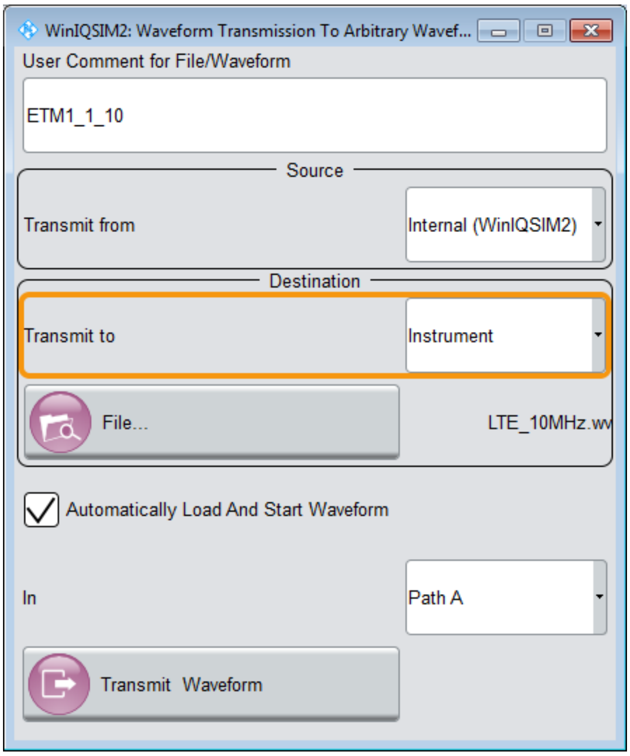
8 Transferring Data

In addition to saving a waveform signal in a file, you can download it directly to an instrument connected via LAN, GPIB or USB. R&S WinIQSIM2 provides a transfer dialog with the necessary setting parameters.

8.1 Waveform Transmission Settings

To configure the data transmission, perform the following steps:

1. Configure the baseband signal that you want generate with the ARB of an instrument.
2. In the menu bar of R&S WinIQSIM2, select "Transmission > Transmit"



The "Waveform Transmission To Arbitrary Signal Generator" dialog contains the parameters required for configuring data transmission or saving.

User Comment for File/Waveform
Enables you to specify a brief remark to the waveform (optionally), e.g. information for identifying the file.

The destination instrument displays the comment in the "Load Waveform" dialog. For example, the R&S SMW200A shows the main characteristics of the waveform file in a tooltip, including the user comment.

Transmit from

Selects the data source.

"Internal (WinIQSIM2)"
Uses last generated internal waveform file.

"File"
Uses a previously stored file.
To select the file, you can access the browser with the displayed File button.

File

Accesses the standard browser for selecting a previously stored waveform file.

Transmit to

Selects the recipient of the waveform file.

"Instrument"
Transmits the waveform file directly to the instrument.
The instrument receives the file and stores it in its user directory. If you have enabled "Automatically Load and Start Waveform" the instrument loads the waveform file directly in the ARB and starts signal generation.

"File"
Stores the waveform into a file on the computer.
To determine the location and file, use the File button.

File

Accesses the standard browser for storing the waveform file. You can select the directory and assign a file name. The file extension is predefined (`*.wv`).

Automatically Load And Start Waveform

Enables the following automatic process sequence when you execute Transmit Waveform:

The generator stores the file on its internal hard disk, loads it into the ARB and starts replaying the signal. The resulting signal is immediately available for the I/Q output.

When disabled, Transmit Waveform transmits the waveform file and stores it on the hard disk of the instrument.

In

Selects the signal path of the instruments for the signal generation.

ARB

Selects the arbitrary waveform generator.

This parameter is provided for connected R&S BTC instruments, since these instruments have several ARB generators in a path.

Transmit Waveform

Executes the waveform transfer according to the configured source and destination media.

8.2 How to Transmit Waveform Data to Instruments or Files

The step by step instructions in this section show you how:

- "To configure the connection and to generate the waveform file" on page 203
- "To transfer the generated waveform signal to the instrument via the network" on page 204
- "To transfer the generated waveform signal to the instrument via USB" on page 204
- "To store the generated waveform signal in a file" on page 205
- "To load a waveform file" on page 205
- "To copy a previously created waveform file" on page 205
- How to transfer waveform data created by R&S WinIQSIM2 to a connected instrument

Using the example of an R&S SMW200A connected in the LAN, we assume that you have already performed the following steps:

- The R&S SMW200A is connected in the LAN and turned on
- R&S WinIQSIM2 is installed on a remote controller also connected in the network.
- R&S WinIQSIM2 is running, reset to an initial state.

This section does not describe any settings for the R&S SMW200A but focuses only the task-related settings. For information on configuration and working with the R&S SMW200A, refer to the user manual of the instrument.

To configure the connection and to generate the waveform file

This instruction leads through the required steps before the data transfer.

1. Perform the steps in "To scan the network for connected instruments" on page 195, to establish the connection to the R&S SMW200A in the LAN.
2. Verify the instrument settings, according to "To configure the instrument" on page 197.
3. In the block diagram of R&S WinIQSIM2, select "Vector Sig Gen > SMW200A".
4. To generate the waveform signal with the required settings:
   a) Follow the example under Chapter 3.3.3, "Generating the Waveform Signal", on page 26
   b) If you want to add an interference signal to the waveform, see Chapter 3.3.4, "Adding Noise (AWGN) to the Signal", on page 29.

To transfer the generated waveform signal to the instrument via the network

At this point, we refer to "To transmit the generated waveform to the R&S SMW200A and activate the ARB signal generation" on page 37 for an example to transmitting the created waveform data to the R&S SMW200A.

► See "To transmit the generated waveform to the R&S SMW200A and activate the ARB signal generation" on page 37.

To transfer the generated waveform signal to the instrument via USB

Alternatively to the file transfer possibility via LAN, GPIB or USB connection, you can use a USB storage device for direct file transfer to the instrument.

It is recommended that you transfer files with user data, e.g. waveform files, data lists, application settings to the instrument, rather than load and play them from a connected USB storage device.

To transfer a file with user data via USB and store it on an instrument, perform the following:

1. Connect a USB storage device, e.g. a USB memory stick to a USB interfaces of the PC.

   The PC recognizes the connected USB storage device automatically and assigns a directory drive.

2. Configure the waveform, e..g. a 3GPP FDD signal.

3. Store the waveform in a file, as described in "To store the generated waveform signal in a file" on page 205 by selecting the target directory on the USB drive letter.

4. Verify the security settings of the destination instrument.
   If disabled:

   a) Enable the file transfer via USB on the instrument.
   b) Enable the write permission to the storage media of the instrument.

   Refer to the unser manual of the instrument for information on how to get the access.

5. Connect the USB memory device to a USB interface of the instrument.

6. On the instruments front panel, press the SAVE/RECALL key.

7. Select "File Manager".

8. In the directory tree, navigate to the USB drive.

9. Select the required file.

10. Select "Copy".

11. In the directory tree, navigate to the internal user directory.

12. Select "Paste".

    The file with user data is transferred to the instrument.

To store the generated waveform signal in a file

► To store the waveform, see "To store the generated waveform file" on page 39.

To load a waveform file

1. In the menu bar, select "Transmission > Transmit".
2. Select "Source > Transmit from > File".
3. Open the browser dialog with the "File" button.
4. Select the required file.
5. Confirm with "Open".

To copy a previously created waveform file

If you want to duplicate for example an existing waveform, you can save a copy with a different name.

1. In the tool bar, select  to open the "Waveform Transmission" dialog.
2. Select "Source > Transmit from > File".
3. Open the browser dialog with the "File" button.
4. Select the required file.
5. Confirm with "Open".
6. Select "Destination > Transmit to > File".
7. Open the browser dialog with the "File" button.
8. Open the browser dialog with "Destination > File".
9. Select the destination directory
10. Enter the new file name.
11. Confirm with "Save".

    R&S WinIQSIM2 stores the same waveform data in the new file.

9 File and Data Management

R&S WinIQSIM2 uses files to save all application data. You can store and load application settings, as well as import and export user data.

The save and recall function for managing of the application settings is available via the "File" menu. Dedicated save/recall functions are available in the settings dialogs of the digital standards or accessible whenever user files are used.

This section focuses on the functions provided for managing of user data files and covers the topics listed below.

For information on the related remote control commands, refer to Chapter 12.4, "MMEMory Subsystem", on page 278.

9.1 About the File System

Depending on the contained information, two file groups are distinguished: program and user files.

R&S WinIQSIM2 stores the program files in a predefined destination directory, as described in "To install the software" on page 22. These files and the system directory are not to be modified.

Therefore, this section focuses on files with user data.

This section is an overview of the file system of R&S WinIQSIM2 and covers the following topics:

- "Types of user data" on page 206
- "File storage Location" on page 207
- "File handling" on page 207
- "File naming conventions" on page 207
- "File extensions" on page 208
- "File Contents" on page 208

Types of user data

The **user data** includes saved settings of the application, data for the different digital standards and import data, as well as the waveform files generated for transmission.

The user data can be roughly divided into the following data types:

- *Settings*, e.g. the frame setting of the GSM/EDGE standard or the current application settings, can be stored and reloaded later.
  See Chapter 9.2, "Storing and Recalling Application Settings", on page 208
- Externally or internally generated *complex modulation and control data* can be loaded.
  See Chapter 9.3, "Accessing Files with User Data", on page 213
- Externally generated *waveforms* can be imported.
  See Chapter 4.8, "Import IQ Data", on page 135

File storage Location

R&S WinIQSIM2 stores user files on an arbitrarily selectable directory, or on an external storage medium, like a memory stick.

Application data storage Location

R&S WinIQSIM2 stores application data in the default application data directory `%APPDATA%\Rohde-Schwarz\winiqsim2`. In manual control, you can access and change the storage directory via "File > Setup > Temporary Files > Browse".

In remote control, you can query the directory with the command `:SYSTem:MMEMory:PATH:USER?`. To change the current directory, use the command `:MMEMory:CDIRectory`.

File handling

To access files and the file system or to use the general file management functions such as copying and moving data, use the standard browser functions (see Chapter 9.6, "Using the File Manager", on page 217).

To transfer files from and to the instruments or to import files, use one of the following alternatives:

- Connect a memory stick to one of the USB interfaces.
  The computer recognizes a connected memory stick and assigns a drive letter automatically.
- Establish a connection to a LAN.
  You can also exchange files with devices / computers that have shared directories on the network for SMB access

  Connected to a LAN you can exchange files via two standard transfer protocols from a remote client:
  – FTP (file transfer protocol)
  – File sharing according to SAMBA/SMB (server message block) protocol.

  See the user manual of the destination instrument, to obtain information on the supported file transfer protocols.

File naming conventions

To enable files to be used in different file systems, the following file naming conventions should be observed.

The *file name* can be of any length and *is case-sensitive*, i.e it is distinguished between uppercase and lowercase letters. All letters and numbers are permitted (numbers are, however, not permitted at the beginning of the file name). Where possible, special characters should not be used. Use of the slashes "\" and "/" should be avoided since they are used in file paths. A number of names are reserved by the operating system, e.g. `CLOCK$`, `CON`, `AUX`, `COM1...COM4`, `LPT1...LPT3`, `NUL` and `PRN`.

File extensions

R&S WinIQSIM2 distinguishes the files according to their extensions; each type of file is assigned a specific file content and hence a specific file extension. The extension is usually of no consequence to the user since access to the files occurs in the individual dialogs where only the relevant type of file is available. For example, files with frame settings can only be saved and loaded in the "GSM/EDGE" dialog.

See Chapter A.2, "Extensions for User Files", on page 419 for an overview of the supported file extensions.

File Contents

In general, a settings file contains user-specific settings of a session, e.g. baseband settings, instrument settings, graphic display settings, etc.

To maintain the file size and to accelerate the loading and processing times, only the settings which differ from their respective default values are stored. Considered is also configuration data for the operating elements and lists with user data, e.g. dialog positions and a data lists. However, if a list data is part of the settings, a reference to this list is stored, not the list itself.

This approach makes sure that the created files contain only relevant information. During the recall process, R&S WinIQSIM2 interprets only the relevant settings; all non-referenced parameters are set to the associated preset values. Error messages indicate the settings which cannot be implemented, like referencing non-existing lists or the attempt to activate settings which are not supported by R&S WinIQSIM2.

9.2 Storing and Recalling Application Settings

This section deals with the save, load, and reset functions of the application settings.

Apart from presetting the application to an initial state, the functions "Save" and "Open", as well as "Save" and "Recall" enable you to store a configured waveform signal for later reuse. These functions store or reload either all performed settings, or exclusively the settings of a digital standard.

For example, if you want to repeat, or restore a specific waveform signal you have already generated. Or you want to transfer the waveform signal to several instruments.

In each of these cases, you can create a file with the complete application settings, or you can choose to store only the settings of a particular digital standard. Both options are scope of this section.

Save/Open the complete application settings

You can save and upload the complete settings of your configuration in a defined storage location with a user-defined filename. The file extension is `* .svrcltxt`. Settings files created in this way are visible in the file system and accessible with the known "File > Open / Save / Save As...." functions for file handling.

The "File > Open" process replaces the current configuration with the reloaded values.

Save/Recall the settings belonging to the digital standards

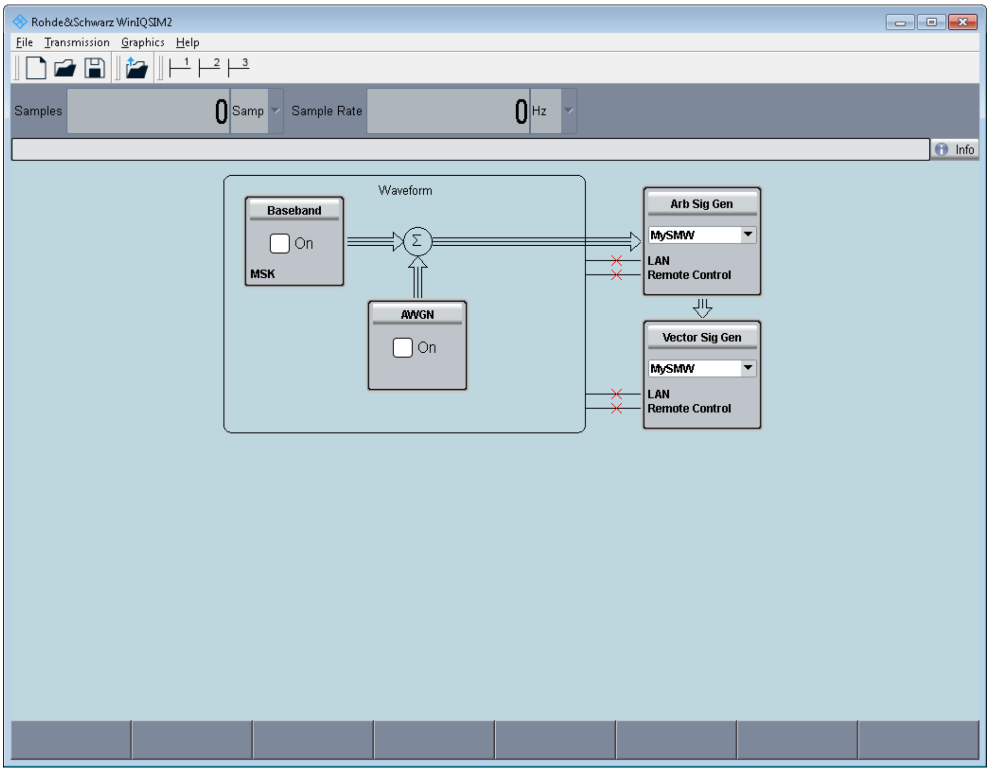
Each of the digital standards provides a special "Save/Recall" function to manage directly the settings associated to the corresponding digital standard, e.g. all settings in the "3GPP FDD" dialog.

The "Save/Recall" functions create files with user-defined names, predefined file extension and on a definable storage location. The files are accessible with the supported methods for file handling.

9.2.1 Resetting the Application

To perform a reset of R&S WinIQSIM2

► In the menu bar, select "File > New".



The function sets the parameters and operating modes to default values predefined in R&S WinIQSIM2.

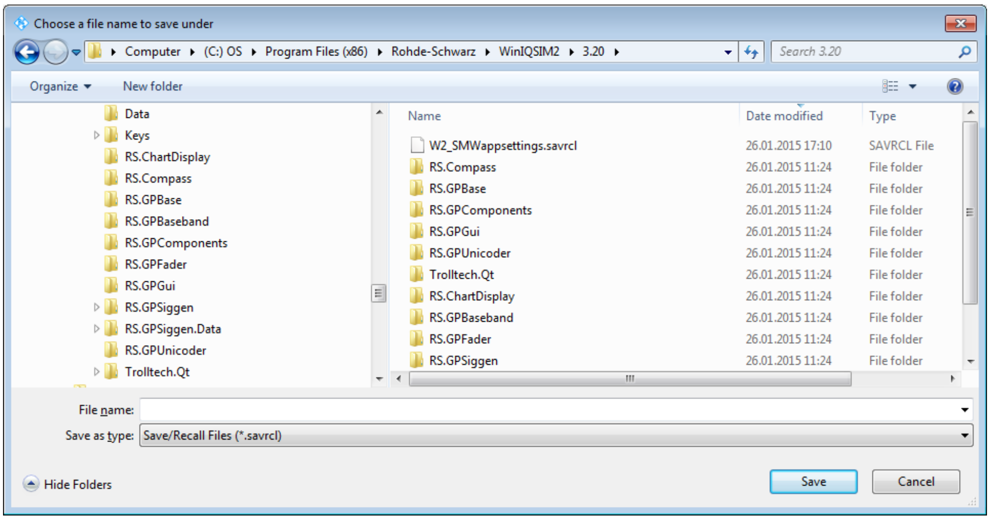
9.2.2 Save / Open Complete Application Settings

To access the dialogs for storing or loading the complete application settings

Depending on the task, perform one of the following:

► In the menu bar,

a) Select "File > Open".
b) Select "File > Save".
c) Select "File > Save As...".



The provided settings of these operations are similar and closely related.
These browser functions detect, display and store only settings files with the file extension `*. svrcl`.

File
Provides the functions for handling of settings files.

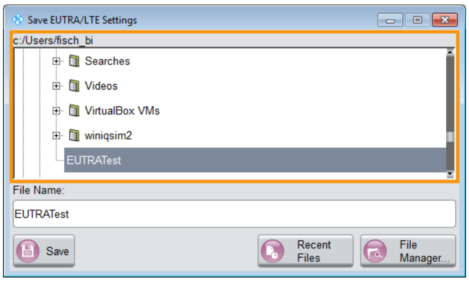
Open ← File
Accesses the standard file browser for loading a previously saved R&S WinIQSIM2 settings file. A file contains the complete application settings.

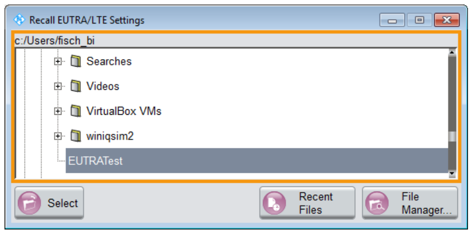
Save / Save As.. ← File
Stores the settings of the current session in a previously selected file or in a new file.

9.2.3 Save/Recall Settings

To access the dialog for storing and loading the settings of a digital standard

► In the general tab of a digital standard dialog, select the "Save" or "Recall".
E.g. select "Block Diagram > Baseband > EUTRA/LTE > General > Save".





The name of the dialog is context-sensitive and varies depending on the performed function and the particular digital standard. The provided functions are similar.

Select Operation
Accesses the functions for storing ("Save") and loading ("Recall") the current configuration of a digital standard.

Directory, File List and File Name
Note:
You access this generic function each time you perform one of the following:
- Store or load (settings) files
- Define a folder these files are to be stored in
- Navigate through the file system

The name of the dialog is context-sensitive but the provided functions are self-explanatory and similar.

With the provided settings, you can ...
- Navigate through the file system, use the directory tree
- Create a new file, load and store files, use the dedicated functions "New", "Select", Using the File Manager and Recent files.
- Perform standard file management functions, like create directories, move, copy, delete files and/or directories, use the standard "File Manager" function (see Using the File Manager).

Remote command:
To list all files in a directory:
`:MMEMory:CDIRectory` on page 281
`:MMEMory:CATalog?` on page 280
Refer to the description of the digital standards for the syntax of the corresponding SCPI command: `[:SOURce<hw>]:BB:<Digital Standard>:SETTing:CATalog`.

Recent files
Displays the files last used.

Save

Saves the current settings belonging to a digital standard under the defined file name.

Remote command:
`:MMEMory:STORe:STATe` on page 286
Refer to the descriptions of the firmware options for the correct syntax of the corresponding SCPI commands:
`[:SOURce<hw>]:BB:<Digital Standard>:SETTing:STORe`
`[:SOURce<hw>]:BB:<Digital Standard>:SETTing:DELete`

Recall

Restores the selected configuration.

During recall, signal generation software considers all related settings, for example active state or lists. An error message indicates the settings which cannot be implemented.

Remote command:
`:MMEMory:LOAD:STATe` on page 284
Refer to the descriptions of the firmware options for the correct syntax of the corresponding SCPI command:
`[:SOURce<hw>]:BB:<Digital Standard>:SETTing:LOAD`

File Manager

Accesses the "File Manager" dialog, see Chapter 9.6, "Using the File Manager", on page 217.

9.2.4 How to Save and Recall Settings

You can save your settings into a file, e.g. for the reuse of the waveform in repeated tests with the same settings.

To save and recall complete application settings

1. In the menu bar, select "File > Save".
2. Select the storage location.
3. To save the settings, perform one of the following:
   a) For an update, select an existing file.
   b) For creating a new file, assign the file name.
4. Confirm with "Save".

   R&S WinIQSIM2 stores the current configuration in the file with the defined name and path and the extension `*.savrcltxt` is created.
5. To restore settings, select "File > Open" in the menu bar.
6. In the target directory, select the previously stored settings file.

7. Confirm with "Open".

   The settings are restored and you can process the waveform signal with the same settings.

9.3 Accessing Files with User Data

By the calculation of signals according to the different digital standards or by the generation of custom digitally modulated signals, R&S WinIQSIM2 uses the data from different data sources (see also Chapter 4.3.1.1, "Data and Signal Sources", on page 55). To simulate the signal of one UE for instance, R&S WinIQSIM2 modulates and codes the provided data for each channel and further processes the signal as described in the corresponding standard.

Whenever a data list file is enabled as a data source, R&S WinIQSIM2 provides direct access to the standard "File Select" function. You can select, create and edit data list files.

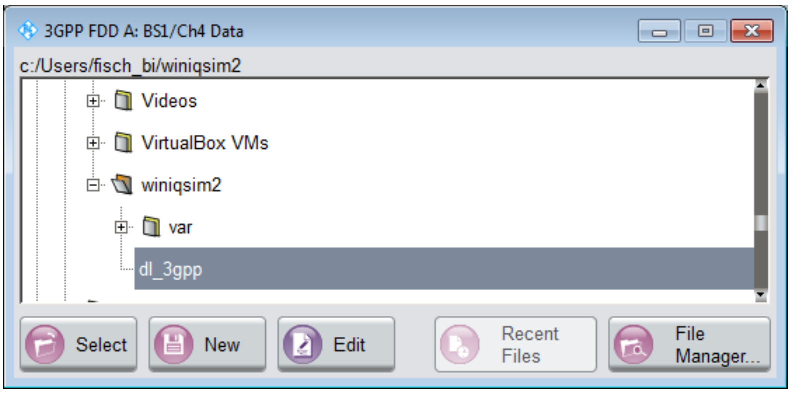
9.3.1 File Select Settings

The "File Select" dialog opens automatically each time you select a data list file as data source or you select a control list.

To access a loadable data list file

1. Select the "Data List Name" in the individual dialog, e.g. for "Baseband > 3GPP FDD > Basestations > BS1 > Channel Table > P-CCPCH > Data = Data List" select "DList = None".

   A "File Select" dialog for loading, creating and modifying a file is displayed.

   Tip: The name of the dialog is context-sensitive and varies according to the digital standard and accessing function. However, the provided functions are similar.

2. To load an exisitng file:

   a) In the file system, navigate to the destination folder.

   b) Select the file.

c) Confirm with "Select".

3. To create a file, for example if there is no data list file specified:

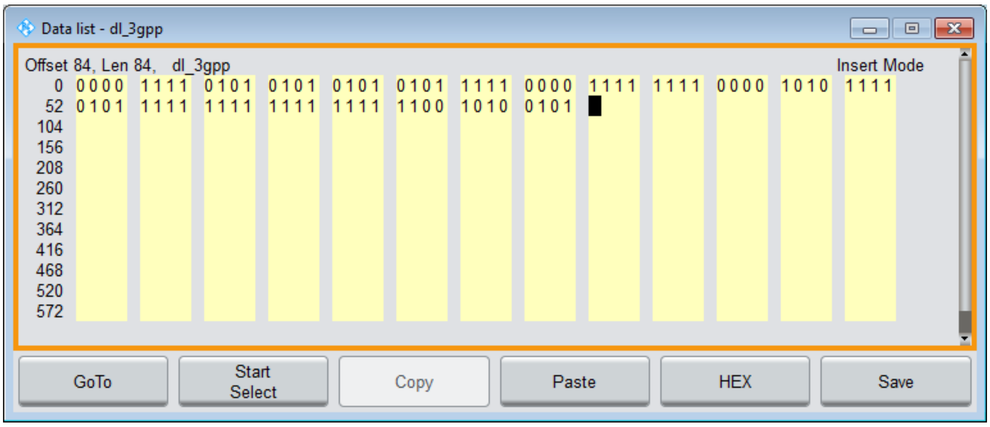
   a) In the file system, navigate to the destination folder.
   b) Select "New".
   c) Enter the file name.
   d) Confirm with "Ok".

   R&S WinIQSIM2 creates and stores a new empty file.

4. To edit an existing or newly created file:

   a) In the file system, navigate to the destination folder.
   b) Select the file.
   c) To open the fle, select "Edit".

   The standard "Data List Editor" dialog opens, see Chapter 4.4.2.8, "Data List Editor", on page 79.



5. Edit the file content.

6. Confirm with "Save".

Directory, File List and File Name
Note:
You access this generic function each time you perform one of the following:
- Store or load (settings) files
- Define a folder these files are to be stored in
- Navigate through the file system

The name of the dialog is context-sensitive but the provided functions are self-explanatory and similar.

With the provided settings, you can ...
- Navigate through the file system, use the directory tree
- Create a new file, load and store files, use the dedicated functions "New", "Select", Using the File Manager and Recent files.
- Perform standard file management functions, like create directories, move, copy, delete files and/or directories, use the standard "File Manager" function (see Using the File Manager).

Remote command:
To list all files in a directory:
`:MMEMory:CDIRectory` on page 281
`:MMEMory:CATalog?` on page 280
Refer to the description of the digital standards for the syntax of the corresponding SCPI command: `[:SOURce<hw>]:BB:<Digital Standard>:SETTing:CATalog`.

Functions for handling of data lists
Provided are the following standard functions for file handling:

"Select" Select and load the file.

Remote command:
`[:SOURce<hw>]:BB:DM:DLISt:SELect` on page 347
Refer to the descriptions of the firmware options for the correct syntax of the corresponding SCPI command:
`[:SOURce<hw>]:BB:<Digital Standard>:...:DATA DLISt`
`[:SOURce<hw>]:BB:<Digital Standard>:...:DSELect`

"New" Creates a file with the specified "File Name".
To confirm, select "OK"; use "Cancel" to undo the operation.
To edit the file content, select "File Select > Edit".

"Edit" Accesses the "Data List Editor" and loads the selected file for editing, see Chapter 4.4.2.8, "Data List Editor", on page 79.

Functions for handling of control lists
Provided are the following standard functions for file handling:

"Select" Select and load the file.

Remote command:
`[:SOURce<hw>]:BB:DM:CLISt:SELect` on page 345

"New" Creates a file with the specified name. To edit the file content, select "File Select > Edit".

"Edit" Accesses the "Control List Editor" and loads the selected file for editing, see Control and Marker Lists Editor.

Recent files
Displays the files last used.

File Manager
Accesses the "File Manager" dialog, see Using the File Manager.

9.3.2 How to Create and Access Data and Control Lists

The general principle for accessing files with user data is described in "To access a loadable data list file" on page 213. Detailed and further information on how to create and access files with user data like data lists and control lists is provided in the following sections:

- Chapter 4.4.3.2, "How to Create and Assign a Data List", on page 87
  Overview of the possible ways and detailed description on how to create data lists

- Chapter 4.4.3.1, "How to Create and Assign a Control List", on page 86
  Overview of the possible ways and detailed description on how to create control lists
- "Storing and Loading Current Settings" on page 279
  Information on accessing files in a remote environment

9.4 Exporting Remote Command Lists

To set specific instrument settings or perform a task automatically, you can create scripts that contain the settings in the form of remote control command sequences.

You can either record or manually create SCPI lists, or generate a list of the current application state in one step, see .

Completed scripts are stored in files, converted to different formats, depending on the used language of the source code.

The R&S WinIQSIM2 supports the most commonly used languages by default, as there are:

- Plain SCPI: `*.txt`
- MATLAB: `*.m`
- NICVI: `*.c`

It is also possible, to convert the SCPI command list to a user-specific language, see .

9.5 Loading, Importing and Exporting Lists

Although R&S WinIQSIM2 provides built-in editors for creating data and control lists, as well as lists for the list mode or lists with user correction data, you may want to create or evaluate them with an external application. The instrument provides interfaces with the following functionality:

- Import and export list files in a standard ASCII format file
- Load files with modulation and control information
- Load of internally or externally generated waveform file

Lists are stored and loaded in the appropriate dialogs. For example, the user correction data list is created and stored in the "User Correction" dialog; waveform files are created with the signal generation software R&S WinIQSIM2 or in the dialogs of some of the digital standards.

For more information, refer to Chapter 9.3.2, "How to Create and Access Data and Control Lists", on page 215.

9.6 Using the File Manager

The "File Manager" is a tool similar to the standard Windows Explorer that helps you to manage mass storage media and files.

It is embedded in R&S WinIQSIM2, especially designed for managing file formats for applications in signal generation. It supports direct data exchange between instruments from Rohde & Schwarz.

You can perform the following tasks:

- Copying files from disk to other media.
  See Chapter 9.7, "Transferring a File to an Instrument", on page 219
- Copying files into another directory, renaming and deleting files.
  See "Cut, Copy&Paste and Delete" on page 219
- Creating directories.
  See "Create New Directory" on page 219

Access:

You can access the "File Manager" via the "File Manager" button. It is available in all dialogs that provide loading or saving data from or to files, e.g. in the dialogs of the *digital Standards*, *Custom digital Modulation* or list management. The instruction leads you to the file manager, by the example of the digital standard application 3GPP FDD:

1. In the block diagram, select "Baseband > 3GPP FDD".
2. Select "Recall".

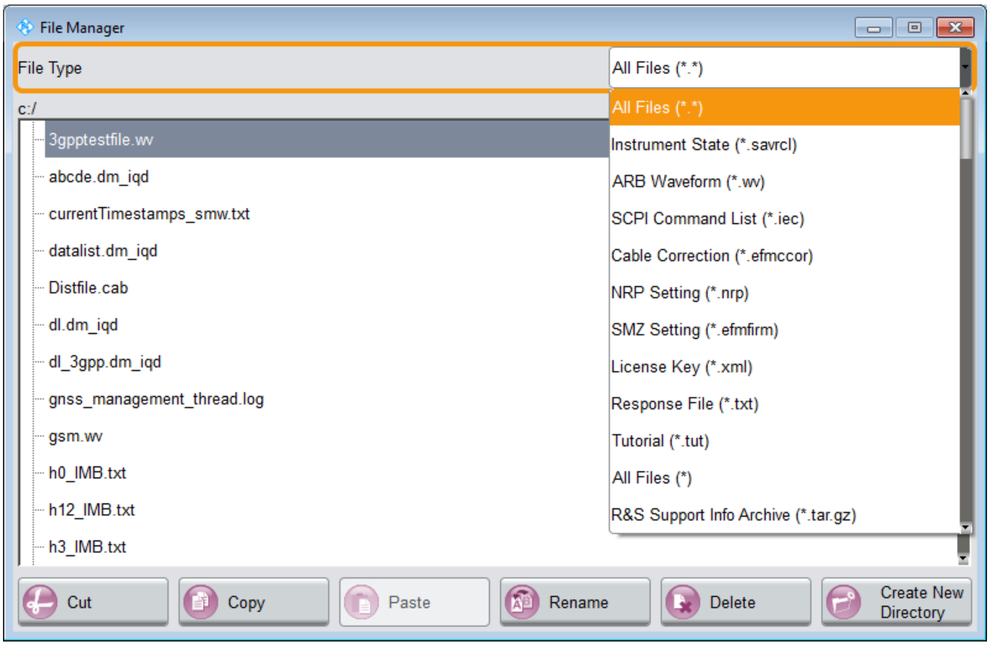
   Tip: Each "Save/Recall" dialog and each "File Select" dialog provides a quick access to the "File Manager", i.e. whenever you select data lists or files with user data.
3. Select "File Manager".

9.6.1 File Manager Settings

Access:

1. In the block diagram, select e.g. "Baseband > 3GPP FDD".
2. In the "General" tab, select "Recall".

3. Select "File Manager".



The dialog provides all standard functions required to manage files. It displays the contents of the selected folder and provides functions to rename, delete, copy or to move files.

Settings:

File Type

Selects the file types to be indicated. If a file type with a specific file extension is selected, only files with this particular extension are listed.

See Chapter A.2, "Extensions for User Files", on page 419 for an overview of the supported file extensions.

Directory and File Name

Allows you to navigate in the contents tree to select the directory and file.

The dialog lists all files of the selected directory, and highlights the selection. The path is indicated above the directory tree.

Unlike the "Save/Recall" and "File Select" dialogs, the "File Manager" displays the full file names including extensions.

Remote command:
`:MMEMory:CDIRectory` on page 281

Cut, Copy&Paste and Delete
Standard file management functions.

Before a file is deleted, a message prompts you to confirm the deletion.

Remote command:
`:MMEMory:DELete` on page 284
`:MMEMory:COPY` on page 282

Rename
Renames the selected file or directory.

Remote command:
`:MMEMory:MOVE` on page 285

Create New Directory
Creates a folder and opens an edit dialog to enter name and path (absolute or relative to the current directory) of the new folder.

Remote command:
`:MMEMory:MDIRectory` on page 285

9.7 Transferring a File to an Instrument

As explained in Chapter 8, "Transferring Data", on page 201, you can transfer a waveform file to an instrument via one of the following ways:

- Directly to an instrument connected over LAN, GPIB or USB, see "To transmit the generated waveform to the R&S SMW200A and activate the ARB signal generation" on page 37 for an example.
- Manually via a connected USB storage device, as described in "To transfer the generated waveform signal to the instrument via USB" on page 204.

Mainly because of security reasons, the access to the file system of an instrument can be denied, because one or all of these access methods are deliberately disabled. Access to the file system via LAN and/or USB requires that the corresponding service is enabled and a write access to the file system is enabled.

To get more detailed information on the security settings of your instrument, refer to the corresponding user manual. For comprehensive information, refer to the application note 1GP72: Connectivity of Rohde&Schwarz Signal Generators.

10 General Functions of the Signal Generation Software

The general functions include basic settings, regardless of the selected operating mode and signal generation. You can access these settings mainly in the file menu or, alternatively via the icons in the toolbar in the main application window.

Refer to Menu bar entries / toolbar icons / keyboard shortcuts for an overview.

The general functions of the menu entries at a glance:

- "File > New", presets R&S WinIQSIM2 to a predefined state, see Chapter 9.2.1, "Resetting the Application", on page 209.
- "File > Open, Save, Save As", saves application settings for reuse, see Chapter 9.2, "Storing and Recalling Application Settings", on page 208.
- "File > Setup", provides information on the software, see Chapter 10.1, "Setup", on page 221
- "Exit", shuts down the application.
  R&S WinIQSIM2 stores the current settings, and reloads this configuration with the next start.
- "Help", provides access to the online help and information on the software version, see Chapter 3.6, "Getting Information and Help", on page 50.

For the description of the application-specific menu entries, see:

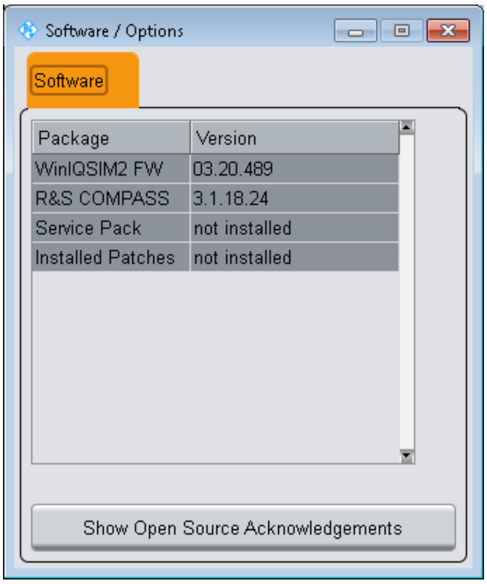
- "Transmission", provides the settings for configuring instruments and data transfer.
  – "Instruments...", see Chapter 7, "Setting Up Instruments", on page 187.
  – "Transmit", see Chapter 8, "Transferring Data", on page 201.
- "Graphics", see Chapter 6, "Displaying Simulated Waveforms Graphically", on page 162

10.1 Setup

10.1.1 Software/Options Settings

Access:

► Select "File > Setup > Software/Options".



The dialog shows the software components of the firmware.

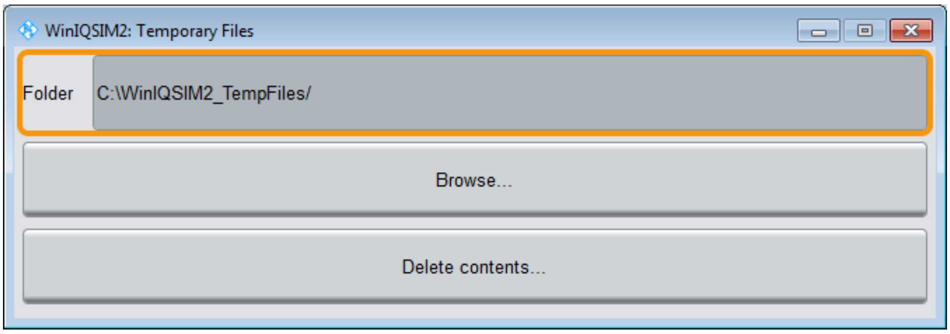
Software

Shows the versions of the software package and the software platform.

Note: Software updates as well as the release Notes describing the improvements and modifications are provided on the Internet at the download site of the Rohde & Schwarz WinIQSIM2 home page. This home page always offers the latest information on R&S WinIQSIM2. How to install / uninstall the software, as well as changes of the software installation procedure, is also described in the release notes.

10.1.2 Temporary Files

► To access the "Temporary Files" dialog, select "File > Setup > Temporary Files".



The dialog contains the parameters for selecting the file location or removing temporary files.

Folder
Indicates the directory where temporary files are stored.

Browse
Accesses the browser for selecting the file location for temporary files.

Delete contents...
Deletes all temporary files in the selected directory.

10.1.3 Undoing or Restoring Actions

"Undo" is a function that removes the effect of the last action on the instrument and reverts it to an older state. Conversely, "Redo" restores a previously undone action.

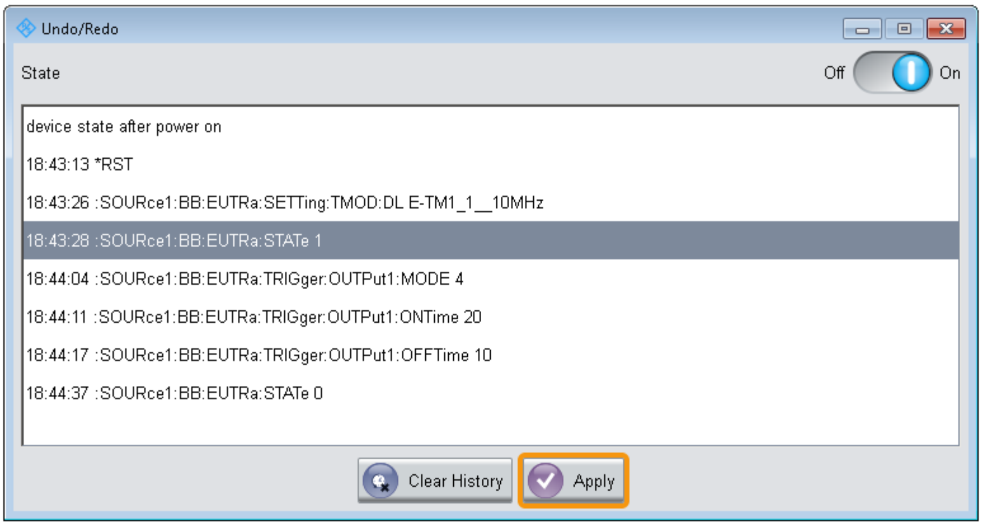
You can "Undo/Redo" actions according to two criteria:

- Step by step
  Gradually undo/redo the actions in reverse order as previously performed.
  Depending on the available memory, the "Undo/Redo" steps can restore all actions.
- Multiple steps at once
  Select any specific action in the history list to "Undo/Redo" multiple actions in a single step.
  Note: This mode requires a system restoration file on the instrument.

10.1.3.1 Undo/Redo Settings

Access:

► Select "Setup > Settings > Undo/Redo".



The dialog contains all functions for enabling the "Undo/Redo" functionality.

State
Enables the recording of the performed actions.

History List
Lists the performed actions, provided "Undo/Redo" state is "On".

Clear History
Deletes the recorded list of the performed steps.

Apply
Performs the "Undo/Redo".

If you select a previously performed action of the list, all subsequent actions are undone. The list entries remain.

If you select a subsequently executed action, you can restore all the actions undone up to this state.

10.1.4 How to Select the File Location and Delete Temporary Files

To determine the storage location for temporary files

In this example, we create a destination folder for temporary files. If the directory already exists, skip the second step.

1. In the main application window, select "File > Setup > Temporary Files..."

R&S WinIQSIM2 displays the current storage location for temporary files in the "Folder" field.

2. Select "Browse" to access the browser, e.g. to create a folder, or to select the new destination folder.
3. Confirm with "Select Folder".

   R&S WinIQSIM2 changes the destination location and displays the complete directory name in the "Folder" field.

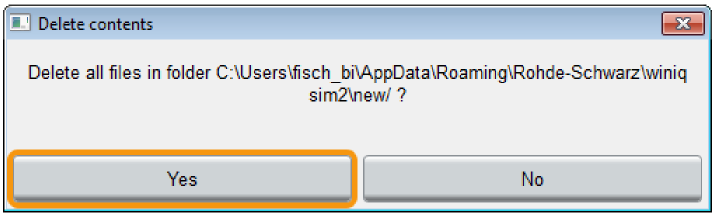
To delete temporary files

Note: This function deletes all files in a directory.

Therefore, before deleting content, verify the currently selected directory, shown under "Folder". If necessary, select the destination folder, as described in To determine the storage location for temporary files.

1. To delete the contents in the folder, select "Delete Contents".

   R&S WinIQSIM2 prompts you to confirm deleting of contents.



2. Confirm with "Yes"

   The function removes all temporary files in the selected folder.

10.2 Querying Error Messages & Info Key

The R&S WinIQSIM2 monitors the functions performed and automatically detects errors and irregularities. The application displays the messages in the info-line and records them in the background with a detailed description.

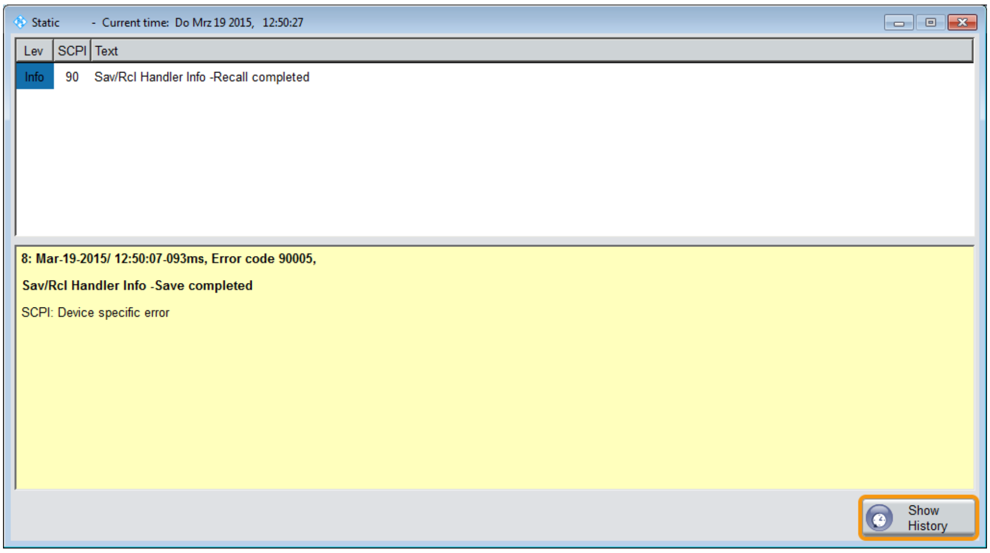
If any critical error occurs, R&S WinIQSIM2 automatically shows the icon in the taskbar. Select the icon to obtain information on the error and the number of occurrences. The icon is assigned to permanent messages, i.e. the message and icon are displayed until the error is eliminated.

Find details to the system messages also in Chapter A.3.1, "Status Information Displayed in the Info Line", on page 421 and Chapter 13, "Troubleshooting and Error Messages", on page 392.

To access the history view with the recorded messages, perform the following:

► In the main application window, select "Info line > Info".

Tip: For some messages, the information line appears briefly in the block diagram, and provides the option to open the info dialog using the "Info" button.

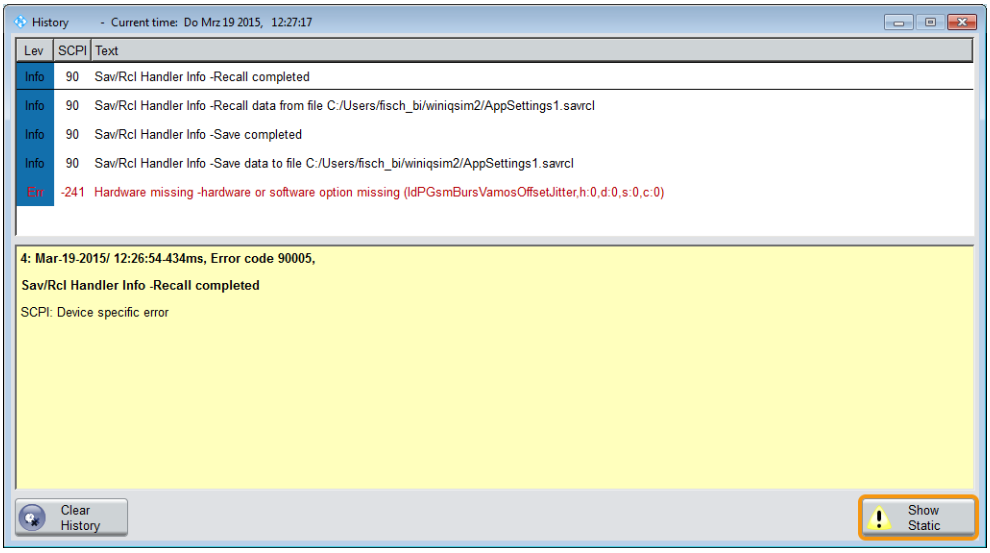


The "Static" info dialog contains the last monitored message in the upper section. The lower section displays additional information on the highlighted message.

10.2.1 History Settings

To access history view with the recorded messages, perform the following:

► In the main application window, select "Info line > Info".
Tip: For some messages, the information line appears briefly in the block diagram, and provides the option to open the info dialog using the "Info" button.



In "History" view, the dialog displays a list of all accumulated messages with a brief detailed description to each message, and provides standard functions for handling the messages.

Show History / Show Static
Selects the view of the info dialog.

- "Static"
  Shows the last monitored message.
- "History"
  Shows all the accumulated messages.

"LEV"	Type of message, abbreviated, and color coded. The following messages types are distinguished: ● "Err": Error message (red colored) ● "Info": Information message (black) ● "Sys": System message ● "Crit": Critical message (red) See also Chapter A.3.1, "Status Information Displayed in the Info Line", on page 421.
"SCPI"	SCPI error code of a message.
"Text"	Description of the message.

Clear History ← Show History / Show Static
Deletes the history list.

Remote command:
`:SYSTem:ERRor:ALL?` on page 382
Each time a `SYST:ERR:ALL?` query is sent, the error queue is returned and at the same time cleared.

10.2.2 How to Manage Messages in the History View

Since the functions in the history view are self-explanatory, the following instructions show you a brief outline of their use.

► To access the "History" view, select "Info line > Info".

To display all accumulated messages

► In the lower right corner of the "History" dialog, select "Show History".

R&S WinIQSIM2 displays all messages generated during the current session, with the most recent message on top of the list.

To get additional information to a message

► In the upper panel of the "History" dialog, select the message.

The lower section displays additional information to the selected message.

To delete the history list

Note: You can delete the history in the history view. The static view does not provide this function.

1. In the "History " dialog, select "Show History".
2. Select "Clear History".

   R&S WinIQSIM2 removes the messages accordingly.

11 Automation of R&S WinIQSIM2

The description in this section requires basic knowledge of the remote control operation. Definitions specified in the SCPI standard are not provided.

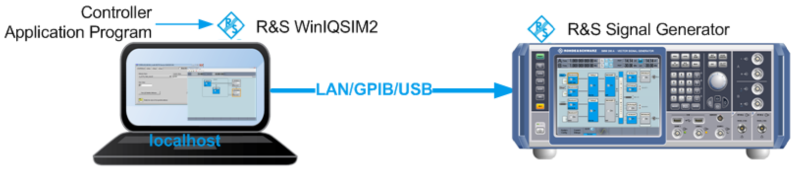
You find some basic information to the SCPI syntax, command lists, and general programming recommendations in Chapter A.1.1, "Additional Basics on Remote Control", on page 397. In addition, this section provides information on the status reporting system of the instrument:

As an alternative to manual operation via the user interface, you can operate R&S WinIQSIM2 also from a remote controller. Remote operation allows automation of the configuration process and is especially useful when a higher configuration speed is required.

Using an appropriate controller program, you can operate R&S WinIQSIM2 remotely. R&S WinIQSIM2 can run either on the controller PC or on a separate PC connected via LAN.

The figures below show the possible scenarios for remote operation of R&S WinIQSIM2 including a connected instrument:

- The controller and R&S WinIQSIM2 are on the same PC.



- In some rare cases, the controller and the software are installed on two separate PCs.

The following sections focus on the possibilities to control R&S WinIQSIM2 remotely. The contents include basics on the remote operation modes, the interfaces and examples for setting up a remote control session with several access tools.

Do not mistake *remote control operation* with the remote control functions provided in the "Arb/Vector ...Sig Gen" blocks, i.e. "Remote Control (SCPI)" and "Remote Desktop". R&S WinIQSIM2 uses these remote operating modes for controlling a connected instrument, see Chapter 7.3, "Configuring Remote Operating Modes", on page 192.

Remote Control (SCPI)

Remote control operation of R&S WinIQSIM2 is characterized by:

- A controller program controls R&S WinIQSIM2, usually via VISA ("Virtual Instrument Software Architecture") interfaces.
- The GUI is not visible.
- Remote control commands (SCPI) perform the settings, either individually or in sequences (SCPI programs).

For background information about the SCPI command structure and basic information on operating the R&S WinIQSIM2 software via remote control, refer to Chapter A.1.1, "Additional Basics on Remote Control", on page 397.

A detailed description of specific SCPI commands available for R&S WinIQSIM2 is provided in Chapter 12, "Remote Control Commands", on page 251. The remote control commands for the digital standards are described in detail after each related function section.

11.1 Remote Control Interfaces and Protocols

R&S WinIQSIM2 supports various protocols to be remotely controlled via the LAN interface:

Table 11-1: Remote control protocols for controlling R&S WinIQSIM2

Interface	Protocols, VISA*) address string and library	Remarks

Local Area Network (LAN)	• **HiSLIP** high-speed LAN Instrument Protocol (IVI-6.1)
	`TCPIP::host address::hislip0[::INSTR]`
	VISA
	• **VXI-11**
	`TCPIP::host address[::LAN device name][::INSTR]`
	VISA
	• **socket communication** (Raw Ethernet, simple Telnet)
	`TCPIP::host address[::LAN device name]::<port>::SOCKET`
	VISA or socket controller
		The interface is based on TCP/IP and supports various protocols.
		For a description of the protocols refer to:
		• Chapter 11.1.1.2, "HiSLIP Protocol", on page 231
		• Chapter 11.1.1.3, "VXI-11 Protocol", on page 232
		• Chapter 11.1.1.4, "Socket Communication", on page 232

*) VISA is a standardized software interface library providing input and output functions to communicate with instruments. A VISA installation on the controller is a prerequisite for remote control over LAN (when using VXI-11 or HiSLIP protocol), USB and serial interface. For remote control via socket communication VISA installation is optional. For more information, see Chapter 11.1.2, "VISA Library", on page 232.

Rohde & Schwarz provides the standardized I/O software library R&S VISA for communication via TCP/IP (LAN: HiSlip, VXI-11 and raw socket), USB (USBTMC) or serial interfaces.

R&S VISA is available for download at the Rohde & Schwarz website http://www.rohde-schwarz.com/rsvisa.

How to configure the remote control interfaces is described in Chapter 11.2, "How to Set up a Remote Control Connection", on page 233.

SCPI (Standard Commands for Programmable Instruments)

SCPI commands - messages - are used for remote control. Commands that are not taken from the SCPI standard follow the SCPI syntax rules. The instrument supports the SCPI version 1999. The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-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.

Tables provide a fast overview of the bit assignment in the status registers. The tables are supplemented by a comprehensive description of the status registers.

For more information, see also Chapter A.1.1, "Additional Basics on Remote Control", on page 397.

11.1.1 LAN Interface

For remote control in a network, the controller PC and the PC with R&S WinIQSIM2 (if not the same) must be connected via the LAN interface to a common network with TCP/IP network protocol. The controller software and, for particular protocols the VISA program library must be installed on the controller PC.

Identifying devices in a network

If several devices (PCs, instruments) are connected to the network, each device has its own IP address and associated resource string. The controller identifies these devices by means of the resource string.

11.1.1.1 VISA Resource Strings

The VISA resource string is required to establish a communication session between the controller and R&S WinIQSIM2 in a LAN. The resource string is a unique identifier, composed of the specific IP address of the PC and some network and VISA-specific keywords.

TCPIP::host address[::LAN device name][::INSTR]

TCPIP	= designates the network protocol
host address	= designates the IP address or hostname of the instrument
[::LAN device name]	= defines the protocol and the instance number of a subinstrument
[::INSTR]	= indicates the instrument resource class (optional)

The **IP address** (host address/computer name) is used by the programs to identify and control R&S WinIQSIM2. It is automatically assigned by the DHCP server the first time the device is registered on the network. Alternatively, you can also assign its **LAN device name**.

See below the characteristics of the VISA resource strings for the corresponding interface protocols. The highlighted characters are crucial.

HiSLIP

TCPIP::host address::**hislip0**[::INSTR]

hislip0 = HiSLIP device name, designates that the interface protocol HiSLIP is used (mandatory)

hislip0 is composed of [::HiSLIP device name[,HiSLIP port]] and must be assigned.

For details of the HiSLIP protocol, refer to Chapter 11.1.1.2, "HiSLIP Protocol", on page 231.

VXI-11

TCPIP::host address**[::inst0]**[::INSTR]

[::inst0] = LAN device name, indicates that the VXI-11 protocol is used (optional)

inst0 currently selects the VXI-11 protocol by default and can be omitted.

For details of the VXI-11 protocol, refer to Chapter 11.1.1.3, "VXI-11 Protocol", on page 232.

Socket Communication

TCPIP::host address::**port::SOCKET**

port = determines the used port number
SOCKET = indicates the raw network socket resource class

Socket communication requires the specification of the port (commonly referred to as port number) and of "SOCKET" to complete the VISA resource string with the associated protocol used.

The registered port for socket communication is port 5025.

See also Chapter 11.1.1.4, "Socket Communication", on page 232.

11.1.1.2 HiSLIP Protocol

The HiSLIP (**Hi**gh **S**peed **L**AN **I**nstrument **P**rotocol) is the successor protocol for VXI-11 for TCP-based instruments specified by the IVI foundation. The 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

Using VXI-11, each operation is blocked until a VXI-11 device handshake returns. However, using HiSLIP, data is sent to the device using the "fire and forget" method with immediate return. Thus, a successful return of a VISA operation such as `viWrite()` does not guarantee that the instrument has finished or started the requested command, but is delivered to the TCP/IP buffers.

For more information see also the application note: 1MA208: Fast Remote Instrument Control with HiSLIP.

11.1.1.3 VXI-11 Protocol

The VXI-11 standard is based on the ONC RPC (Open Network Computing Remote Procedure Call) protocol which in turn relies on TCP/IP as 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.

11.1.1.4 Socket Communication

An alternative way for remote control of the software is to establish a simple network communication using sockets. The socket communication, also referred to as "Raw Ethernet communication", does not necessarily require a VISA installation on the remote controller side. It is available by default on all operating systems.

The simplest way to establish socket communication is to use the built-in telnet program. The telnet program is part of every operating system and supports a communication with the software on a command-by-command basis. For more convenience and to enable automation by means of programs, user-defined sockets can be programmed.

Socket connections are established on a specially defined port. The socket address is a combination of the IP address or the host name of the instrument and the number of the port configured for remote-control. All R&S WinIQSIM2 use port number 5025 for this purpose. The port is configured for communication on a command-to-command basis and for remote control from a program.

11.1.2 VISA Library

VISA is a standardized software interface library providing input and output functions to communicate with devices and applications. Thus, you can configure the interface without having to adjust the application program to the used interface. The I/O channel (LAN or TCP/IP, USB, GPIB,...) is selected at initialization time with the channel-specific address string ("VISA resource string"). Alternatively, you can define a VISA alias (short name), see Table 11-1 for an overview.

Access via VXI-11 or HiSLIP protocols is achieved from high level programming platforms using VISA as an intermediate abstraction layer. VISA encapsulates the low-level VXI or GPIB function calls and thus makes the transport interface transparent for the user.

A VISA installation is a prerequisite for remote control using the following interfaces:

- LAN Interface using HiSLIP Protocol
- LAN interface using VXI-11 Protocol

Access via the LAN socket protocol can be operated both, with or without the VISA library, see Chapter 11.1.1.4, "Socket Communication", on page 232. For more information about VISA library, refer to the user documentation.

11.2 How to Set up a Remote Control Connection

This section guides you through the steps required to set up remote control connections of the available interfaces.

- Setting Up a Remote Control Connection over LAN Using VXI-11 Protocol
- Setting Up a Remote Control Connection over LAN Using Socket Communication

A remote control program must open a connection to R&S WinIQSIM2, before it can send commands to and receive device responses from R&S WinIQSIM2.

11.2.1 Setting Up a Remote Control Connection over LAN Using VXI-11 Protocol

In this example, the I/O software library R&S VISA from Rohde & Schwarz is used to set up a LAN remote control link and remotely control the R&S WinIQSIM2. R&S VISA is running on the controller PC that can be the same or a different PC. When the connection is set up you can send commands to the instrument, and receive the responses.

A remote control connection requires a VISA installation but no additional hardware on the controller PC. The LAN I/O channel is selected at initialization time using the VISA resource string (also referred to as "address string"). A VISA alias (short name) is used to replace the complete resource string. The host address is either the R&S WinIQSIM2's computer name (hostname) or the IP address. See also Chapter 11.1.1, "LAN Interface", on page 230.

In this example, it is assumed that:

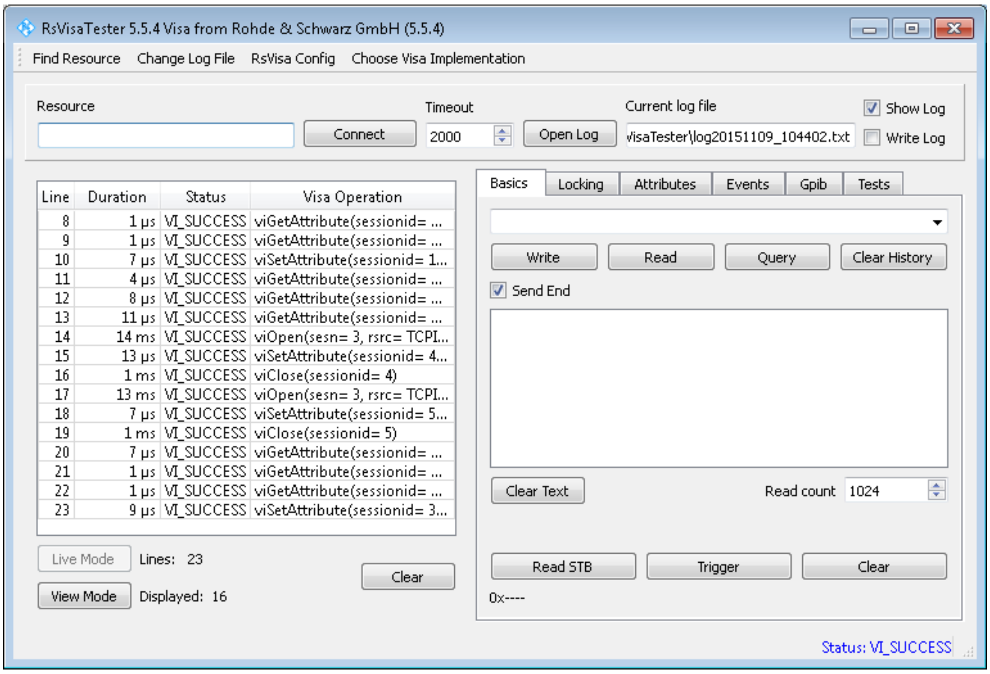
- A LAN remote control link between the controller PC and the R&S WinIQSIM2 is set up.
- The R&S VISA program is installed on the remote PC, see "http://www.rohde-schwarz.com/rsvisa > RS VISA Release Notes".

For detailed information, refer to section "Remote Control Basics" in the user manual or to the online help of the "R&S VISA" program.

To set up the controller with R&S VISA

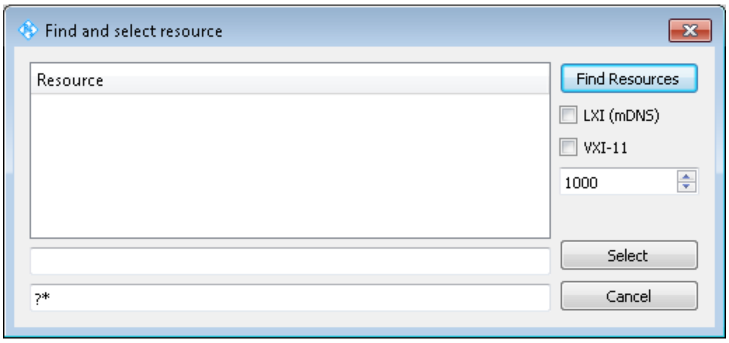
To remote control the R&S WinIQSIM2, we use the "R&S VISA Tester" application. The application communicates via TCP/IP protocol.

1. On the controller, start "R&S VISA > Tester 32bit" or "R&S VISA > Tester 64bit", respectively.



2. In the menu bar, select "Choose VISA Implementation > Rohde & Schwarz Visa".

3. Select "Rohde & Schwarz Visa".
4. Confirm with "OK".
5. In the menu bar, select "Find Resource" to search for the corresponding computer in the LAN.

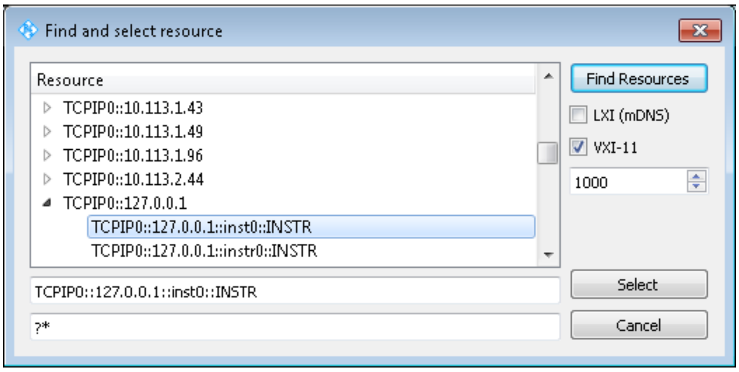


6. Select "VXI-11" and "Find Resources".

   R&S VISA scans the network for all connected devices and lists all detected devices in the "Resource" list.
   Note: The search can take some time, particularly in large networks.

7. Select the IP address of the corresponding PC.

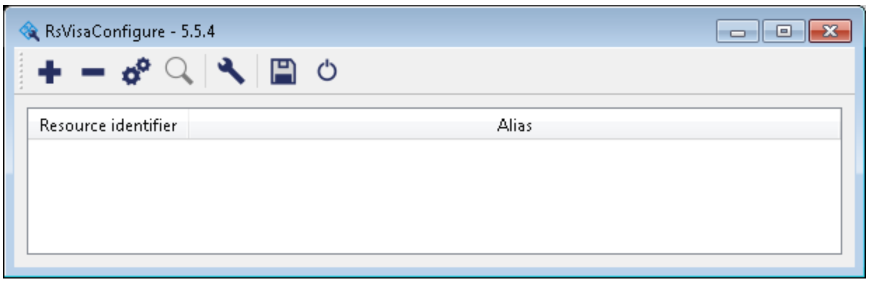
8. Confirm with "Select".



   The "Find and select resource" dialog closes and R&S VISA indicates the IP address in the "Resource" field of the main application window.

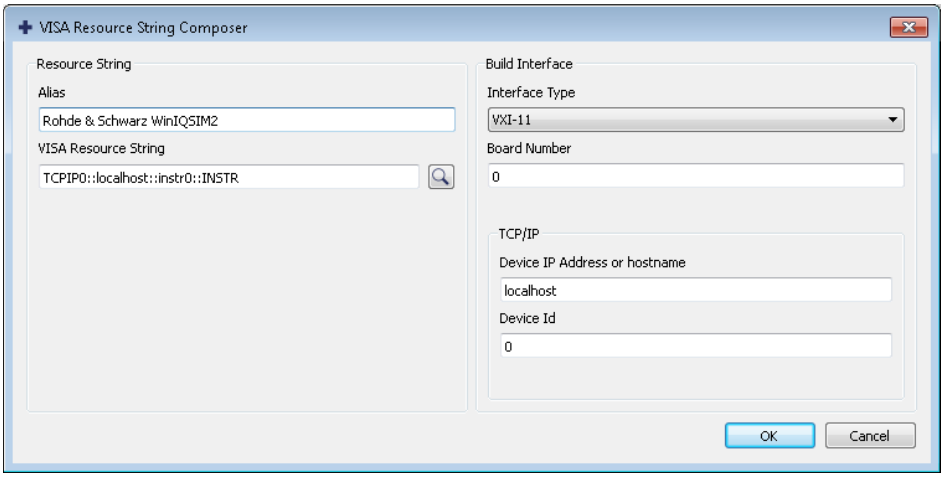
9. As an alternative to the IP address, you can assign an alias name to the R&S WinIQSIM2:

   a) In the menu bar, select "RsVisaConfig".

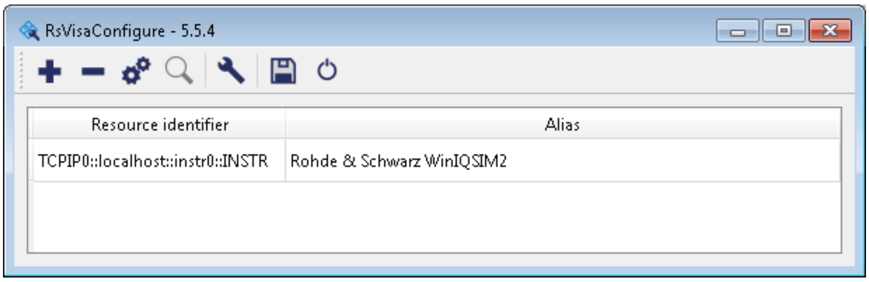


   b) In the toolbar, select "+" to access the "VISA Resource String Composer".
   c) Fill in the "Alias" name, the "VISA Resource String" and the "Device IP Address or host name" as shown in the figure.

d) Confirm with "OK".



The application assigns the "Alias" name.



e) Close the dialog.
The R&S WinIQSIM2 is now registered in the program and can be addressed via the resource string or alias name.

10. In the main window, select "Connect".

    R&S VISA establishes the connection to R&S WinIQSIM2.

    Now you can send settings to configure the instrument and receive its responses.

    Note: If the connection cannot be set up, R&S VISA displays an error in the log view. For information on how to proceed when network failures occur, see Chapter 13.4, "Resolving Network Connection Failures", on page 395.

    For further information on the functions to read and write to an open session, as well as the utility applications the software provides, see the R&S VISA User Manual.

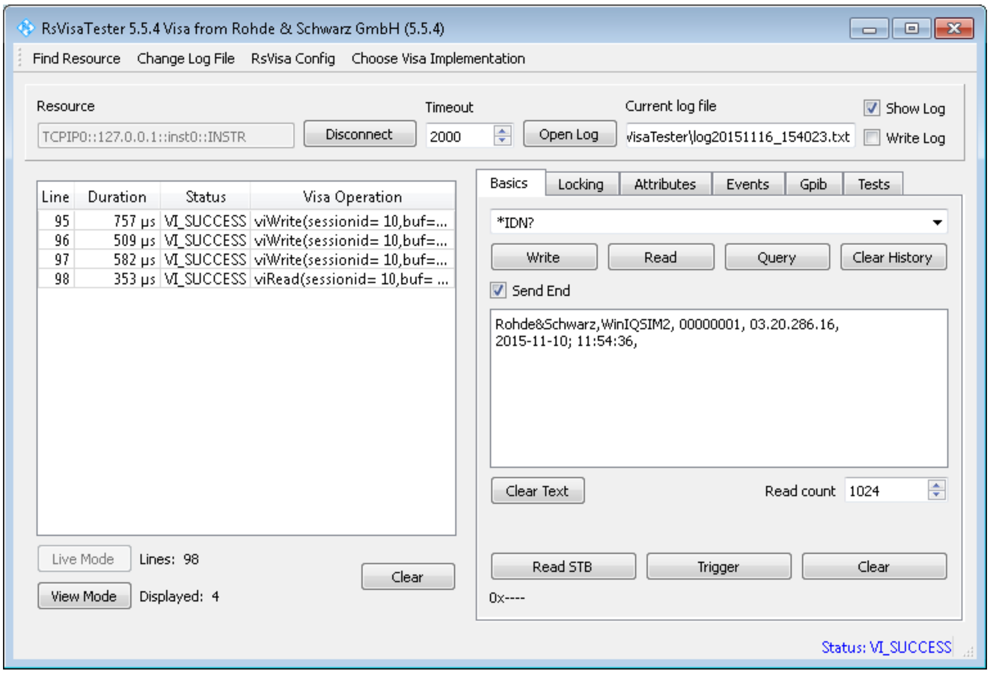
Starting a remote control session over LAN with R&S VISA

To set R&S WinIQSIM2 to remote control, you can use the addressed command `&GTR`, or send any command from the controller.

1. Start the "R&S VISA Tester".
2. Establish the connection to the R&S WinIQSIM2, see "To set up the controller with R&S VISA" on page 234.

3. In the "R&S VISA > Basics" tab, enter an SCPI command, e.g. "*IDN?".
4. Confirm with "Query".

   R&S WinIQSIM2 is switched to remote control when it receives a command from the controller.
5. Select "Read" to obtain the response from R&S WinIQSIM2.



   Tip: If the "Show Log" checkbox is checked, R&S VISA displays each VISA function call in the log-view on the left. If you check the "Write Log" checkbox, the log-view entry is written to the log file as well. You can operate the log-view in two modes: the "Live Mode" shows only the most recent messages whereas the "View Mode" allows you to scroll the history.
6. To activate baseband signal generation, select e.g. `SOUR:BB:W3GP:STAT 1`.
7. Confirm with "Write".
   To check the performed setting, and select "Read".
   Response: `1`.

   Note: R&S WinIQSIM2 does not indicate when it is controlled remotely. To return to manual control, operate R&S WinIQSIM2 directly.

11.2.2 Setting Up a Remote Control Connection over LAN Using Socket Communication

This section provides an example on how to establish a remote control connection over Telnet client, which is embedded in every operating system.

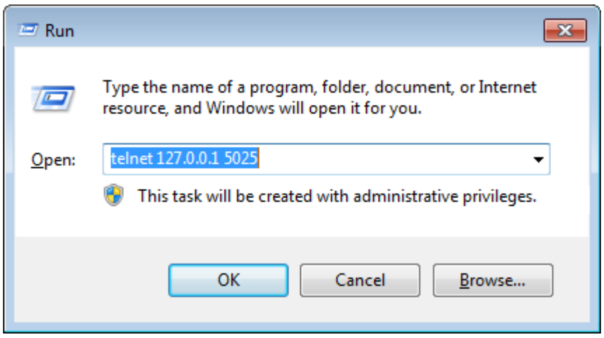
Note that the Telnet client transmits information unencrypted. Therefore, for sensitive information we recommend that you use a client which supports secure protocols, as e.g. SSH.

In the following example, we assume basic knowledge of programming and operation of the controller. You can find information on the interface commands in the corresponding manuals.

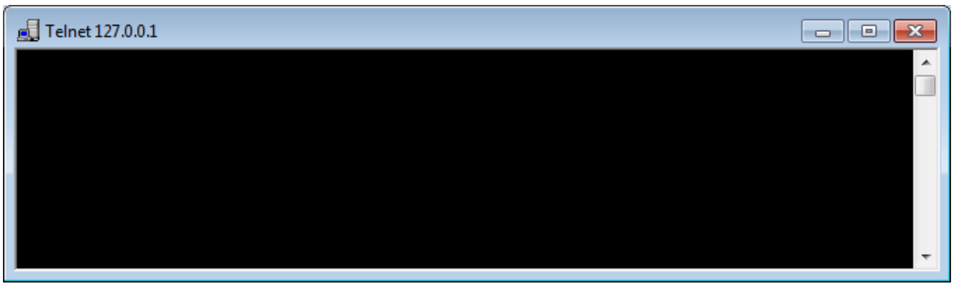
To set up a Telnet connection

To control the software, only a Telnet program is required.

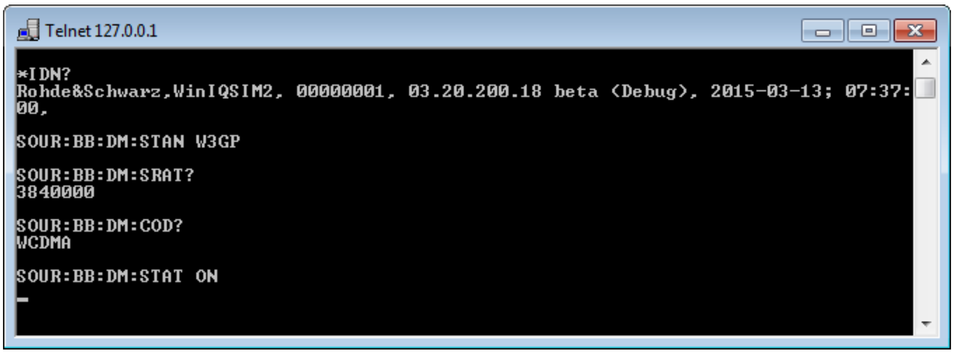
1. To establish a Telnet connection with R&S WinIQSIM2, start the Telnet program.
2. Enter the socket address.
   The socket address is a combination of the IP address or the hostname of R&S WinIQSIM2's PC and the number of the port configured for remote-control via Telnet.
   Tip: R&S WinIQSIM2 uses the port number 5025 for remote connection via Telnet.

   The connection to R&S WinIQSIM2 is set up and you can send remote commands.

3. Insert a command, e.g. `*IDN`.
   Note that Telnet does not reflect your first entry.
4. Confirm with "Enter".
   If you get a response, the connection is working properly. Then the client displays all subsequent inputs and responses.



11.3 Automating Tasks with Remote Command Scripts

To achieve fast configuration, work with complex test setups or with reproducible repeating measurements, you can automate the required settings with remote command scripts. A script contains a series of remote commands that perform the settings sequentially. When completed, it is converted to an executable format, stored in a file and can be run whenever needed.

11.3.1 About the Creation and Use of Remote Command Scripts

For creating remote command scripts, R&S WinIQSIM2 provides a function, which records manual settings in SCPI format.

Creating an SCPI list

You can create an SCPI command list directly in R&S WinIQSIM2 and save the list in a file. If you want to edit or write a script manually, use a suitable editor.

Even if you manually write the command sequence, R&S WinIQSIM2 supports you by displaying the command syntax for a parameter, including the current parameter value, see e.g. "To find out the remote command with "Show SCPI Command" " on page 246, and "To find out the remote command using the "Online Help"" on page 246.

Directly in R&S WinIQSIM2, you can create an SCPI list at any time of operation. Both, automatic or manual SCPI recording is possible. I.e., you can record all the settings that you perform between a start and an end point, or select deliberately the commands you want to record.

- Record all performed steps automatically
  The instrument records the SCPI commands including the setting values of each step you perform, and then writes the commands to the file system, see "To record SCPI lists automatically" on page 244.
  Some parameters cannot be set by an SCPI command. If so, "no SCPI command found" is entered instead of a command.
- Record certain steps manually

In manual recording mode, you can deliberately record an SCPI command with the "Add SCPI command to sequence" function, see "To record SCPI lists manually" on page 244.

- Write a script manually
  To write or edit settings in a script, you must know the exact syntax of the SCPI command and the setting value. R&S WinIQSIM2 helps you, if you need to look up a command:
  – "Show SCPI command" (context-sensitive menu)
    Displays the SCPI command of the selected parameter with the setting value. With the "Copy" function, you can paste the command directly into the script, see "To find out the remote command with "Show SCPI Command" " on page 246.
  – Online help
    Describes each function, and includes a cross-reference to the corresponding SCPI command. The referenced section describes the SCPI command syntax with all setting values and their functionality, see "To find out the remote command using the "Online Help"" on page 246.

  Tip: Conversely, if you are looking for a function in the GUI, which belongs to an SCPI, you find it via the cross-reference in the online help, see "To find out the GUI function, which corresponds to an SCPI command" on page 247.

Displaying a recorded SCPI list

R&S WinIQSIM2 lists the recorded remote commands in the "SCPI Sequence dialog", see Chapter 11.3.2, "SCPI Sequence Settings", on page 241.

Depending on the starting point, you can access the "SCPI Sequence" dialog as follows:

- During recording
  Select "Show current SCPI Sequence" in the context sensitive menu.
- At any time outside recording
  Select "Show last SCPI Sequence" in the context sensitive menu.
  This function implies that at least one recording has been executed after power-on.
- At the end of the recording
  The dialog automatically opens. That is, when you have selected "Stop automatic SCPI recording" or "Stop manual SCPI recording".
- After you have exported the script to a file, select "Show file content" in the "SCPI Recording Export" dialog, see Chapter 11.3.3, "SCPI Recording Export Settings", on page 242.

Checking and revising a recorded SCPI list

It is recommended that you check a recorded list and revise if necessary. It can be that ...

- A parameter has not assigned an SCPI command.
- An element of the user interface has not an assigned parameter.

In these cases, the recording function writes `:SYST:INF:SCPI 'SCPI command not available'` in the list instead.

Such entries are also detected during execution. The instrument recognizes an incomplete command and displays an error message.

See "To check a recorded SCPI list" on page 250 for some suggestions on how you validate the recorded commands.

Creating and exporting an executable script file

When the script list is completed, a code generator translates the SCPI commands into the source code of a proprietary programming language, using a code template. Therefore, each language requires an appropriate code template. When converted, you can store the script in a file with an extension corresponding to the programming language.

The R&S WinIQSIM2 provides the following predefined code templates by default:

- Plain SCPI
  Represents SCPI base format, that is ASCII format, stored as text file (`*.txt`).
- MATLAB
  A programming environment, frequently used in signal processing and test and measurement applications (`*.m`).
  You can directly use this format with MATLAB(c) toolkit. For comprehensive information on this topic, refer to the application note 1GP60: MATLAB Toolkit for R&S Signal Generators.
- NICVI
  An ANSI C programming environment designed for measurements and tests (`*.c`).
  You can directly use this format with "National Instruments LabWindows CVI".

You can also convert a script to a user-specific format. In this case, you need a code template with the extension `*.expcodetmpl`.

For information on how to select the code template and store the script in a file, see "To convert and store SCPI lists" on page 249.

Executing an SCPI script

To run a remote command script, you need a remote control program on the controller PC, which supports the corresponding script format.

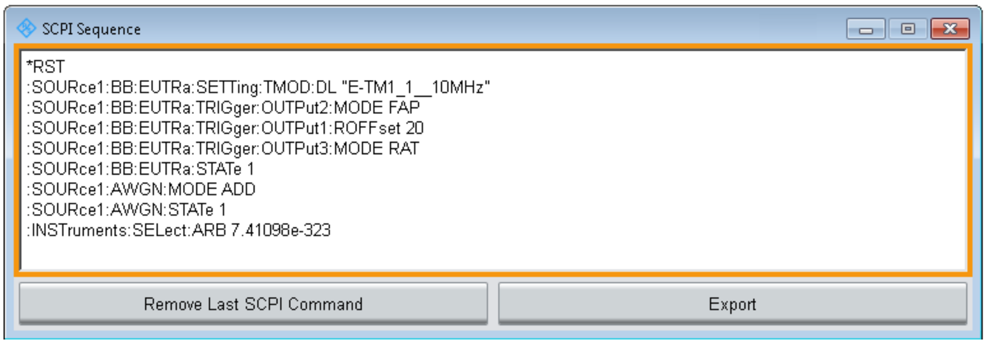
11.3.2 SCPI Sequence Settings

Access:

Depending on the starting point, perform the corresponding step:

1. When recording is in progress:
   In the context-sensitive menu, select "Show current SCPI Sequence".
2. When recording is not running:
   In the context-sensitive menu, select "Show last SCPI Sequence".

3. When you stop recording, the dialog opens automatically.



The "SCPI Sequence" dialog lists the last recorded and exported SCPI command list. It enables you to delete the last recorded command, and you can access the export dialog for storing the list in a file.

Export

Accesses the Chapter 11.3.3, "SCPI Recording Export Settings", on page 242 dialog for configuring the file parameters for export.

Remove last SCPI Command

Deletes the last recorded command when SCPI recording is in progress.

Once the recording is stopped, the entries are internally stored and cannot be changed.

11.3.3 SCPI Recording Export Settings

R&S WinIQSIM2 enables you to store recorded command lists in the "SCPI Recording Export" dialog. It provides access to this dialog directly after a recording, or if you have recorded and stored a command list at a previous time.

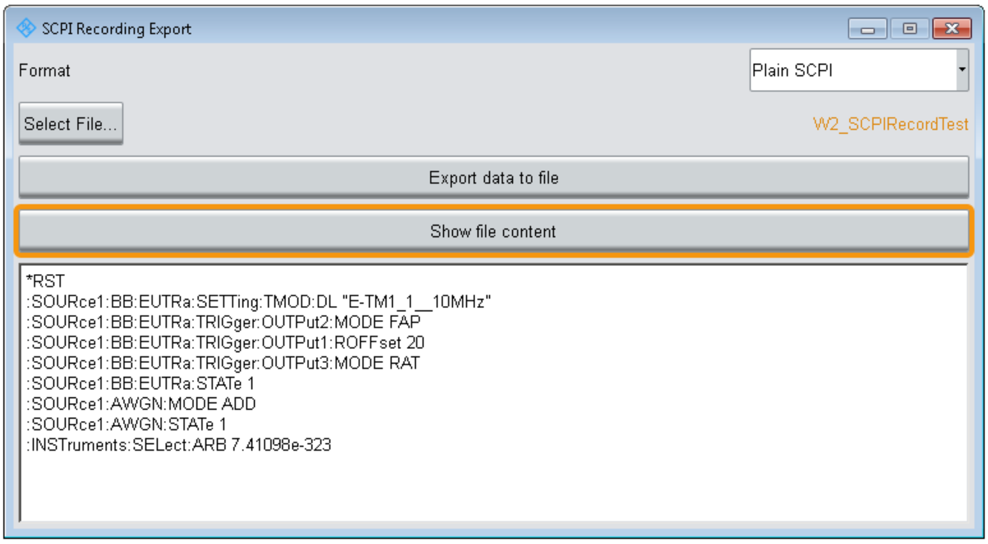
In any case, you can access the dialog only in the "SCPI Sequence" dialog by selecting the "Export" button.

Access:

You can access the dialog only in the "SCPI Sequence" dialog by selecting the "Export" button.

1. To open the "SCPI Sequence" dialog, proceed as described in "Access:" on page 241

2. In the "SCPI Sequence" dialog, select "Export".



The "SCPI Recording Export" dialog contains all functions required for export of command lists to a file. It enables you to select the source code format, assign an individual file name and display the file content.

Format

Selects the source code format for the command list.

"Plain SCPI" Uses SCPI syntax.

"Predefined Code Generator"
Accesses the predefined templates for common source code generators that convert the recorded settings in the programming languages MATLAB or NICVI.

"User Code Generator"
Provides the ability to convert a script by a user-specific code generator.

Select Code Template

Opens the standard "File Select" dialog and lists the predefined or user-defined code templates.

See Chapter 9.3.1, "File Select Settings", on page 213.

Select File

Opens a standard file select dialog "Select Output File", see Chapter 9.3.1, "File Select Settings", on page 213.

Export data to file

Executes data export.

Show file content

Displays the content of the script in the appropriate source code.

11.4 How to Use the SCPI Record Function

This section describes how you record the remote control commands during the manual setting. It contains short instructions how to convert the recorded command lists to executable script files.

To record SCPI lists automatically

This example explains briefly how to proceed when you want to record all performed steps.

For comprehensive information on this topic, refer to the application note 1GP98: SCPI Recorder Test Automation on a Fingertip.

1. On the screen, open the context-sensitive menu with a right mouse click.
2. Select "Start automatic SCPI recording".

   Starting from now, all steps you perform are recorded.
3. To stop SCPI recording, select "context-sensitive menu > Stop automatic SCPI recording".

   The "SCPI Sequence" dialog opens automatically.
4. Proceed with "To check a recorded SCPI list" on page 250.

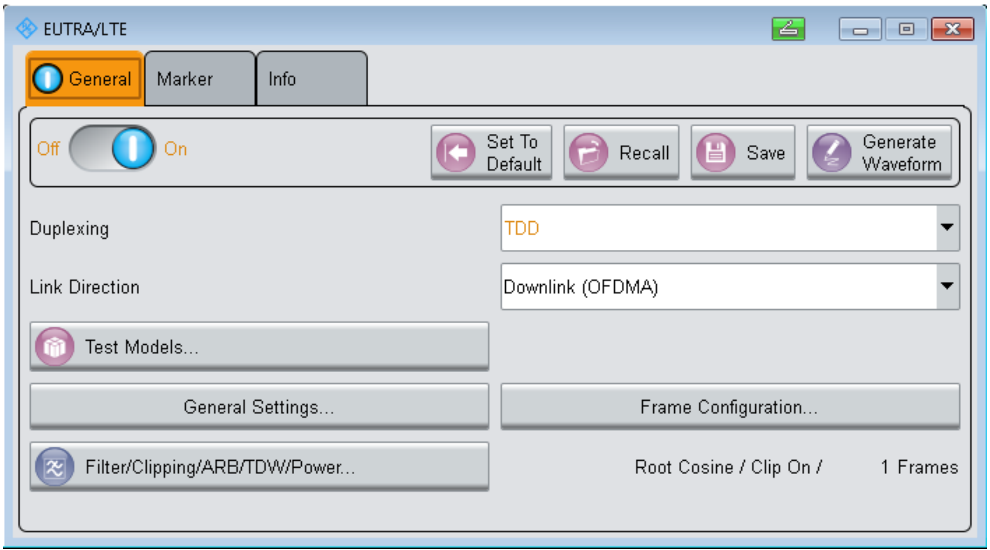
To record SCPI lists manually

Using this function, you can deliberately determine the settings, you need in the script. For this application, it is useful to mark all modified parameters. It helps you to check if you have recorded all necessary settings.

1. To retrace your settings, open the context-sensitive menu.
2. Select "Mark all parameters changed from preset".

   This function identifies all settings you have changed, both in the block diagram, and in the dialogs. They appear orange.



3. Open the context-sensitive menu
4. Select "Start manual SCPI recording".

Now you can selectively record your steps:

a) Set the parameter.
b) Open the context-sensitive menu.
c) Select "Add SCPI command to sequence"

Tip: You cannot see "Add SCPI ..." in the menu?
It happens, if you open the context-sensitive menu outside of a dialog or input field, for example in the block diagram. Open the context-sensitive menu within the corresponding dialog or input field, and the feature is available.

d) Continue with the next setting, and repeat steps *a to b* whenever needed.

Each time you select "Add SCPI ...", the SCPI command is appended to a temporary list.

5. To check the progress of the recording, select "context-sensitive menu > Show current SCPI sequence".

The "SCPI Sequence" dialog opens with the list of all recorded settings so far.

6. To stop SCPI recording, open the context-sensitive menu again.
7. Select "Stop manual SCPI recording".

The "SCPI Recording Export" dialog automatically opens.

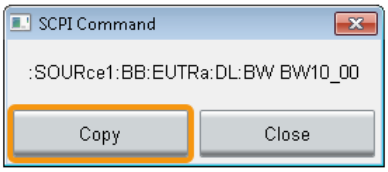
8. Proceed with "To check a recorded SCPI list" on page 250.

To find out the remote command with "Show SCPI Command"

To find out the SCPI command of a parameter in manual operation, perform the following:

1. Select the corresponding parameter.
2. Select "context-sensitive menu > Show SCPI command".

You get the detailed command syntax, including the currently set value.



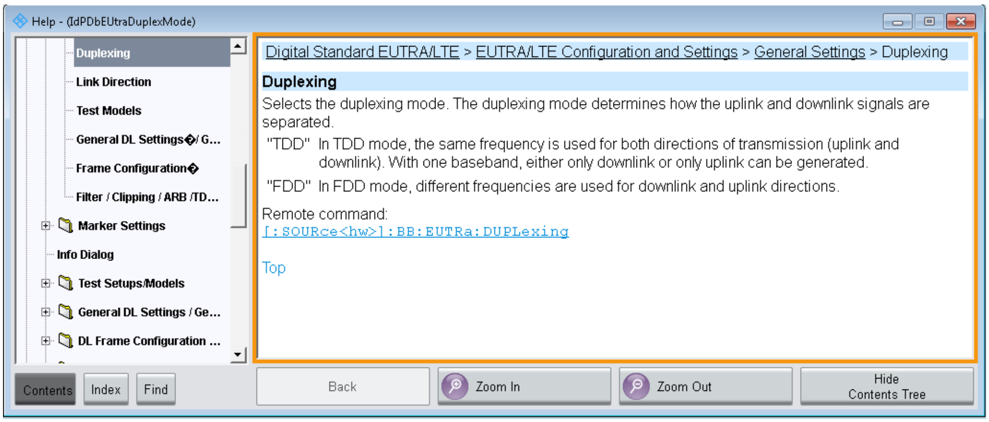
To find out the remote command using the "Online Help"

If you are looking for the remote command to a function in manual operation, you find it in the description of the online help.

1. Select the corresponding parameter.
2. Select "context-sensitive menu > Help".

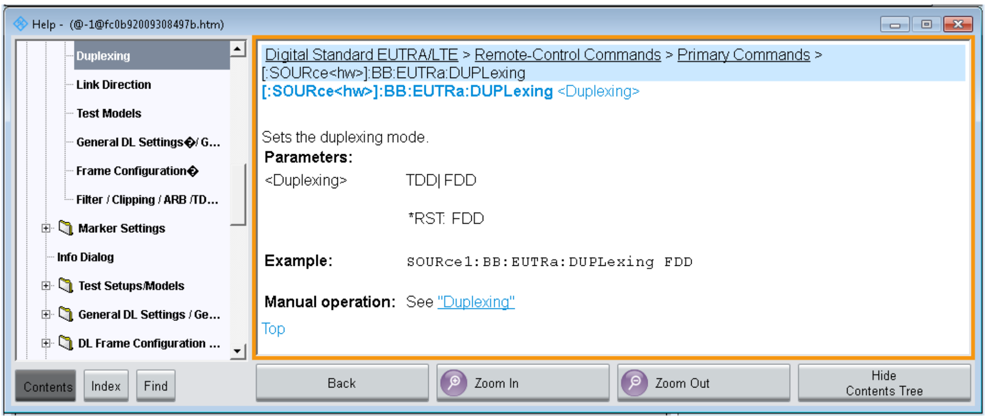
Alternatively, you can open the online help with the "F1" key.

The help topic opens. It describes the function and the corresponding parameter values, if applicable. A link leads you to the description of the remote command.



3. To find out the syntax of the remote command including the parameter values, follow the link.

   The help topic of the SCPI command description opens.

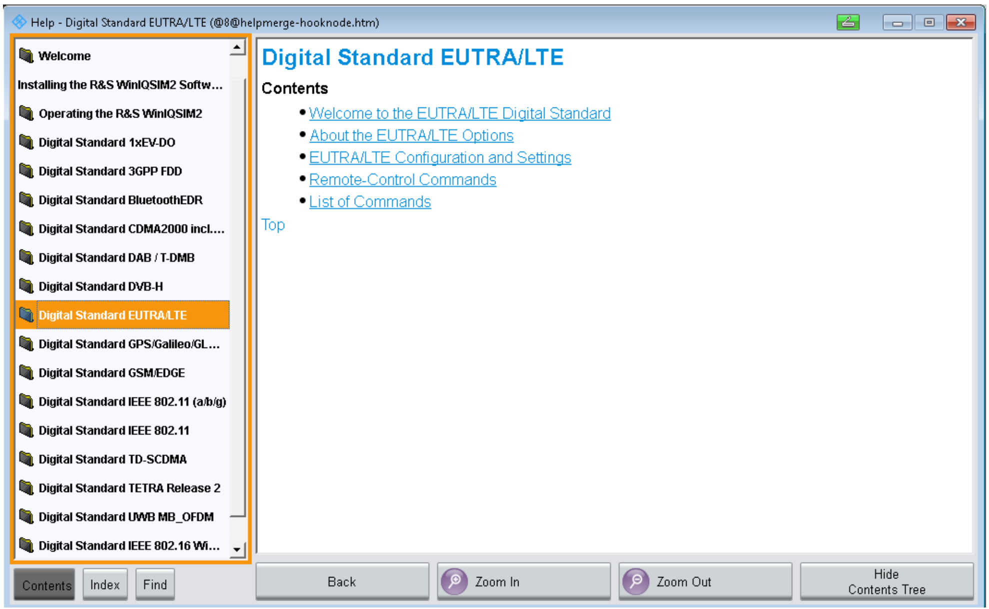


To find out the GUI function, which corresponds to an SCPI command

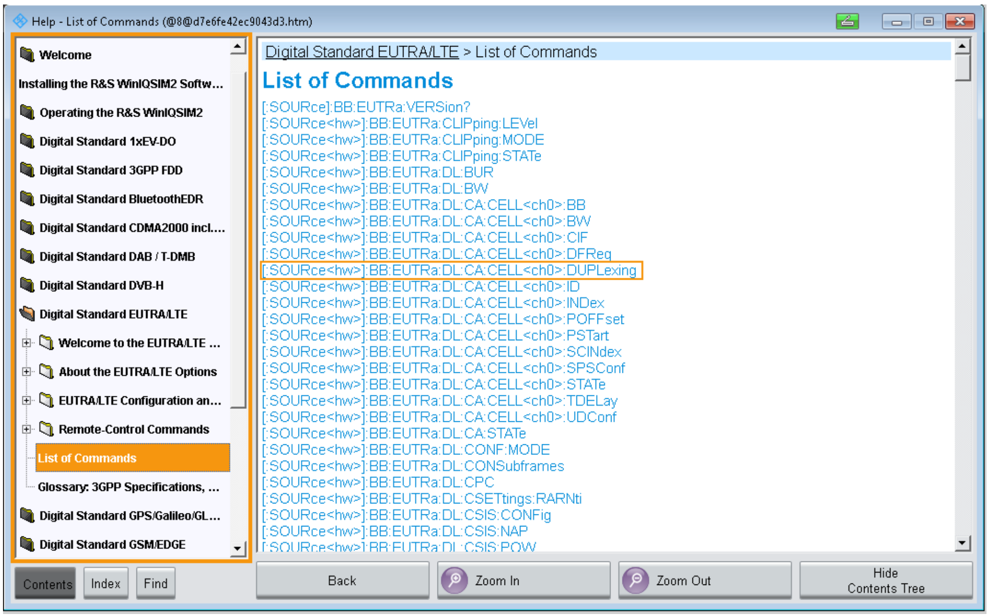
If you have an SCPI command, and look for the associated function in the manual operation (GUI), you find it via the cross-reference in the online help.

1. Select any block or parameter in the graphical user interface of R&S WinIQSIM2.
2. Select "context-sensitive menu > Help".
3. In the navigation panel of the online help, select the section to which the command applies, e.g. "Operation of WinIQSIM", "Digital Standard EUTRA/LTE", etc.

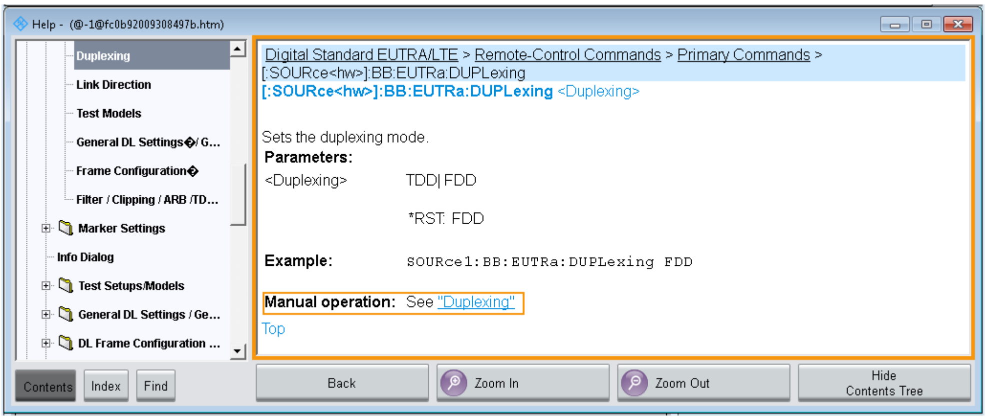
   E.g. you are looking for the function of the command `[SOURce:BB:EUTRa:DL:CELL<ch0>DUPLexing`.



4. Select "List of Commands".



5. Select the command.



6. Follow the link under "Manual Operation:"

   The help topic of the manual operation opens. The breadcrumb navigation on top shows you where this parameter is located.

To convert and store SCPI lists

To convert an SCPI list to the desired format, you need to load the list into the "SCPI Recording Export" dialog.

You can access this dialog only in the "SCPI Sequence" dialog by selecting the "Export" button.

1. To open the "SCPI Sequence" dialog, proceed as described in Chapter 11.3.2, "SCPI Sequence Settings", on page 241.
2. In the "SCPI Sequence" dialog, select "Export".
3. Select the "Format" for the command syntax to which you want to convert and save the list.
4. Select "Export".
5. Select "Select Code Template".
   Note: The code template must be selected before exporting.
   Depending on the selected format, proceed accordingly:
   a) "Plain SCPI"
   b) "Predefined Code Generator"
      The "SCPI Recording Export - Select Predefined Code Template" dialog opens.
      Select the desired predefined code template.
      Confirm with "Select".
   c) "User Code Generator"
      A file system ("SCPI Recording Export - Select User Code Template") dialog opens.
      Select your user-defined code template. The code template must have file extension `*.expcodetmpl`.
6. "Select File..."

The "SCPI Recording Export - Select Output File" dialog opens.

a) Select the directory.
b) Assign a file name for storing the recorded list.
c) Confirm with "Save".

7. "Export data to file"
Stores the recorded data either in ASCII format (plain SCPI), or in the corresponding format of the used code template, and activates the "Show File Content" button.

8. "Show file content"
Displays the source code in the corresponding syntax.

To check a recorded SCPI list

Some suggestions on how you can check and revise a list:

1. Export the list file as described in "To convert and store SCPI lists" on page 249.
2. Open the list file in a suitable editor.
3. Search and remove missing command entries, e.g. `:SYST:INF:SCPI 'SCPI command not available'`.
4. Remove unnecessary content that has been written after a preset.
5. Rearrange the commands to a reasonable order. E.g. if you set a `STATe` command to the last position of a list, you can avoid intermediate calculations of the signal.
6. Preview the list for completeness by comparing it with the modified settings in the manual mode.

   a) To retrace your settings in manual operation, open the context-sensitive menu.
   b) Select "Mark all parameters changed from preset".
   The function identifies all settings you have changed, both in the block diagram, and in the dialogs. They appear orange.
   c) Check whether there is a command in the list for all modified settings.

12 Remote Control Commands

In the following, all remote-control commands will be presented in detail with their parameters and the ranges of numerical values.

12.1 Conventions used in SCPI Command Descriptions

Note the following conventions used in the remote command descriptions:

- **Command usage**
  If not specified otherwise, commands can be used both for setting and for querying parameters.
  If a command can be used for setting or querying only, or if it initiates an event, the usage is stated explicitly.
- **Parameter usage**
  If not specified otherwise, a parameter can be used to set a value and it is the result of a query.
  Parameters required only for setting are indicated as **Setting parameters**.
  Parameters required only to refine a query are indicated as **Query parameters**.
  Parameters that are only returned as the result of a query are indicated as **Return values**.
- **Conformity**
  Commands that are taken from the SCPI standard are indicated as **SCPI confirmed**. All commands used by the R&S WinIQSIM2 follow the SCPI syntax rules.
- **Asynchronous commands**
  A command which does not automatically finish executing before the next command starts executing (overlapping command) is indicated as an **Asynchronous command**.
- **Reset values (*RST)**
  Default parameter values that are used directly after resetting the instrument (`*RST` command) are indicated as ***RST** values, if available.
- **Default unit**
  This is the unit used for numeric values if no other unit is provided with the parameter.
- **Manual operation**
  If the result of a remote command can also be achieved in manual operation, a link to the description is inserted.

12.2 Common Commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect and are employed in the same way on different devices. The headers of these commands consist of "*" followed by three letters. Many common commands are related to the Status Reporting System.

Available common commands:

***CLS**

Clear status

Sets the status byte (STB), the standard event register (ESR) and the `EVENt` part of the `QUEStionable` and the `OPERation` registers to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

Usage: Setting only

***ESE** <Value>

Event status enable

Sets the event status enable register to the specified value. The query returns the contents of the event status enable register in decimal form.

Parameters:

<Value> Range: 0 to 255

***ESR?**

Event status read

Returns the contents of the event status register in decimal form and subsequently sets the register to zero.

Return values:

<Contents> Range: 0 to 255

Usage: Query only

***IDN?**

Identification

Returns the instrument identification.

Return values:

<ID> "Rohde&Schwarz,<device type>,<serial number>,<firmware version>"

Example: `Rohde&Schwarz,WinIQSIM2,1407.6004k02/000000, 3.1.17.1-03.01.158`

Usage: Query only

***IST?**

Individual status query

Returns the contents of 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

***OPC**

Operation complete

Sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request. The query form writes a "1" into the output buffer as soon as all preceding commands have been executed. This is used for command synchronization.

***OPT?**

Option identification query

Queries the options included in the instrument. For a list of all available options and their description refer to the data sheet.

Return values:

<Options> The query returns a list of options. The options are returned at fixed positions in a comma-separated string. A zero is returned for options that are not installed.

Usage: Query only

***PRE** <Value>

Parallel poll register enable

Sets parallel poll enable register to the indicated value. The query returns the contents of the parallel poll enable register in decimal form.

Parameters:

<Value> Range: 0 to 255

***PSC** <Action>

Power on status clear

Determines whether the contents of the `ENABle` registers are preserved or reset when the instrument is switched on. Thus a service request can be triggered when the instrument is switched on, if the status registers ESE and SRE are suitably configured. The query reads out the contents of the "power-on-status-clear" flag.

Parameters:

<Action> 0 | 1

0
The contents of the status registers are preserved.

1
Resets the status registers.

***RCL** <Number>

Recall

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.

It also activates the instrument settings which are stored in a file and loaded using the `MMEMory:LOAD <number>, <file_name.extension>` command.

***RST**

Reset

Sets the instrument to a defined default status. The default settings are indicated in the description of commands.

The command is equivalent to `SYSTem:PRESet`.

Usage: Setting only

***SAV** <Number>

Save

Stores the current instrument settings under the specified number in an intermediate memory. The settings can be recalled using the command `*RCL` with the associated number.

To transfer the stored instrument settings in a file, use the command `:MMEMory:STORe:STATe`.

***SRE** <Contents>

Service request enable

Sets the service request enable register to the indicated value. This command determines under which conditions a service request is triggered.

Parameters:

<Contents> Contents of the service request enable register in decimal form. Bit 6 (MSS mask bit) is always 0.

Range: 0 to 255

***STB?**

Status byte query

Reads the contents of the status byte in decimal form.

Usage: Query only

***TRG**

Trigger

Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal. This common command complements the commands of the `TRIGger` subsystem.

Usage: Event

***TST?**

Self-test query

Initiates self-tests of the instrument and returns an error code

Return values:

<ErrorCode> **integer > 0 (in decimal format)**
An error occurred.
(For details see the Service Manual supplied with the instrument).

0
No errors occurred.

Usage: Query only

***WAI**

Wait to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed and all signals have settled (see also command synchronization and `*OPC`).

Usage: Event

12.3 Waveform, Data and List Format

R&S WinIQSIM2 uses a simple tag-oriented format for externally or internally generated waveforms, data and control lists.

Tag general format

Tags are self-contained information units, enclosed in braces { }. Their general format is `{Name: Data}` or `{Name-Length: Data}`. The colon separates the name part and the data part. The colon can be followed by a space for the sake of legibility.

- The **`Name`** identifies the tag. It is always expressed in capital letters.
- The **`Data`** is tag-specific, and in most cases it is in ASCII plain text.
- The **`Length`** specifies the number of bytes in a `WAVEFORM` tag, `DATA LIST` tag or `EMPTYTAG`
  Length is an ASCII integer value, defining the number of bytes from the colon `:` to the end brace `}`

Rules

Each waveform file must begin with the `TYPE` tag. The sequence of the remaining tags is arbitrary. For each tag an indication shows whether it must be included in the file concerned (mandatory) or may be included (optional).

Unknown tags are not analyzed by an instrument; they are left unchanged and saved without an error message for a possible further read back.

In all examples of file contents listed in this section, the tags have been separated by line breaks for better reading.

12.3.1 Tag Description

This section describe the mandatory `TYPE` tag followed by description of all other tags in an alphabetical order. Most tags are valid for all three file types. If a tag is valid only for a single file type, e.g. only for a waveform, this fact is indicated in the description.

{TYPE: magic, xxxxxxxx}

(mandatory, must be the first tag in the file)

Identifies the file as a valid R&S WinIQSIM2 file. It must be present and must be the first in the waveform. If a file of the same name exists on the target medium, it is overwritten.

Setting parameters:

magic — Designates the file type and has the following values:

SMU-WV
A valid R&S WinIQSIM2 waveform.

SMU-MWV
A valid R&S WinIQSIM2 multi-segment waveform.

SMU-DL
A valid R&S WinIQSIM2 data list.

SMU-CL
A valid R&S WinIQSIM2 control list.

xxxxxxxx

Is an ASCII-coded checksum of the data part of the `WAVEFORM` tag in the file. This value is always 0 for data lists and control lists.
The checksum for waveforms is used for detecting transmission errors. If the `TYPE` tag contains 0 or a non-numeric value for the checksum, it is ignored.
It is calculated in accordance with the algorithm given bellow, where 'start' is a pointer to the first byte after the '#' character in the `WAVEFORM` tag and 'length' is the number of bytes between 'start' and the closing curly bracket (excluding the latter; 'length' must be divisible by 4 without a remainder):

```
UINT32 checksum(void *start, UINT32 length)
{
UINT32 i, result = 0xA50F74FF;
for(i=0; i < length/4; i++)
result = result ^ ((UINT32 *)start)[i];
return(result);
}
```

Example:

`{TYPE: SMU-WV,106656}`
`BB:ARB:WAV:TAG? 'TYPE'`
Queries the content of the `TYPE` tag.
`Response: 'SMU-WV,106656'`
This is a valid waveform.

{CLOCK: frequency}

(mandatory for waveforms)

The tag specifies the clock frequency at which the waveform has to be output, in Hz (on multi-segment waveforms this tag contains the maximal clock of all segments).

A query of `ARB:CLOCk?` after loading the waveform returns the value set using the `CLOCK` tag. This value can later be altered with the command `ARB:CLOCk?`.

Example:

`{CLOCK: 54000000}`
`BB:ARB:WAV:TAG? 'CLOCK'`
Queries the content of the `CLOCK` tag.
`Response: 54000000`
The clock frequency is set to 54 MHz.

Usage: Setting only

{COMMENT: string}

The tag contains a plain text ASCII string of arbitrary length. The string is not analyzed in R&S WinIQSIM2. It is used to describe the file. The string is allowed to contain all printable ASCII characters except the closing curly bracket.

Example: `{COMMENT: File with data for 3GPP enhanced channels}`
`BB:ARB:WAV:TAG? 'COMMENT'`
queries the content of the `COMMENT` tag of the selected waveform file.
`Response: 'File with data for 3GPP enhanced channels'`
the comment on the waveform reads "File with data for 3GPP enhanced channels".

Usage: Setting only

{COPYRIGHT: string}

The tag contains an ASCII string of arbitrary length. The string is not analyzed in R&S WinIQSIM2. It is used to store copyright information about the file content.

Example: `{COPYRIGHT: Rohde&Schwarz}`
`BB:ARB:WAV:TAG? 'COPYRIGHT'`
queries the content of the `COPYRIGHT` tag of the selected waveform file.
`Response: 'Rohde&Schwarz'`
copyright resides with Rohde&Schwarz.

Usage: Setting only

{DATA BITLENGTH: BitLength}

(mandatory for data lists)

The tag contains the length of the data held in the `DATA LIST` tag in bits in ASCII format.

Example: `{DATA BITLENGTH: 444}`
`BB:DM:DLIS:SEL "D:\user\dl"`
`BB:DM:DLIS:TAG? "dl","DATA BITLENGTH"`
queries the content of the `DATA BITLENGTH` tag of the selected data list file.
`Response: '444'`
the data list is 444 bits long.

Usage: Setting only

{DATA LIST-Length: #d0d1...dx...dN-1…}

(mandatory for data lists)

The tag contains the actual bit sequence of the data list in binary format.

Setting parameters:

Length
: Defines the number of bytes in the `DATA LIST` tag in ASCII Format (see `{WAVEFORM-Length: #I0Q0I1Q1...IxQx...IN-1QN-1…}` for detailes).

dx
: Data bits in binary format (8-bit unsigned characters, MSB first).

Example: `{DATA LIST-17: #d0d1...dx...d127}`
16 bytes containing 128 data bits, first bit is the MS bit of the first byte.

Usage: Setting only

{DATE: yyyy-mm-dd;hh:mm:ss}

(optional)

The tag contains the date and time at which the file was created. The year must be expressed as four digits. The instrument does not analyze this tag.

Example: `{DATE: 2009-04-02;14:32:12}`
`BB:ARB:WAV:TAG? 'DATE'`
queries the content of the `DATE` tag of the selected waveform file.
`Response: '2009-04-02;14:32:12'`
the waveform was created on April 2, 2009 at 14 hrs 32 min

Usage: Setting only

{EMPTYTAG-Length: #EmptySequence}

(mandatory in automatically generated one and multi-segment waveforms)

This tag is empty, i.e. contains no data, and is used as placeholder.

Setting parameters:

Length
: An ASCII integer value that specifies the number of bytes in the `EMPTYTAG`, i.e. defines the number of bytes from the colon `:` to the end brace `}`
Note: If you change the content of a waveform file, change also the `{EMPTYTAG-Length}` value. For example, if you add a tag or add bytes to a tag, reduce the length by the number of newly introduced bytes.

EmptySequence
: An empty sequence containing blanks only. The number of used blanks is calculated as the difference between the hex addresses of the `{WAVEFORM}` tag and the hash sign `#` in the `{EMPTYTAG}`.
The `{WAVEFORM}` tag always starts at hex address #4000.

Example:

```
{TYPE:SMU-WV, 837236424}
{COPYRIGHT:2003 Rohde&Schwarz SMU}
{DATE:2012-07-11;14:38:01}
{SAMPLES:80000}
{CLOCK:8666666.666666666}
{VECTOR MAX:1.000000038569158}
{LEVEL OFFS:3.333553817875577e-07,0}
{MARKER LIST 1:0:1;1:0;1249:0}
{MARKER LIST 2:0:1;1:0;1249:0}
{MARKER LIST 3:0:1;1:0;1249:0}
{MARKER LIST 4:0:1;1:0;1249:0}
{EMPTYTAG-15947:# ...}
{WAVEFORM-320017:#IQIQIQ...}
```

The example waveform file contains 436 (0x1b4) bytes before the # sign in the EMPTYTAG; the hex address of the # sign is *0x1b5*. The {WAVEFORM} starts at *0x4000*. The EMPTYTAG contains *15946* blanks and has a length of (15946+1) bytes.

Usage: Setting only

{CONTROL LENGTH: ControlLength}

(optional / recommended for marker and control lists)

The tag specifies the length of *all* control or marker list in ASCII format.

The control length influences the way the marker and control lists are processed, in particular the way traces are repeated; see Figure 12-1.

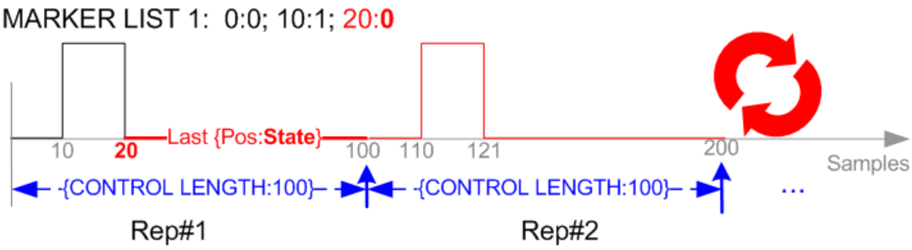


Figure 12-1: Example: Processing of MARKER TRACE if CONTROL LENGHT is specified

If the CONTROL LENGTH tag is not used, the marker and control list length are determined by the last position, that is the last {**Pos**:State} couple, defined in the particular [TRACE] LIST tag; see Figure 12-2.

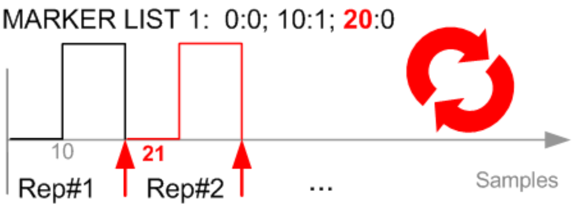


Figure 12-2: Example: Processing of MARKER TRACE if CONTROL LENGHT is not used

To maintain marker signals and waveform data synchronized, set the `CONTROL LENGTH` to be equal to the number of samples specified with the tag SAMPLES.

See also Chapter 12.3.2, "How to Define Periodically Repeating Traces", on page 269.

Example:
`{CONTROL LENGTH: 500}`
`SOURce:BB:ARBitary:CLISt:TAG? 'CONTROL LENGTH'`
Queries the length of the control list.
`Response: 500`

Manual operation: See "Total List Length" on page 83

{LEVEL OFFS: RMSOffset_dB,PeakOffset_dB}

(recommended for waveforms)

The tag determines the level of the ARB signal in the waveform file. The offset levels define the offset of RMS and peak value relative to the 16-bit full scale modulation (-32767 to + 32767) = 0 dB.

Setting parameters:

RMSOffset_dB
Defines the RMS level offset of the signal relative to full scale ARB signal in the `WAVEFORM` tag. The offset is defined in ASCII float format. The value is always positive.
A 3 dB value indicates that the RMS level of the signal is 3 dBs below the full scale.
full scale = max. amplitude of vector of I/Q samples = $|S_{IQ}|$max = sqrt(I^2+Q^2)max = 0 dB

PeakOffset_dB
Defines the peak level offset of the signal relative to full scale for the ARB signal in the `WAVEFORM` tag. The offset is defined in ASCII float format.
The value usually equals 0 dB as usually the I/Q samples (signed 16-bit integer values) are modulated to full scale: Full scale = 0 dB = max. amplitude of vector of I/Q samples = $|S_{IQ}|$ max = sqrt(I^2+Q^2)max = (2^{15})-1 = 32767.
A positive `PeakOffset_dB` value indicates that a headroom to full scale is provided when generating the waveform. A negative PeakOffset_dB value indicates that overrange is likely for some samples, i.e. clipping might occur.
The crest factor can be calculated from the two values as follows:
`Crest Factor = |PeakOffset_dB - RMSOffset_dB|`

Example:
`{LEVEL OFFS: 3.45,2}`
`BB:ARB:WAV:TAG? 'LEVEL OFFS'`
Queries the content of the `LEVEL OFFS` tag of the selected waveform file.
`Response: 3.45,2`
The level of the waveform is below full scale, clipping does not occur.

Usage: Setting only

{SAMPLES: Samples}

(recommended for waveforms)

The tag contains the number of I/Q samples in the waveform in ASCII format.

On multi-segment waveforms, this tag contains the total I/Q samples of all segments.

Example:
`{SAMPLES: 1000}`
`BB:ARB:WAV:TAG? 'SAMPLES'`
Queries the content of the `SAMPLES` tag of the selected waveform file.
`Response: 1000`
The waveform contains 1000 I/Q samples.

Usage: Setting only

See also Chapter 12.3.2, "How to Define Periodically Repeating Traces", on page 269.

{[TRACE] LIST [#]: Pos0:State0; Pos1:State1; ...PosN-1:StateN-1}

(mandatory for control lists / optional for waveforms)

The tag contains the data for the marker and control signals in the control list or the marker signals of ARB waveforms.

Traces are processed different, depending on the selected CONTROL LENGTH.

See also Chapter 12.3.2, "How to Define Periodically Repeating Traces", on page 269.

Setting parameters:

[TRACE] MARKER | BURST | LEVATT | CW MODE | HOP | MAP
Name of the marker or control signal.
For ARB waveforms, it is only meaningful to define marker signals; in the ARB multi-segment waveforms these tags are ignored!

[#] 1 .. four
Sets the marker or control trace number; supported is only LEVATT LIST 1.

Pos Specifies in ASCII format the position (i.e. sample number or data value), with effect from which the binary `State` of the marker or of the control signal changes.

State 0 | 1
Specifies the binary state of the marker or of the control signal from $\mathtt{Pos}_x$ to $\mathtt{Pos}_{x+1}$ exclusive in ASCII format.

Example: {MARKER LIST 1: 0:0;10:1;20:0;30:1}
BB:DM:CLIS:TAG? 'MARKER LIST 1'
Queries the content of the MARKER LIST 1 tag of the selected control list file.
Response: '0:0;10:1;20:0;30:1'
The marker setting for samples 0 to 9 = 0 (low), for 10 to 19 = 1 (high) and for 20 to 29 = 0. From sample 30 onward the marker setting = 1.

Example: {LEVATT LIST 1: 0:0;10:1;20:0;30:1}
BB:DM:CLIS:TAG? 'LEVATT LIST 1'
Queries the content of the LEVATT LIST 1 tag of the selected control list file.
Response: '0:0;10:1;20:0;30:1'
Level attenuation applies to data values 10 to 19 (high) and from data value 30 onward.

Usage: Setting only

Manual operation: See "Select Ramp to Edit" on page 82

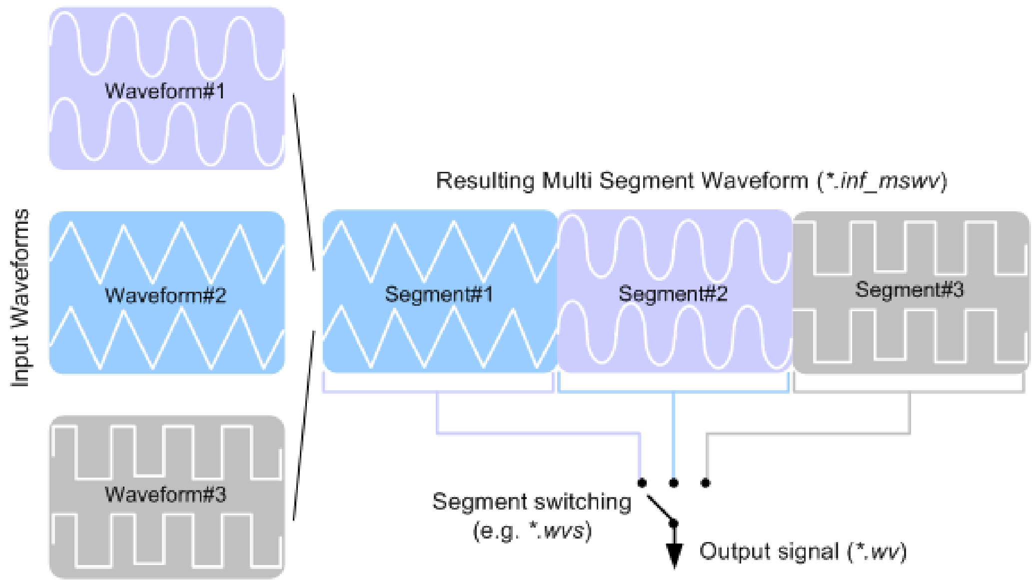
{WAVEFORM-Length: #I0Q0I1Q1...IxQx...IN-1QN-1…}

(mandatory for waveforms)

The tag contains the actual waveform data or multi-segment waveform data (I/Q stream).

Use the TYPE tag to define whether the waveform file contains a normal waveform or a multi-segment waveform.

The following figure shows the principle of multi-segment work mode.



Setting parameters:

Length
: Specifies the number of bytes in a `WAVEFORM` tag and is calculated as follows:
`Length` = `Number of I/Q pairs * 4` (2 bytes per I and 2 bytes per Q value) + `1 byte` (the length of the `#`)

IxQx
: `IxQx`… represents binary data (16-bit signed integer in 2's complement notation) containing the I and Q component alternately and starting with the I component. Each component consists of 2 bytes in Little endian format representation, i.e least significant byte (LSB) first.
The values of the 2 bytes in an I component and a Q component are in the range 0x0 to 0xFFFF (-32767 to +32767). This value is transferred to the D/A converter.
This tag is also used to store multi-segment waveforms. The I/Q streams of the individual waveforms are directly concatenated to one collectively waveform I/Q stream.
The number of segments and the start offset and length of the individual segments inside the total waveform I/Q stream is determined by the additional tags MWV_SEGMENT_COUNT, MWV_SEGMENT_START, and MWV_SEGMENT_LENGTH.
Further `MWV_SEGMENT_...` tags are also available, for example for level and clock information.

Example: **One segment waveform**
{WAVEFORM-401:#I_0,Q_0,I_1,Q_1,I_2,Q_2,...I_{99},Q_{99}}
100 I/Q pairs with 4 bytes each are transmitted - none multi-segment

Example: **Multi-segment waveform**
{WAVEFORM-1201:
#$I_{0,Seg0}$,$Q_{0,Seg0}$,$I_{1,Seg0}$,$Q_{1,Seg0}$,...$I_{99,Seg0}$,$Q_{99,Seg0}$,$I_{0,Seg1}$,$Q_{0,Seg1}$,$I_{1,Seg1}$,$Q_{1,Seg1}$,... $I_{199,Seg1}$,$Q_{199,Seg1}$}
2 segments: segment 0 with 100 I/Q pairs; segment 1 with 200 I/Q pairs. Each I/Q pair consists of 2*16 bit = 4 bytes

Usage: Setting only

{MWV_SEGMENT_COUNT: NumOfSeg}

(mandatory for multi-segment waveforms)

The tag contains the number of segments in the multi-segment waveform in ASCII integer format.

Example: `{MWV_SEGMENT_COUNT: 2}`
Multi-segment waveform with 2 segments

Usage: Setting only

{MWV_SEGMENT_LENGTH: SamplesSeg0, SamplesSeg1, …, SamplesSegN-1}

(mandatory for multi-segment waveforms)

The tag contains a list of I/Q sample lengths for every segment in the multi-segment waveform in ASCII integer format.

Example: `{MWV_SEGMENT_LENGTH: 100,200}`
2 segments: 100 samples in segment 0 and 200 samples in segment 1.

Usage: Setting only

{MWV_SEGMENT_START: SampleStartOffsetSeg0, SampleStartOffsetSeg1, …, SampleStartOffsetSegN-1}

(mandatory for multi-segment waveforms)

The tag contains a list of I/Q sample start offsets for every segment in the multi-segment waveform in ASCII integer format.

Example: `{MWV_SEGMENT_START: 0,100}`
2 segments with 100 samples in segment 0 and 200 samples in segment 1.
The start offset of first segment is 0 samples, start offset of next segment 1 is the sample length of segment 0 = 100 samples.

Usage: Setting only

{MWV_SEGMENT_CLOCK_MODE: Mode}

(mandatory for multi segment waveforms)

The tag contains a string in ASCII format which supplies the clock rate mode, that was used for calculation of the multi segment output waveform (see also "Clock" on page 132).

The tag `CLOCK` contains always the highest clock rate of all segments. The tag `MWV_SEGMENT_CLOCK` contains the clock rates of the individual segments.

Setting parameters:

Mode

UNCHANGED
The segments may have different clock rates; each segment is output with the clock rate defined in its waveform file.

HIGHEST
All segments are output at the highest available clock rate.

USER
All segments are output at the clock rate defined by the the user.
Note: Only upsampling is allowed, no downsampling!

Example: `{MWV_SEGMENT_CLOCK_MODE: UNCHANGED}`

Usage: Setting only

{MWV_SEGMENT_CLOCK: ClockSeg0, ClockSeg1, …, ClockSegN-1}

(mandatory for multi segment waveforms)

The tag contains a list of clock frequencies for every segment in the multi segment waveform in ASCII floating point format.

Example: `{MWV_SEGMENT_CLOCK: 100e6,80e6}`
2 segments: clock of segment 0 is 100 MHz, clock of segment 1 is 80 MHz.
Note: If the segments have different clock frequencies, there are some restrictions on signal output, i.e. seamless switching between segments is only possible, if all segments have the same clock frequency. Software resampling (upsampling) can be used to bring all segments to the same clock.

Usage: Setting only

**{MWV_SEGMENT_LEVEL_OFFS:
RMSOffs_dBSg0,PeakOffs_dBSg0, …, RMSOffs_dBSgN-1, PeakOffs_dBSgN-1}**

(mandatory for multi segment waveforms)

The tag contains a list of level pairs in ASCII floating point format, one pair for every segment in the multi segment waveform. The first value of a level pair defines the rms offset and the second value the peak offset relative to the 16-bit full scale modulation (-32767; + 32767) = 0 dB. The meaning of one level value pair is the same as in the LEVEL OFFS tag for normal waveforms.

Example: `{MWV_SEGMENT_LEVEL_OFFS: 3.0,0.0,6.0,0.0}`
2 segments: RMS level of segment 0 is 3dB below full scale;
RMS level of segment 1 ist 6dB below full scale.
Peak level of both segments is 0 dB full scale.

Usage: Setting only

**{MWV_SEGMENT_FILES:
“FileNameSeg0.wv”, “FileNameSeg1.wv”, …, “FileNameSegN-1.wv”}**

(optional for multi segment waveforms)

The tag contains a list of file names for every segment in the multi segment waveform in ASCII format.

Example: `{MWV_SEGMENT_FILES: “d:\waveforms\sine.wv”, “d:\waveforms\rect.wv”}`

Usage: Setting only

{MWV_SEGMENTx_COMMENT: text}

(optional for multi segment waveforms)

The tag contains a user comment for a specific segment x = [0…NumOfSeg-1] in the multi segment waveform in ASCII format.

Example: `{MWV_SEGMENT1_FILES: segment 1 contains a QPSK signal.}`

Usage: Setting only

{CONTROL LIST WIDTH4–Length: #m0m1…mx…mM-1}

(optional for waveforms and multi segment waveforms)

The tag contains a binary marker element stream, which will be given out synchronously to the I/Q sample sequence. One marker element `mx` consists of 4 bit, which are assigned to the 4 possible marker lines of the instrument (one bit per marker line). One 4 bit marker elements is required for every I/Q sample in the `WAVEFORM` tag. Hence, the number of marker elements `m` should be equal to the number of I/Q samples. The `CONTROL LENGTH` tag has to contain the number of all marker elements `m`.

MSB 7	Byte						LSB 1
Marker element m_x (synchronous to I/Q Sample x)				Marker element m_{x+1} (synchronous to I/Q Sample x+1)			
Marker 4	Marker 3	Marker 2	Marker 1	Marker 4	Marker 3	Marker 2	Marker 1

Figure 12-3: Marker element in 4-bit binary format bit order

For standard waveforms the `MARKER LIST x` tags are a more compact way to define markers, but in prinziple this `CONTROL LIST WIDTH4` format can also be used instead of the `MARKER LIST x` tags.

For multi segment waveforms this `CONTROL LIST WIDTH4` format is required for marker definition. The binary marker streams of the individual segments are directly concatenated (without any gap) to one collectively marker stream.

Setting parameters:

Length — Defines the number of bytes in the `CONTROL LIST WIDTH4` tag in ASCII Format and is calculated as follow:
*Length = Size of "#" (1 byte) + Number of marker elements m_x * (4 bit) / (8 bits/byte)*
The value is rounded up for byte alignment.

mx — Marker element in 4-bit binary format.

Example: {CONTROL LIST WIDTH4-51: #$m_0m_1…m_x…m_{99}$}
100 marker elements, each marker element with 4 bits

Usage: Setting only

12.3.2 How to Define Periodically Repeating Traces

If a marker trace is required that marks for example each frame start, it is sufficient to define the trace ones and repeat it over the length of a waveform. This is useful if you describe a long waveform and a periodical marker is required.

The following examples use marker traces; control lists are processed in the same way.

To define periodical marker trace

The waveform in the example on Figure 12-4 consists of 3 frames, each frame is 100-samples long. The waveform is processed continuously ("Trigger Mode > Auto").

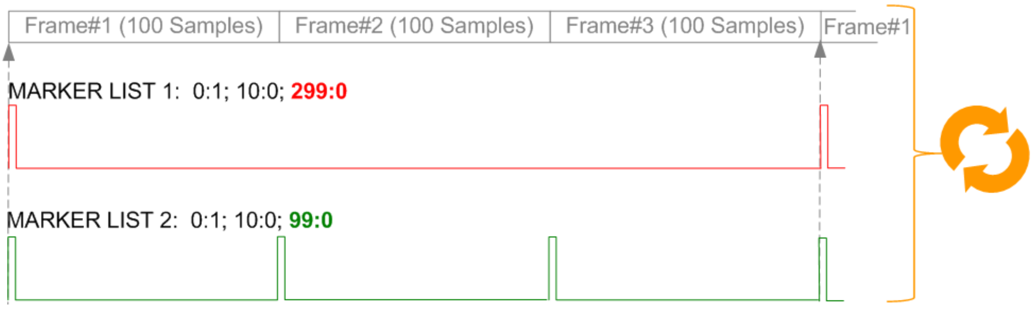


Figure 12-4: Example: Defining marker traces (CONTROL LENGTH tag is not used)

To define a restart marker and a frame start marker, use the following tags:

1. The waveform is 300 samples long, i.e. set {SAMPLES: 300}.
2. Set two [TRACE] LIST tags:
   - For Marker 1 that acts as a restart marker:
     `{MARKER LIST 1: 0:1; 10:0; 299:0}`
   - For Marker 2 that marks each frame start:
     `{MARKER LIST 2: 0:1; 10:0; 99:0}`
3. Do not use the CONTROL LENGTH tag.

   The length of the repeated patterns is determined by the last sample number in the `[TRACE] LIST`, that is the last {**Pos:**State} .

Example: How the CONTROL LENGTH tag influences the processing of the traces

For the example on Figure 12-4, use the same marker traces and set the `CONTROL LENGTH` tag, e.g. `{CONTROL LENGTH: 150}`.

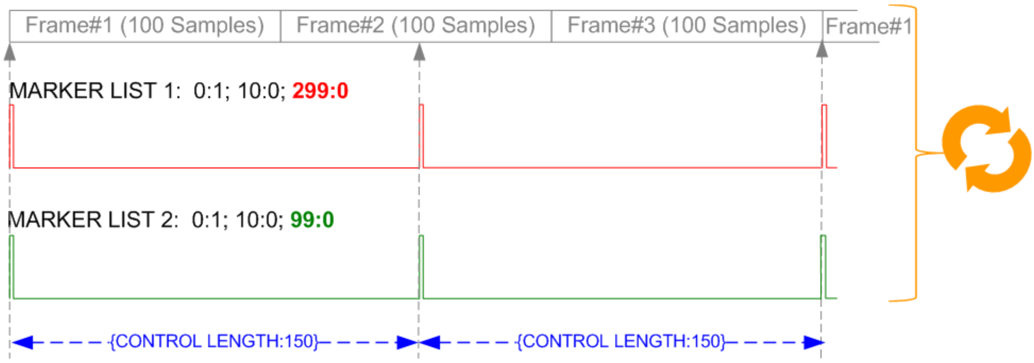


Figure 12-5: Example: Processing of control signals if the CONTROL LENGTH tag is used

The length of **all** control signals is determined by the `CONTROL LENGTH`. Observe how the marker traces are processed. In this example, both marker traces are repeated each 150 samples.

12.3.3 How to Manually Create a Waveform Using Tag File Format

The provided example uses a sine function in the I channel and a cosine function in the Q channel, each with 20 points. The example uses a short program written in the programming language C to calculate the sine and cosine values (see Example "C-program for creating a waveform file" on page 273). They are stored in the file `SICO.txt`. The decimal values in `SICO.txt` are normalized such that they are between –1.0 and +1.0. The data is converted into binary format. The appropriate mandatory tags are added and the data is packed into the `WAVEFORM` tag. As result, the waveform file `SICO.wv` is generated.

This example follows the general principle of creating of a waveform manually, using the tag file format. The Figure 12-6 illustrates this general workflow.

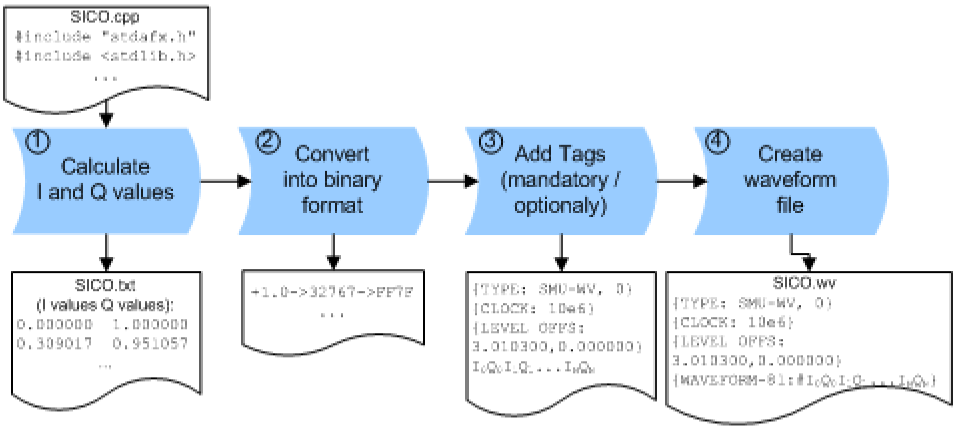


Figure 12-6: Principle of creating a waveform manually

The following steps outline how to create the waveform file `SICO.wv`:

1. Calculate the sine and cosine values, e.g. use the `SICO.cpp` program.

   The result is stored in the file `SICO.txt`.

```
 0.000000   1.000000
 0.309017   0.951057
 0.587785   0.809017
 0.809017   0.587785
 0.951057   0.309017
 1.000000  -0.000000
 0.951057  -0.309017
 0.809017  -0.587785
 0.587785  -0.809017
 0.309017  -0.951057
-0.000000  -1.000000
-0.309017  -0.951056
-0.587785  -0.809017
-0.809017  -0.587785
-0.951056  -0.309017
-1.000000   0.000000
-0.951056   0.309017
-0.809017   0.587785
-0.587785   0.809017
-0.309017   0.951057
```

   Figure 12-7: Contents of SICO.txt: first column Sine (I), second column Cosine (Q)

2. Convert the values from the file `SICO.txt` into binary format consisting of 16-bit signed integer numbers. The numeric range between –1.0 and +1.0 corresponds to the modulation range of the waveform 16-bit D/A converter of -32767 to +32767.

+1.0 ->	32767 ->	= 0x7FFF
0.0 ->	0 ->	= 0x0000
-1.0 ->	-32767 ->	= 0x8001

The Figure 12-8 shows the calculation and conversion steps. The highlighted columns contain the resulting I and Q values represented in Little endian format.

Sample n	deg = 360°/20 * n	I = sin(deg)	$I_{quant,dec}$ = I * FS = I * (2^{15}-1)	$I_{quant,hex}$	$I_{quant,hex}$ (Little endian waveform file representation)	Q = cos(deg)	$Q_{quant,dec}$ = I * FS = I * (2^{15}-1)	$Q_{quant,hex}$	$Q_{quant,hex}$ (Little endian waveform file representation)
0	0	0.000000	0	0000	0000 I_0	1.000000	32767	7FFF	FF7F Q_0
1	18	0.309017	10126	278E	8E27	0.951057	31163	79BB	BB79
2	36	0.587785	19260	4B3C	3C4B	0.809017	26509	678D	8D67
3	54	0.809017	26509	678D	8D67	0.587785	19260	4B3C	3C4B
4	72	0.951057	31163	79BB	BB79	0.309017	10126	278E	8E27
5	90	1.000000	32767	7FFF	FF7F	0.000000	0	0000	0000
6	108	0.951057	31163	79BB	BB79	-0.309017	-10126	D872	72D8
7	126	0.809017	26509	678D	8D67	-0.587785	-19260	B4C4	C4B4
8	144	0.587785	19260	4B3C	3C4B	-0.809017	-26509	9873	7398
9	162	0.309017	10126	278E	8E27	-0.951057	-31163	8645	4586
10	180	0.000000	0	0000	0000	-1.000000	-32767	8001	0180
11	198	-0.309017	-10126	D872	72D8	-0.951057	-31163	8645	4586
12	216	-0.587785	-19260	B4C4	C4B4	-0.809017	-26509	9873	7398
13	234	-0.809017	-26509	9873	7398	-0.587785	-19260	B4C4	C4B4
14	252	-0.951057	-31163	8645	4586	-0.309017	-10126	D872	72D8
15	270	-1.000000	-32767	8001	0180	0.000000	0	0000	0000
16	288	-0.951057	-31163	8645	4586	0.309017	10126	278E	8E27
17	306	-0.809017	-26509	9873	7398	0.587785	19260	4B3C	3C4B
18	324	-0.587785	-19260	B4C4	C4B4	0.809017	26509	678D	8D67
19	342	-0.309017	-10126	D872	72D8	0.951057	31163	79BB	BB79

Figure 12-8: I and Q values calculation and conversion

3. Use an ASCII editor which is able to handle binary data. Create and add the following mandatory tags before this binary data set can be further processed:
   - CLOCK
   - LEVEL OFFS

   An example of the `SICO.wv` file contents could be:

   ```
   {TYPE: SMU-WV, 0}{CLOCK: 10e6}{LEVEL OFFS: 3.010300,0.000000}
   0000FF7F8E27BB79 ... 72D8BB79
   ```

   To simplify the example, the checksum is set to 0, i.e. the instrument does not evaluate a checksum.

   Tip: The tags `TYPE`, `CLOCK`, `LEVEL OFFS` and `WAVEFORM` are mandatory for each waveform. All other tags are optional and can be inserted after the `TYPE` tag in arbitrary order.

4. Pack the binary data into a WAVEFORM tag with the described structure.

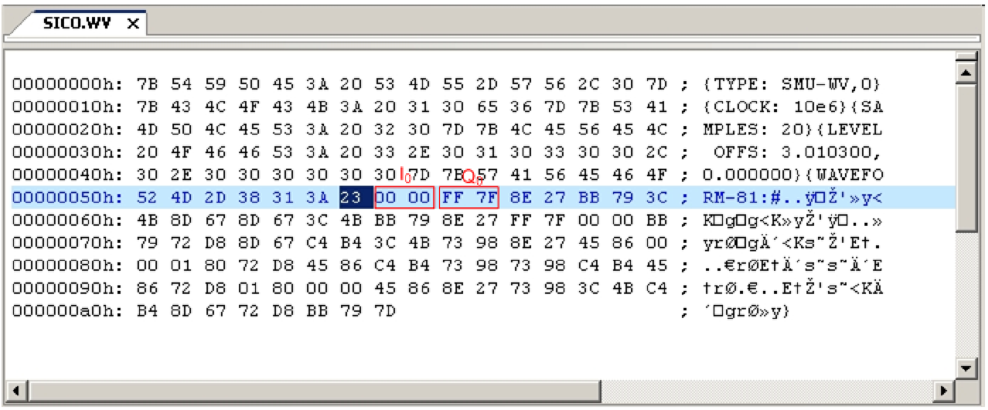
   ```
   {WAVEFORM-Length: #I0Q0I1Q1I2Q2 ... InQn}
   ```

   - Calculate the `Length`
     Length = Number of I/Q pairs * 4 + 1 = 20*4 + 1 = 81 bytes
   - Place the string `{WAVEFORM-81:#` at the beginning of the data set
   - Place the symbol `}` at the end of the data set

   The contents of the waveform file `SICO.wv` for 20 I/Q pairs is now ready for operation and reads:

   ```
   {TYPE: SMU-WV,0}
   {CLOCK: 10e6}
   {LEVEL OFFS: 3.010300,0.000000}
   {WAVEFORM-81:#I0Q0I1Q1...InQn}
   ```

Note: There is no readable representation for binary values in this document. This is why we use the sequence `I0Q0I1Q1...InQn` to characterize the binary code in the present example.
The following figure shows this waveform in a data editor.



Example: C-program for creating a waveform file

C-program `SICO.cpp` for creating the file `SICO.txt` containing 20 sine and cosine pairs, converting them into binary data and creating the waveform file `SICO.wv`.

```
// SICO.cpp
// Defines the entry point for the console application

#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int _tmain(int argc, _TCHAR* argv[])
{
    const unsigned int samples = 20;
    const float pi = 3.141592654f;
    int i;

// SICO.txt
// Creating the file SICO.txt containing 20 sine and cosine pairs
    float grad, rad;
    FILE *fp;
    fp = fopen("SICO.txt", "w");
    if (fp == 0)
        return;
    for (i=0; i<samples; i++)
    {
        grad = (360.0f / (float)samples) * (float)i;
        rad = grad * (pi / 180.0f);
        fprintf(fp, "%f %f\n", sin(rad), cos(rad));
    }
    fclose(fp);
```

```
// SICO.wv
// Generating a binary data set from the I/Q pairs in the file SICO.txt
// and storing the result to file SICO.wv
    FILE *fp_sour, *fp_dest;
    float i_float, q_float;
    unsigned short i_usint, q_usint;
    fp_sour = fopen("SICO.TXT", "rt");
    if (fp_sour == 0)
        return -1;
    fp_dest = fopen("SICO.WV", "wb");
    if (fp_dest == 0)
    {
        fclose(fp_sour);
        return -1;
    }
    // Write required tags to waveform file
    fprintf(fp_dest, "{TYPE: SMU-WV,0}");
    fprintf(fp_dest, "{CLOCK: 10e6}");
    fprintf(fp_dest, "{SAMPLES: %d}", samples);
    // RMS, Peak
    fprintf(fp_dest, "{LEVEL OFFS: %f,%f}", -1.0f * 20.0f * log10(1.0f/sqrt(2.0f)), 0.0f);
    fprintf(fp_dest, "{WAVEFORM-%d:#", (samples * 4) + 1);
    for (i=0; i<samples; i++)
    {
      // Read I/Q pair from ASCII file
         if (fscanf(fp_sour, "%f %f", &i_float, &q_float) == EOF)
            break;
        // Convert I/Q pair to unsigned short
        i_usint = (unsigned short)floor((i_float * 32767.0) + 0.5);
        q_usint = (unsigned short)floor((q_float * 32767.0) + 0.5);
        // Write converted I/Q pair to waveform file
        fwrite(&i_usint, 2, 1, fp_dest);
        fwrite(&q_usint, 2, 1, fp_dest);
    }
    fprintf(fp_dest, "}");
    fclose(fp_dest);
    fclose(fp_sour);
return 0;
}
```

12.3.4 How to Create a Control List Using Tag File Format

R&S WinIQSIM2 provides the following ways to create a file containing control signals:

- to use the dedicated "Control Data Editor" and create a file in ASCII format and with extension `*.dm_iqc`, see "To create a control list in ASCII format manually" on page 86

- to use the tag-oriented format and create a control list file, see "To create a control list using tag file format" on page 275
- to use SCPI commands and create a file in binary format, see "To create a control list in binary format" on page 275

To create a control list using tag file format

To create an ASCII control list file directly, use the provided tag commands.

1. Use a hex data editor and create the **mandatory** tags:
   - TYPE
   - [TRACE] LIST

   The `[TRACE] LIST` tag defines the individual markers or control traces in a combined `{Pos:State}` way within the control list period (`CONTROL LENGTH`).
2. Use a hex data editor and create the **recommended** tag CONTROL LENGTH.

   This tag defines the *periodicity* of the total control list
3. Add the required optional tags.
   They can be inserted after the `TYPE` tag in arbitrary order.

   An example of the control list file contents could be:
   ```
   {TYPE:SMU-CL}{COPYRIGHT:Rohde&Schwarz}
   {DATE:2012-06-11;15:00:09}{HOP LIST:0:0;498:1;506:0}
   {CW MODE LIST:0:0;380:1}{LEVATT LIST 3:0:0;464:1}
   {BURST LIST:0:0;122:1;270:0;582:1;924:0}
   {MARKER LIST 4:0:0;706:1;764:0}
   {MARKER LIST 3:0:0;530:1;633:0}
   {MARKER LIST 2:0:0;350:1;457:0}
   {MARKER LIST 1:0:0;108:1;160:0}
   {CONTROL LENGTH:1000}
   ```
   The Figure 4-6 shows the representation of the created control list in the "Control Data Editor".
   Compare the displayed ramp values of "Marker 1" and the "Total List Length" with the values in the corresponding tags.
   Note: In the provided example, the tags have been separated by line breaks for better reading.
   See also Example "How to assign and activate control signals from a control list" on page 276.

To create a control list in binary format

Generation of a control list in binary format is not necessary but possible.

► Use the commands `BB:DM:CLISt:...` to generate a control list in binary format (see Chapter 12.9.2, "DM Lists", on page 340 and the example in `[:SOURce<hw>]:BB:DM:CLISt:DATA` on page 343).

  See also Example "How to assign and activate control signals from a control list" on page 276.

Example: How to assign and activate control signals from a control list

Note: Irrespective of the way they are created, generated control lists are not automatically used.

We assume, that a control list `clist.dm_iqc` containing information on marker 2, burst gate and level attenuation control signals is created and stored in the directory `D:\user\temp\`.

The following example shows how to enable R&S WinIQSIM2 to:

- use the control list for a particular marker output, e.g. the "Custom Digital Modulation > Marker 2".
- use the Burst Gate and Level Attenuation control signals as defined in a control list.

```
MMEM:CDIRectory "/var/user/temp"
SOURce1:BB:DM:CLISt:CATalog?
// Response: clist
SOURce1:BB:DM:CLISt:SELect "clist"
SOURce1:BB:DM:TRIGger:OUTPut2:MODE CLISt

SOURce1:BB:DM:PRAMp:SOURce INTernal
```

12.3.5 How to Create a Data List Using Tag File Format

R&S WinIQSIM2 provides the following ways to create a data list file:

- to use the dedicated "Data List Editor" and create a file with extension `*.dm_iqd`, see "To create a data list manually" on page 88.
- to use the tag-oriented format and create a data list file, see "To create a data list using tag file format" on page 276.
- to use SCPI commands and create a file in binary format, see "To create a data list in binary format" on page 277.

To create a data list using tag file format

To create a data list file directly, use the provided tag commands.

► Use a hex data editor and create the mandatory tags, the TYPE, the {DATA BITLENGTH} and {DATA LIST-Length}.
Consider the rules of the required tags.

An example of the data list file contents could be:
```
{TYPE:SMU-DL}{COPYRIGHT:Rohde&Schwarz}
{DATE:201-06-11;15:00:09}
{DATA BITLENGTH: 8}
{DATA LIST-2: #01110101}
```
Note: In the provided example, the tags have been separated by line breaks for better reading.
See Figure 4-7 for representation of the created data list in the "Data List Editor".
See also Example "How to assign and activate a data list" on page 277.

To create a data list in binary format

► Use the commands `BB:DM:DLISt:...`, see Chapter 12.9.2, "DM Lists", on page 340.

```
MMEM:CDIRectory "/var/user"
// create a new data list file
SOURce1:BB:DM:DLISt:SELect "dl_new"
// append data to the data list and query the content
:FORMat ASCii
SOURce1:BB:DM:DLISt:DATA:APPend 0,1,1,1,0,1,0,1
SOURce1:BB:DM:DLISt:DATA?
// Response: 0,1,1,1,0,1,0,1
```

See also Example "How to assign and activate a data list" on page 277.

Example: How to assign and activate a data list

Note: Irrespectively on the way they are created, generated data lists are not automatically used.

We assume, that a data list `dl.dm_iqd` is created and stored in the directory `D:\user\`.

The following example shows how to enable R&S WinIQSIM2 to use this data list as data source for the custom digital modulation.

```
MMEM:CDIRectory "/var/user"
SOURce1:BB:DM:DLISt:CATalog?
// Response: dl
SOURce1:BB:DM:DLISt:SELect "dl"
```

12.3.6 Editing Waveform Files, Data and Control Lists

You can edit the internally and externally crated waveform files, data and control lists. The waveform, data and control lists files contain binary and ASCII data.

Consider the following rules while editing files with binary data.

Rules for editing binary data (waveforms, data and control lists)

- **Use hex data editor**
  Always use a hex data editor to edit files containing binary data.
  Editing of binary data file with a text editor, even if you only change the ASCII part of the file, corrupts the file.
- **Adapt the length information in the `{EMPTYTAG}`**
  If you change the content of a waveform file, change also the `{EMPTYTAG-Length}` value.
  For example, if you add a tag or add bytes to an existing tag, reduce the length information by the number of newly introduced bytes.

12.4 MMEMory Subsystem

The MMEMory subsystem (**M**ass **Mem**ory) contains the commands for managing files and directories as well as for loading and storing complete instrument settings in files.

The various drives can be selected using the "mass storage unit specifier " <msus>. The internal hard disk is selected with `D:\`, and a memory stick which is inserted at the USB interface is selected with `E:\`. The resources of a network can also be selected with <msus> in the syntax of the respective network, for example using the UNC format (Universal Naming Convention): `\\server\share`.

The default drive is determined using the command `MMEMory:MSIS <msus>`.

The `C:` drive is a protected system drive. This drive should not be accessed. Reconstruction of the system partition will not be possible without loss of data.

Use the command `:SYSTem:MMEMory:PATH:USER?` to query the path of the directory for user-defined data.

12.4.1 File Naming Conventions

To enable files in different file systems to be used, the following file naming conventions should be observed.

The file name can be of any length and no distinction is made between uppercase and lowercase letters.

The file and the optional file extension are separated by a dot. All letters and numbers are permitted (numbers are, however, not permitted at the beginning of the file name). If possible, special characters should not be used. The use of the slashes "\" and "/" should be avoided since they are used in file paths. A number of names are reserved for the operating system, e.g. `CLOCK$`, `CON`, `AUX`, `COM1...COM4`, `LPT1...LPT3`, `NUL` and `PRN`.

In the R&S WinIQSIM2 all files in which lists and settings are stored are given a characteristic extension. The extension is separated from the actual file name by a dot (see Chapter A.2, "Extensions for User Files", on page 419 for an overview of the file types).

The two characters "*" and "?" function as "wildcards", meaning they are used for selecting several files. The "?" character represents exactly one character, while the "*" character represents all characters up to the end of the file name. "*.*" therefore stands for all files in a directory.

When used in conjunction with the commands, the parameter `<file_name>` is specified as a string parameter with quotation marks. It can contain either the complete path including the drive, only the path and the file name, or only the file name. The file name must include the file extension. The same applies for the parameters `<directory_name>` and `<path>`.

Depending on how much information is provided, either the values specified in the parameter or the values specified with the commands `MMEM:MSIS` (default drive) and `MMEM:CDIR` (default directory) are used for the path and the drive settings in the commands.

Before the instrument settings can be stored in a file, they have to be stored in an intermediate memory using common command `*SAV <number>`. The specified number is subsequently used in the `:MMEMory:STORe:STATe` on page 286 command. Also, subsequently to loading a file with instrument settings with command `:MMEMory:LOAD:STATe` on page 284, these settings have to be activated with the common command `*RCL <number>`.

12.4.2 Examples

In these examples, the current instrument setting is stored in the file `test1.savrcl` in the directory `D:\user\`.

Storing and Loading Current Settings

1. Store the current setting in an intermediate memory with the number 4. This setting can be called using command `*RCL` and the associated number of the memory, for example `*RCL 4`.
   `*SAV 4`
2. To store the settings in a file in a specific directory, specify the complete path.
   `MMEM:STOR:STAT 4,"D:\user\test.savrcl"`
3. To store the settings in a file in the default drive, set the default drive and specify only the file name.
   `MMEM:MSIS 'D:\user\'*SAV 4`
   `MMEM:STOR:STAT 4,"test.savrcl"`
4. Load the file `test.savrcl` in the `user` directory.
   `MMEM:LOAD:STAT 4,'D:\user\test.savrcl'`
5. Activate the instrument setting of the file `test.savrcl`.
   `*RCL 4`

Working with Files and Directories

1. Read out all files in the specified directory.
   `MMEM:CAT? 'E:\user'`

   Response: `127145265,175325184,"test,DIR,0","temp,DIR,0", "readme.txt,ASC,1324","state.savrcl,STAT,5327", "waveform.wv,BIN,2342"`

   the directory `E:\user` contains the subdirectories `test` and `temp` as well as the files `readme.txt`, `state.savrcl` and `waveform.wv` which have different file types.

   Tip: To query only the subdirectories of the current or specified directory, perform:

`MMEM:DCAT? 'E:\user'`

Response: `'test', 'temp'`

To query only the number of subdirectories in the current or specified directory, perform:

`MMEM:DCAT:LENG? 'E:\user'`

Response: `2`

2. To query the number of files in the current or specified directory, perform:
   `MMEM:CAT:LENG? 'E:\user'`
   Response: `3`
3. Create a new subdirectory for mass memory storage in the specified directory.
   `MMEM:MDIR 'E:\new'`
4. Copy the file `state` to a new file.
   `MMEM:COPY 'D:\user\state.savrcl','E:\new'`
5. Rename the file `state`.
   `MMEM:MOVE 'state.savrcl','state_new.savrcl'`
6. Remove the `test` directory.
   `MMEM:RDIR 'E:\test'`

12.4.3 Remote Control Commands

:MMEMory:CATalog? <path>

Returns the content of a particular directory.

Query parameters:

<path> string

String parameter to specify the directory.
If you leave out the path, the command returns the contents of the directory selected with `:MMEMory:CDIRectory`.
The path may be relative or absolute.

Return values:

<UsedDiskSpace> Byte size of all files in the directory.

<FreeDiskSpace> Remaining disk space in bytes.

<FileInfo> <NameFileN>,<SuffixFileN>,<SizeFileN>

List of files, separated by commas

<NameFileN>
Name of the file.

<SuffixFileN>
Type of the file. Possible suffixes are: ASCii, BINary, DIRectory

<SizeFileN>
Size of the file in bytes.

Example: See "Working with Files and Directories" on page 279.

Usage: Query only

Manual operation: See "Directory, File List and File Name" on page 211

:MMEMory:CATalog:LENGth? <Path>

Returns the number of files in the current or in the specified directory.

Query parameters:

<Path> string

String parameter to specify the directory. If the directory is omitted, the command queries the content of the current directory, queried with `:MMEMory:CDIRectory` command.

Return values:

<FileCount> integer

Number of files.

Example: See "Working with Files and Directories" on page 279.

Usage: Query only

:MMEMory:CDIRectory <Directory>

Changes the default directory for mass memory storage. The directory is used for all subsequent `MMEM` commands if no path is specified with them.

Parameters:

<Directory>	<directory_name> String containing the path to another directory. The path can be relative or absolute. To change to a higher directory, use two dots '..' .

Example: See "Working with Files and Directories" on page 279.

Usage: SCPI confirmed

Manual operation: See "Directory, File List and File Name" on page 211

:MMEMory:COPY <SourceFile>[,<DestinationFile>]

Copies an existing file to a new file. Instead of just a file, this command can also be used to copy a complete directory together with all its files.

Setting parameters:

<SourceFile>	string String containing the path and file name of the source file
<DestinationFile>	string String containing the path and name of the target file. The path can be relative or absolute. If `<DestinationFile>` is not specified, the `<SourceFile>` is copied to the current directory, queried with the `:MMEMory:CDIRectory` command. **Note:** Existing files with the same name in the destination directory are overwritten without an error message.

Example: See "Working with Files and Directories" on page 279.

Usage: Setting only
SCPI confirmed

Manual operation: See "Cut, Copy&Paste and Delete" on page 219

:MMEMory:DATA <Filename>, <BinaryBlock>
:MMEMory:DATA? <Filename>

The setting command writes the block data `<BinaryBlock>` to the file identified by `<Filename>`.

Set the GPIB-bus terminator to `EOI` to ensure correct data transfer.

The query command transfers the specified file from the instrument to the GPIB-bus and then on to the controller. It is important to ensure that the intermediate memory on the controller is large enough to take the file. The setting for the GPIB-bus terminator is irrelevant.

Tip: Use this command to read/transfer stored instrument settings or waveforms directly from/to the instrument.

Parameters:

<BinaryBlock> #<number><length_entry><data>

`#`: Hash sign; always comes first in the binary block
`<number>`: the first digit indicates how many digits the subsequent length entry has
`<length_entry>`: indicates the number of subsequent bytes
`<data>`: binary block data for the specified length.
For files with a size with more than nine digits (gigabytes), the instrument allows the syntax `#(<Length>)`, where `<Length>` is the file size in decimal format.

Parameters for setting and query:

<Filename> string

String parameter to specify the name of the file.

Example: `MMEMory:DATA 'D:\user\test.txt',#15hallo`
Writes the block data to the file `test.txt`.
The digit `1` indicates a length entry of one digit; the digit `5` indicate a length of the binary data (`hallo`) in bytes.
`MMEMory:DATA? 'D:\user\test.txt'`
Sends the data of the file `test.txt` from the instrument to the controller in the form of a binary block.
`Response: #15hallo`

Usage: SCPI confirmed

:MMEMory:DCATalog? <path>

Returns the subdirectories of a particular directory.

Query parameters:

<path> String parameter to specify the directory. If the directory is omitted, the command queries the content of the current directory, queried with `:MMEMory:CDIRectory` command.

Return values:

<Catalog> <file_entry>

Names of the subdirectories separated by colons. The first two strings are related to the parent directory.

Example: See "Working with Files and Directories" on page 279.

Usage: Query only

:MMEMory:DCATalog:LENGth? [<Path>]

Returns the number of subdirectories in the current or specified directory.

Query parameters:

<Path> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be queried with `:MMEMory:CDIRectory` command.

Return values:

<DirectoryCount> integer

Number of parent and subdirectories.

Example: See "Working with Files and Directories" on page 279.

Usage: Query only

:MMEMory:DELete <Filename>

Removes a file from the specified directory.

Setting parameters:

<Filename> string

String parameter to specify the name and directory of the file to be removed.

Example: See "Working with Files and Directories" on page 279.

Usage: Event
SCPI confirmed

Manual operation: See "Cut, Copy&Paste and Delete" on page 219

:MEMory:HFRee?

Returns the used and available memory in Kb.

Return values:

<TotalPhysMemKb> integer

Total physical memory.

<ApplicMemKb> integer

Application memory.

<HeapUsedKb> integer

Used heap memory.

<HeapAvailableKb> integer

Available heap memory.

Usage: Query only

:MMEMory:LOAD:STATe <SavRclStateNumb>, <file_name>

Loads the specified file stored under the specified name in an internal memory.

After the file has been loaded, the instrument setting must be activated using an `*RCL` command.

Setting parameters:

<SavRclStateNumb> Determines to the specific <number> to be used with the `*RCL` command, e.g. `*RCL 4`.

<file_name> String parameter to specify the file name with extension `*.savrcl`.

Example: See "Storing and Loading Current Settings" on page 279.

Usage: Setting only

Manual operation: See "Recall" on page 212

:MMEMory:MDIRectory <Directory>

Creates a subdirectory for mass memory storage in the specified directory. If no directory is specified, a subdirectory is created in the default directory. This command can also be used to create a directory tree.

Setting parameters:

<Directory> string

String parameter to specify the new directory.

Example: See "Working with Files and Directories" on page 279.

Usage: Event

Manual operation: See "Create New Directory" on page 219

:MMEMory:MOVE <SourceFile>, <DestinationFile>

Moves an existing file to a new location or, if no path is specified, renames an existing file.

Setting parameters:

<SourceFile> string

String parameter to specify the name of the file to be moved.

<DestinationFile> string

String parameters to specify the name of the new file.

Example: See "Working with Files and Directories" on page 279.

Usage: Event
SCPI confirmed

Manual operation: See "Rename " on page 219

:MMEMory:MSIS <Msis>

Defines the drive or network resource (in the case of networks) for instruments with windows operating system, using `msis` (MSIS = Mass Storage Identification String).

This setting applies to all `MMEMory` commands that do not explicitly specify the drive.

Parameters:

<Msis> string

*RST: D:\

Example: See "Storing and Loading Current Settings" on page 279.

Usage: SCPI confirmed

:MMEMory:RDIRectory <Directory>

Removes an existing directory from the mass memory storage system. If no directory is specified, the subdirectory with the specified name is deleted in the default directory.

Setting parameters:

<Directory> string

String parameter to specify the directory to be deleted.

Example: See "Working with Files and Directories" on page 279.

Usage: Event

:MMEMory:STORe:STATe <savrcl_state_nr>, <file_name>

Stores the current instrument setting in the specified file.

The instrument setting must first be stored in an internal memory with the same number using the common command `*SAV`.

Setting parameters:

<savrcl_state_nr> Corresponds to the specific `<number>` defined with the `*SAV` command, e.g. `*SAV 4`.

<file_name> String parameter to specify the file name with extension `*.savrcl`.

Example: See "Storing and Loading Current Settings" on page 279.

Usage: Event

Manual operation: See "Save" on page 212

:SYSTem:MMEMory:PATH:USER?

Queries the user directory, that means the directory the instrument stores user files on.

Return values:

<PathUser> string

Example: `SYSTem:MMEMory:PATH:USER?`
Response: "D:\user\

Usage: Query only

12.5 Transmission Commands

This chapter contains the remote control commands to set waveform generation, instrument configuration and transmission of the file containing the waveform.

:GENerate:WAVeform:DURation?

Queries the signal duration in seconds of the available samples within the generated waveform file. The value of this parameter is evaluated by parameter `GENerate:WAVeform:SAMPles`.

Return values:

<Duration> float

Example: `GEN:WAV:DUR?`
queries the signal duration in seconds.

Usage: Query only

Manual operation: See "Total Signal Duration" on page 173

:GENerate:WAVeform:OSAMpling?

Queries the baseband oversampling factor. The value is determined by the content of the generated waveform file. To adjust the oversampling rate, the corresponding parameters defined within the standards must be used, e.g. `SOUR:BB:WLAN:FILT:OSAM`.

Return values:
<Osampling> float

Example: `GEN:WAV:OSAM?`
queries the oversampling factor.

Usage: Query only

Manual operation: See "Oversampling Factor" on page 173

:GENerate:WAVeform:SAMPles?

Queries the number of available samples within the generated waveform file. It equals to the I and respectively Q sample sequence length and is calculated as follow:

Number of Samples = Number of symbols * Oversampling factor

Return values:
<Samples> float

Example: `GEN:WAV:SAMP?`
queries the number of available samples.

Usage: Query only

Manual operation: See "Sample Rate (fc)" on page 173

:GENerate:WAVeform:SRATe?

Queries the sample rate in [Hz], where

Sampling Rate = Number of symbols * Oversampling factor.

Return values:
<Srate> float

Example: `GEN:WAV:SRAT?`
queries the sample rate.

Usage: Query only

Manual operation: See "Sample Rate (fc)" on page 173

:INSTruments:GPIB:ADDRess <Address>

Defines the GPIB addresses of the instruments in the device list.

Parameters:
<Address> <Address#1>,<Address#2>,...
Range: 1 to 30

Example: `INST:GPIB:ADDR 8,9,10`
sets the GPIB addresses of the first three devices in the list.

:INSTruments:NAME <Name>

Defines the symbolic name of the instruments in the device list.

Parameters:

<Name> <SymbName#1>,<SymbName#2>,...

Example: `INST:NAME 'MYAFQ','MYOTHERAFQ','MYSMU'`
Defines the symbolic name of the instruments in the device list.

:INSTruments:REMote:CHANnel <Channel>

Defines the remote channels of the instruments in the device list.

Parameters:

<Channel> <Channel#1>,<Channel#2>,...
List of the remote channel (LAN | GPIB | USB) used by the individual instruments

Example: `INST:REM:CHAN USB,LAN,GPIB,LAN`
Sets the hardware channels of the first four devices in the list.

:INSTruments:REMote:NAME <Name>

Defines the instrument (device) names of the instruments in the device list. The instrument name of an instrument is used to access the instrument over the network.

Parameters:

<Name> <InstrName#1>,<InstrName#2>,...

Example: `INST:REM:NAME 'RSAFQ100A1000001',`
`'RSAFQ100A1000002','RSSMU200A1000025'`
defines the network names of the three devices in the list.

:INSTruments:SCAN <Scan>

Triggers a scan function to search for instruments connected via one of the possible interfaces, LAN, GPIB or USB.

Parameters:

<Scan> 0 | 1 | OFF | ON

Example: `INST:SCAN ON`
starts a scan.
`INST:SCAN?`
`Response: 1`
the scan process is still running
`Response: 0`
the scan process is completed and the list of available instruments can be queried (`INST:NAME?`)

Manual operation: See "Scan" on page 189

:INSTruments:SELect:ARB <Arb>

Selects one of the instruments available in the network as an ARB instrument

Parameters:

<Arb> float

Range: 0 (=undefined) to Number of defined devices
Increment: 1
*RST: 0

Example:

`INST:SCAN ON`
starts a scan of the network
`INST:SCAN?`
Queries whether the scan process is still running
`Response: 0`
the scan process is completed, i.e. the list of available instruments can be queried
`INST:NAME?`
Queries the list of the available instruments
`Response: "AFQ100AB10","SMUB10"`
`INST:SEL:ARB 2`
Selects the second instrument in the list (in this case the instrument with name SMUB10) as an ARB instrument.
`INST:SEL:ARB 0`
Selects an "Undefined" instrument

:INSTruments:SELect:VECTor <Vector>

Selects one of the instruments available in the network as a vector signal generator.

Parameters:

<Vector> float

Range: 0 (=undefined) to Number of defined devices
Increment: 1
*RST: 0

Example: `INST:SCAN ON`
starts a scan of the network
`INST:SCAN?`
Queries whether the scan process is still running
`Response: 0`
the scan process is completed, i.e. the list of available instruments can be queried
`INST:NAME?`
Queries the list of the available instruments
`Response: "SMW200A","SMBV100A",AFQ100AB10", "SMUB10"`
`INST:SEL:VECT 2`
Selects the second instrument in the list (in this case the instrument with name SMUB10) as an ARB instrument.
`INST:SEL:VECT 0`
Selects an “Undefined” instrument

:INSTruments:TYPE <Type>

Defines the instrument type in the device list, e.g., R&S AFQ, R&S SMU, R&S AMU, etc.

Parameters:
<Type> AFQB10 | AFQB11 | SMUB9 | SMUB10 | SMUB11 | SMJB9 | SMJB10 | SMJB11 | SMJB50 | SMJB51 | SMATEB9 | SMATEB10 | SMATEB11 | AMUB9 | AMUB10 | AMUB11 | CMW | AFQ100BB12 | AFQ100BB11 | SMBVB10 | SMBVB50 | SMBVB51 | SMBVB10B55 | SMBVB50B55 | SMBVB51B55

Example: `INST:TYPE AFQB10,AMUB11,SMUB10`
defines the device type of one of the three devices in the list.

:INSTruments:USB:SERial <Serial>

Defines the serial numbers of the instruments in the device list. The serial number of an instrument is required to access it over USB.

Parameters:
<Serial> <SerialNumb#1>,<SerialNumb#2>,...

Example: `INST:USB:Serial 100001,100002,100025`
defines the USB serial number of the three devices in the list.

:TRANsmit:AUTO:PATH <Path>

Defines on which path of the instrument the signal is to be generated, if a transmitted waveform file shall be automatically loaded and started on the instrument.

Parameters:
<Path> A | B

Example: `TRAN:AUTO:PATH A`
defines the generation path of the signal.

:TRANsmit:AUTO[:STATe] <State>

Defines whether the transmitted waveform file should be automatically loaded and started on the instrument after the waveform file has been transmitted.

Parameters:
<State> 0 | 1 | OFF | ON

Example: `TRAN:AUTO:STATe On`
defines the automatically loading of the waveform.

:TRANsmit:COMMent <Comment>

Specifies a comment which will be written into the header of generated waveform file.

Parameters:
<Comment> string

Example: `TRAN:COMM 'This is a comment'`
sets a comment.

:TRANsmit:DESTination <Destination>

Defines the transmission destination of the waveform file. There are two options: The Instrument (arbitrary signal generator) or a file on a local drive.

Parameters:
<Destination> INSTrument | FILE

Example: `TRAN:DEST INST`
sets the transmission destination to the instrument.

:TRANsmit:DESTination:IFILe <Ifile>

Defines the name of the waveform destination file on the instrument, if `INSTrument` is selected as the transmit destination.

Parameters:
<Ifile> string

Example: `TRAN:DEST INST`
sets the transmission destination to the instrument.
`TRAN:DEST:IFIL 'D:\3GPP.WV'`
sets the name of the destination file on the instrument.

:TRANsmit:DESTination:LFILe <Lfile>

Defines the name of the waveform destination file on a local hard disc (file copy), if `FILE` is the selected transmit destination.

Parameters:

<Lfile> string

Example: `TRAN:DEST FILE`
sets the transmission destination to file.
`TRAN:DEST:LFIL 'c:\temp\3gpp.wv'`
sets the destination of the waveform file to the local PC hard disc.

:TRANsmit:SOURce <Source>

Defines the source of the waveform to be transmitted. The waveform file can be generated internally or generated internally and stored locally.

Parameters:

<Source> INTernal | FILE

Example: `TRAN:SOUR INT`
Sets the generation of the waveform file to internal. The waveform is not stored.

:TRANsmit:SOURce:LFILe <Lfile>

Defines the name of a source file, that is stored locally and is to be transmitted

Parameters:

<Lfile> string

Example: `TRAN:SOUR:LFIL 'c:\temp\3gpp.wv'`
sets the name of a source file.

:TRANsmit:STATe <State>

Starts/stops a transmission. The query informs about the current state of a transmission.

Parameters:

<State> 0 | 1 | OFF | ON

Example: `TRAN:STAT ON`
starts a transmission.

12.6 SOURce:AWGN Subsystem

The `SOURce:AWGN` subsystem contains the commands for setting the noise generator.

These commands require option R&S WinIQSIM2-K262 (Additive White Gaussian Noise).

[:SOURce<hw>]:AWGN:BRATe <BRate>

Sets the bit rate which is used for calculation of bit energy to noise power ratio from carrier/noise ratio for Digital Standard signals.

Valid units are bps, kbps and mabps as well as b/s, kb/s and mab/s.

Parameters:

<BRate> float

Range: 400 bps to 250E6 bps
Increment: 0.001
*RST: 100000

Example: `SOURce1:AWGN:BRATe?`
queries the bit rate which is used for calculation of the Eb/N0 value from the C/N value.

Manual operation: See "Bit Rate" on page 157

[:SOURce<hw>]:AWGN:BWIDth <BWidth>

Sets the system bandwidth. The noise signal at the level which corresponds to the specified carrier/noise ratio is generated in the bandwidth specified here.

This command is available for modes Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`).

Parameters:

<BWidth>	float Range: 1.0E3 to 100.0E6 Increment: 0.1E3 *RST: 3.84E6

Example:	`SOURce:AWGN:MODE ADD`
	`SOURce:AWGN:BWIDth 10 MHz`
	sets a system bandwidth of 10 MHz.

Manual operation:	See "System Bandwidth" on page 154

[:SOURce<hw>]:AWGN:BWIDth:NOISe?

This command is available for modes In Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`) modes, queries the real noise bandwidth.

Return values:

<Noise>	float Range: 0 to 200E6 Increment: 100 *RST: 0
Example:	`AWGN:BWID:NOIS?` queries the noise bandwidth.
Usage:	Query only
Manual operation:	See "Noise Bandwidth" on page 155

[:SOURce<hw>]:AWGN:BWIDth:RATio <Ratio>

In Additive Noise and Noise Only (`SOUR:AWGN:MODE ADD|ONLY`) modes, sets the ratio of minimum real noise bandwidth to system bandwidth.

The overall bandwidth is calculated as follows and may not exceed the total bandwidth specified in the data sheet:

The value range of this parameter is automatically adjusted depending on the selected instrument. The maximum ratio minimum noise bandwidth to system bandwidth is determinate by the max sample clock of the selected instrument and is calculated so that the overall bandwidth, i.e. "System BW" x "Minimum Noise"/"System BW Ratio" has to be less or equal to the sampling rate of the selected instrument.

Parameters:

<Ratio>	float Range: 1 to Max Increment: 0.1 *RST: 1
Example:	`AWGN:BWID:RAT 2` sets a minimum noise/system bandwidth ratio of 2.
Manual operation:	See "Min. Noise/System Bandwidth Ratio" on page 155

[:SOURce<hw>]:AWGN:CNRatio <CnRatio>

In Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`) mode, sets the carrier/interferer ratio. The value range depends on the selected AWGN mode.

Parameters:

<CnRatio> float

Range: -50 to 40
Increment: 0.01
*RST: 0
Default unit: dB

Example: `AWGN:CNR 10`
sets a carrier/noise ratio of 10 dB.

Manual operation: See "Carrier/Noise Ratio, Carrier/Interferer Ratio" on page 157

[:SOURce<hw>]:AWGN:ENRatio <EnRatio>

In Additive Noise mode, sets the ratio of bit energy to noise power density.

For **Digital Standard** signals, the bit rate used for calculation of E_b/N_0 value from C/N value is defined with command `SOUR:AWGN:BRAT`.

Parameters:

<EnRatio> float

Range: -50 to depends on hardware
Increment: 0.01
*RST: 15.84
Default unit: dB

Example: `AWGN:ENR 10`
sets a ratio of bit energy to noise power density of 10 dB

Manual operation: See "E_b/N_0" on page 157

[:SOURce<hw>]:AWGN:FREQuency:RESult?

In CW inerferer mode, queries the actual frequency of the sine.

The resulting "CW Frequency Offset" is the correction of the desired value based on generating an integer multiple of periods out of the output sequence length.

Return values:

<Result> float

Range: -40E6 to 40E6
Increment: 0.01
*RST: 0

Example: `AWGN:FREQ:RES?`
queries the actual frequency of the interfering sine

Usage: Query only

Manual operation: See "Resulting CW Frequency Offset" on page 155

[:SOURce<hw>]:AWGN:FREQuency:TARGet <Target>

Sets the desired frequency of the sine in `AWGN:MODE CW` mode.

The range of this parameter is limited to ± (Symbol Rate + Sample Rate)/4

Parameters:

<Target> float

Range: -40E6 to 40E6
Increment: 0.01
*RST: 0

Example: `AWGN:FREQ:TARG 2kHz`
sets a frequency of 2 kHz for the interfering sine.

Manual operation: See "Target CW Frequency Offset" on page 155

[:SOURce<hw>]:AWGN:MODE <Mode>

Selects the mode for generating the interfering signal.

Parameters:

<Mode> ONLY | ADD | CW

ADD
The AWGN noise signal is added to the baseband signal.

ONLY
The pure AWGN noise signal is modulated to the carrier. The connection to the baseband is interrupted.

CW
The sine interfering signal is added to the baseband signal.

*RST: ADD

Example:
```
SOURce:AWGN:MODE ONLY
SOURce:AWGN:STATe ON
```
selects the generation of a pure noise and activates it

Manual operation: See "Mode" on page 154

[:SOURce<hw>]:AWGN:POWer:CARRier <Carrier>

available for Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`) modes

Parameters:

<Carrier> float

Increment: 0.01
*RST: 0
Default unit: dBm

Example: `AWGN:POW:CARR?`
queries the carrier power

Manual operation: See "Carrier Power" on page 157

[:SOURce<hw>]:AWGN:POWer:MODE <Mode>

In Additive Noise (`SOUR:AWGN:MODE ADD`) mode, selects the mode for setting the noise power.

Parameters:

<Mode> CN | SN | EN

CN|SN
The noise power is set on the basis of the value entered for the carrier/noise or signal/noise ratio (`SOURce:AWGN:CNRatio|SNRatio`).

EN
The noise power is set on the basis of the value entered for the ratio of bit energy to noise power density (`AWGN:ENR`).

*RST: CN|SN

Example: `SOUR:AWGN:POW:MODE CN`
the noise power is set on the basis of the value entered for the carrier/noise ratio (`AWGN:CNR`).

Manual operation: See "Set Noise Power Via" on page 156

[:SOURce<hw>]:AWGN:POWer:NOISe <Noise>

Sets or queries the noise power in the system bandwidth depending on the selected mode:

- `SOUR:AWGN:MODE ADD|CW`
  Queries the noise/interferer power which is derived from the entered S/N value.
- "Noise Only" mode
  The command sets the noise power.

Parameters:

<Noise> float

Increment: 0.01
Default unit: dBm

Example: `SOURce:AWGN:POWer:NOISe?`
queries the noise power in the system bandwidth.
Response: `10`
the noise power in the system bandwidth is 10 dBm.

Manual operation: See "Noise Power (System Bandwidth) / Interferer Power" on page 157

[:SOURce<hw>]:AWGN:POWer:NOISe:TOTal?

In Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`) modes, queries the noise power in the total bandwidth.

Return values:

<Total> float

Range: -145 to 20
Increment: 0.01
*RST: -30

Example: `SOURce:AWGN:POWer:NOISe:TOTal?`
queries the noise power in the total bandwidth.
Response: `15`
the noise power in the total bandwidth is 15 dBm.

Usage: Query only

Manual operation: See "Noise Power (Total Bandwidth)" on page 158

[:SOURce<hw>]:AWGN:POWer:SUM?

In Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`) modes, queries the overall power of the noise signal plus useful signal.

Return values:

<Sum> float

Range: -145 to 20
Increment: 0.01
*RST: 0

Example: `SOURce:AWGN:POWer:SUM?`
queries the overall power of the noise signal plus useful signal.

Usage: Query only

Manual operation: See "Carrier + Noise Power, Carrier + Interferer Power" on page 158

[:SOURce<hw>]:AWGN:POWer:SUM:PEP?

In Additive Noise and CW Interferer (`SOUR:AWGN:MODE ADD|CW`) modes, queries the peak envelope power of the overall signal comprised of noise signal plus useful signal.

Return values:

<Pep> float

Range: -145 to 20
Increment: 0.01
*RST: 0

Example: `SOURce:AWGN:POWer:SUM:PEP?`
queries the peak envelope power of the overall signal.

Usage: Query only

Manual operation: See "Carrier+Noise PEP, Carrier + Interferer PEP" on page 158

[:SOURce<hw>]:AWGN:SLENgth <SLength>

In mode Noise Only (`SOUR:AWGN:MODE ONLY`), sets the sequence length of the signal in samples.

The sequence length depends on the configured instrument. If the configured instrument is changed, the value range of this parameter is changed as well.

Parameters:

<SLength> integer

Range: 128 to 1073741824
*RST: 20480

Example:
`AWGN:MODE ONLY`
activates the generation of a pure noise.
`AWGN:STAT ON`
switches on the generation of a pure noise.
`AWGN:SLEN 1024`
sets the sequence legth.

Manual operation: See "Sequence Length" on page 154

[:SOURce<hw>]:AWGN:SRATe <SRate>

This command sets the bit rate which is used for calculation of bit energy to noise power ratio from carrier/noise ratio for Digital Standard signals. For Custom Digital Mod signals, the bit rate which is used for calculation can be queried with this command.

Parameters:

<SRate> float

Range: 400 to 250E6 bps
Increment: 0.001
*RST: 40E6
Default unit: Valid units are bps, kbps and Mbps.

Example:
`AWGN:BRAT?`
queries the bit rate which is used for calculation of the Eb/N0 value from the C/N value.

Manual operation: See "Sample Rate" on page 154

[:SOURce<hw>]:AWGN:STATe <State>

Activates or deactivates the white noise (AWGN = Averaged White Gaussian Noise).

Use the command `[:SOURce<hw>]:AWGN:MODE` to define the mode the AWGN generator is working in.

Parameters:
<State> 0 | 1 | OFF | ON
*RST: 0

Example: `SOURce:AWGN:STATe ON`

Manual operation: See "State" on page 154

12.7 SOURce:BB:ARB:MCAR Subsystem

The `MCARrrier` subsystem contains the commands for setting the Multi Carrier Waveform Generator.

CARRier<ch>

The numerical suffix under `CARRier` distinguish between the carriers. The value range is 0 .. 31.

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:COUNt <Count>

Sets the number of carriers in the ARB multi carrier waveform.

Parameters:

<Count> integer

Range: 1 to 512
*RST: 1

Example: `BB:ARB:MCAR:CARR:COUN 10`
sets 10 carriers for the multi carrier waveform.

Manual operation: See "Number of Carriers" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:MODE <Mode>

The command sets the carrier frequency mode.

Parameters:

<Mode> EQUidistant | ARBitrary

EQUidistant
Sets an equidistant carrrier spacing. The carrier frequency in the carrier table is not configurable.

ARBitrary
Enables you to specify the carrier frequency in the carrier table. Carrier spacing is irrelevant.

*RST: EQUidistant

Example: `BB:ARB:MCAR:CARR:MODE EQU`
sets an equidistant carrier spacing. The carrier frequency can not be set.

Manual operation: See "Mode" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier:SPACing <Spacing>

The command sets the frequency spacing between adjacent carriers of the multi carrier waveform. The carriers are generated symmetrically around the RF carrier. The maximum carrier spacing is limited to **Carrier spacing = Total baseband bandwidth / (Number of carriers - 1).**

The total baseband bandwidth is 80 MHz.

Note: In order to avoid wrap-around problems, the effective "Carrier Spacing" might be slightly modified. The "Carrier Spacing" is rounded in that way that the carrier closest to the center RF frequency shows no phase jump assuming that the carrier is unmodulated.

- For odd number of carriers:
  RoundedCarrierSpacing=1/OutputSignalDuration* round(CarrierSpacing * OutputSignalDuration);
- For even number of carriers:
  RoundedCarrierSpacing=2/OutputSignalDuration*round(0.5 *CarrierSpacing * OutputSignalDuration);

Parameters:

<Spacing> float

Range: 0.0 to 120E6
Increment: 0.01
*RST: 0
Default unit: Hz

Example: `BB:ARB:MCAR:CARR:SPAC 10 MHz`
sets a carrier spacing of 10 MHz.

Manual operation: See "Carrier Spacing" on page 113

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:CONFlict?

Queries carrier conflicts. A conflict arises when the carriers overlap.

Return values:

<Conflict> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:ARB:MCAR:CARR2:CONF?`
queries the multi carrier conflict state.
Response: `0` no conflict has occurred.

Usage: Query only

Manual operation: See "!!!" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:DELay <Delay>

Sets the start delay of the selected carrier.

Parameters:

<Delay> float

Range: 0 to 1
Increment: 1E-9
*RST: 0
Default unit: s

Example: `BB:ARB:MCAR:CARR15:DEL 5us`
sets a start delay of 50 us for carrier 15.

Manual operation: See "Delay" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:FILE <File>

Selects the file with I/Q data to be modulated onto the selected carrier.

Parameters:

<File> <file name>

Example: `BB:ARB:MCAR:CARR15:FILE "D:\user\IQ_wcdma"`
selects file `IQ_wcdma`. The data of the file is modulated onto carrier 15.

Manual operation: See "File" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:FREQuency <Frequency>

Sets or indicates the carrier frequency, depending on the selected carrier frequency mode.

The carrier frequnecy can be set in "Arbitrary Carrier frequency" mode. For "Equidistant Carrier Spacing", the carrier spacing is determined automatically.

Parameters:

<Frequency> integer

Vvalue range depends on the max bandwidth, see data sheet

Range: -40E6 to 40E6
*RST: 0

Example: `BB:ARB:MCAR:CARR:MODE ARB`
activates ARB multi carrier mode.
`BB:ARB:MCAR:CARR:FREQ 5.0`
sets 5.0 MHz carrier frequency.

Manual operation: See "Carrier Freq [MHz]" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:PHASe <Phase>

The command sets the start phase of the selected carrier.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Phase> float

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example: `BB:ARB:MCAR:CARR15:PHAS 90 DEG`
sets a start phase of 90° for carrier 15.

Manual operation: See "Phase" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:POWer <Power>

The command sets the gain of the selected carrier.

Parameters:

<Power> float

Range: -80 to 0
Increment: 0.01
*RST: 0
Default unit: dB

Example: `BB:ARB:MCAR:CARR15:POW -50 dB`
sets the power of carrier 15 to -50 dB.

Manual operation: See "Gain" on page 119

[:SOURce<hw>]:BB:ARBitrary:MCARrier:POWer:REFerence <Reference>

Defines the way the individual carriers in a composed multi carrier signal are leveled.

Parameters:

<Reference> RMS | PEAK

*RST: RMS

Manual operation: See "Power Reference" on page 115

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CARRier<ch>:STATe <State>

Enables/disables the selected carrier.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `SOURce1:BB:ARBitrary:MCARrier:CARRier15:STATe ON`

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CFACtor:MODE <Mode>

The command sets the mode for optimizing the crest factor by calculating the carrier phases.

Parameters:

<Mode> OFF | MIN | MAX

OFF
There is no automatic setting for minimizing or maximizing the crest factor. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is in use.

MIN
The crest factor is minimized by internally calculating optimized carrier phases. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is invalid.

MAX
The crest factor is maximized by internally calculating optimized carrier phases. The Phase setting (command `BB:ARB:MCAR:CARR:PHAS`) is invalid.

*RST: OFF

Example: `BB:ARB:MCAR:CFAC:MODE OFF`
switches off automatic crest factor optimization. The setting `SOUR:BB:ARB:MCAR:CARR:PHAS` has an effect.

Manual operation: See "Crest Factor Mode" on page 113

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping:CFACtor <CFactor>

Sets the value of the desired crest factor, if baseband clipping is enabled (`BB:ARB:MCAR:CLIP:STAT ON`).

A Target Crest Factor above the crest factor of the unclipped multicarrier signal has no effect.

Parameters:

<CFactor> float

Range: -50 to 50
Increment: 0.01
*RST: 50
Default unit: dB

Example: `BB:ARB:MCAR:CLIP:STAT ON`
enables clipping.
`BB:ARB:MCAR:CLIP:CFAC 37 dB`
sets the target crest factor.

Manual operation: See "Target Crest Factor" on page 113

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping:CUToff <Cutoff>

Sets the cut off frequency of the final lowpass filter, if baseband clipping is enabled (`BB:ARB:MCAR:CLIP:STAT ON`).

When the cut off frequency is set as half of the output sample rate, a final lowpass filter improves the spectrum of the clipped multicarrier signal, but may also increase the resulting crest factor.

Parameters:

<Cutoff> float

Range: 0 to 50E6
Increment: 0.01
*RST: 50E6
Default unit: MHz

Example: `BB:ARB:MCAR:CLIP:STAT ON`
enables clipping
`BB:ARB:MCAR:CLIP:CUT 50 MHz`
sets the cut off frequency of the filter.

Manual operation: See "Filter Cut Off Frequency" on page 114

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLIPping[:STATe] <State>

Switches baseband clipping on and off.

Clipping reduces the peak power of the resulting multi carrier signal according to the value set with the command `BB:ARB:MCAR:CLIP:CFAC`.

The resulting clipped peak power is defined by sum of the the RMS level of the unclipped multi carrier signal and the input parameter Target Crest Factor. Note that clipping reduces also the RMS level. Hence the resulting crest factor of the clipped signal is slightly above the Target Crest Factor. In order to get the unclipped parts of the clipped multicarrier signal matched with the unclipped multicarrier signal, the RF output power should be reduced by the difference between resulting crest factor and Target Crest Factor.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:ARB:MCAR:CLIP:STAT ON`
enables clipping
`BB:ARB:MCAR:CLIP:CFAC 37 dB`
sets the target crest factor.

Manual operation: See "Clipping" on page 113

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLOad

Creates a multi carrier waveform using the current entries of the carrier table.

This multi carrier waveform is saved with the file name specified with command `SOUR:BB:ARB:MCAR:OFIL`. The file extension is `*.wv`. Digital standard "ARB" is activated, the new multi carrier waveform is loaded and is output in accordance to the trigger settings.

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:MCAR:OFIL 'mcar1_2'`
defines the file name `mcar1_2.wv` for the multi carrier waveform.
`BB:ARB:MCAR:CLO`
creates multi carrier waveform `mcar1_2.wv`.
The new multi carrier waveform is loaded and digital standard "ARB" is activated.

Usage: Event

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CLOCk?

The command queries the resulting sample rate at which the multi carrier waveform is output by the arbitrary waveform generator. The output clock rate depends on the number of carriers, carrier spacing and input sample rate of the leftmost or rightmost carriers.

Return values:
<Clock> float

Range: 400 to Max
Increment: 1E-3

Example: `BB:ARB:MCAR:CLOC?`
queries the ARB multi carrier output clock rate.

Usage: Query only

Manual operation: See "Clock Rate" on page 117

[:SOURce<hw>]:BB:ARBitrary:MCARrier:CREate

Creates a multi carrier waveform using the current settings of the carrier table. The multi carrier waveform is saved into the file defined with command `SOUR:BB:ARB:MCAR:OFIL`. The file extension is `*.wv`.

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:MCAR:OFIL 'multi_wv1'`
defines the file name `multi_wv1.wv` for the multi carrier waveform.
`BB:ARB:MCAR:CRE`
creates multi carrier waveform `multi_wv1.wv`.

Usage: Event

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:DELay:STEP <Step>

The command sets the step width by which the start delays of the carriers in the defined carrier range will be incremented.

Parameters:

<Step> float

Range: -1 to 1
Increment: 1E-9
*RST: 0
Default unit: s

Example: `BB:ARB:MCAR:EDIT:CARR:DEL 5 us`
sets a start delay of 5 us for the carriers in the carrier range.
`BB:ARB:MCAR:EDIT:CARR:DEL:STEP 1 us`
the start delay is incremented by 1us for each carrier, i.e. the first carrier has a start delay of 5us, the second a start delay of 6 us, etc.

Manual operation: See "Delay Step" on page 122

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:DELay[:STARt] <Start>

The command sets the start delay for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:DEL:STEP` is used to define a step width, the delay entered here applies only to the starting carrier. The delays of the remaining carriers are stepped up or down by the delay value specified in the `:BB:ARB:MCAR:EDIT:CARR:DEL:STEP` command.

Parameters:

<Start> float

Range: 0 to 1
Increment: 1E-9
*RST: 0
Default unit: s

Example: `BB:ARB:MCAR:EDIT:CARR:DEL 5us`
sets a start delay of 5 us for the carriers in the carrier range.

Manual operation: See "Delay Start" on page 121

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:EXECute

The command adopts the settings for the carrier range which has been defined using the `:BB:ARB:MCAR:EDIT:CARR:...` commands.

Example: `BB:ARB:MCAR:EDIT:CARR:STAR 4`
the carrier range starts at carrier 2.
`BB:ARB:MCAR:EDIT:CARR:STOP 20`
the carrier range stops at carrier 20.
`BB:ARB:MCAR:EDIT:CARR:STAT ON`
sets all the carriers in the carrier range (2 to 20) to ON.
`BB:ARB:MCAR:EDIT:CARR:EXEC`
transfers the assistant settings for carrier 2 to 20 into the carrier table.

Usage: Event

Manual operation: See "Apply Assistant Settings" on page 122

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:FILE <File>

Selects input file

Parameters:

<File> string

Example: `BB:ARB:MCAR:EDIT:CARR:FILE "D:\user\IQ_wcdma'`

Manual operation: See "Input Waveform File" on page 122

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:PHASe:STEP <Step>

The command sets the step width by which the start phases of the carriers in the defined carrier range will be incremented.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Step> float

Range: -359.99 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example: `BB:ARB:MCAR:EDIT:CARR:PHAS 90 DEG`
sets a start phase of 90° for the carriers in the carrier range.
`BB:ARB:MCAR:EDIT:CARR:PHAS:STEP 1 DEG`
the start phase is incremented by 1° for each carrier, i.e. the first carrier has a start phase of 90°, the second a start phase of 91°, etc.

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:PHASe[:STARt] <Start>

The command sets the start phase for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:PHAS:STEP` is used to define a step width, the phase entered here applies only to the starting carrier. The phases of the remaining carriers are stepped up or down by the phase value specified in the `:BB:ARB:MCAR:EDIT:CARR:PHAS:STEP` command.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:ARB:MCARrier:CFACtor:MODE OFF`).

Parameters:

<Start> float

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example: `BB:ARB:MCAR:EDIT:CARR:PHAS 90 DEG`
sets a start phase of 90° for the carriers in the carrier range.

Manual operation: See "Phase Start" on page 121

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:POWer:STEP <Step>

The command sets the step width by which the starting power of the carriers in the defined carrier range will be incremented.

Parameters:

<Step> float

Range: -80 to 80
Increment: 0.01
*RST: 0
Default unit: dB

Example: `BB:ARB:MCAR:EDIT:CARR:POW -80dB`
sets a power of -80 dB for the carriers in the carrier range.
`BB:ARB:MCAR:EDIT:CARR:POW:STEP 1 dB`
the power is incremented by 1dB for each carrier, i.e. the first carrier has -80dB, the second -79dB, etc.

Manual operation: See "Gain Step" on page 121

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:POWer[:STARt] <Start>

The command sets the power for the individual carriers in the defined carrier range. If the command `:BB:ARB:MCAR:EDIT:CARR:POW:STEP` is used to define a step width, the power entered here applies only to the starting carrier. The power of the remaining carriers is stepped up or down by the power specified in the `:BB:ARB:MCAR:EDIT:CARR:POW:STEP` command.

Parameters:

<Start> float

Range: -80 to 0
Increment: 0.01
*RST: 0
Default unit: dB

Example: `BB:ARB:MCAR:EDIT:CARR:POW -50 dB`
sets the power of the carriers in the carrier range to -50 dB.

Manual operation: See "Gain Start" on page 121

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STARt <Start>
[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STOP <Stop>

The command selects the last carrier in the carrier range to which the settings with the `:BB:ARB:MCAR:EDIT:CARR:..` commands shall apply.

Parameters:

<Stop> integer

Range: 0 to 511
*RST: 0

Example: `BB:ARB:MCAR:EDIT:CARR:STOP 4`
the carrier range stops at carrier 4.

Manual operation: See "Carrier Start/Stop" on page 121

[:SOURce<hw>]:BB:ARBitrary:MCARrier:EDIT:CARRier:STATe <State>

The command switches all the carriers in the selected carrier range on or off.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 1

Example: `BB:ARB:MCAR:EDIT:CARR:STAT ON`
sets all the carriers in the carrier range to ON.

Manual operation: See "State" on page 118

[:SOURce<hw>]:BB:ARBitrary:MCARrier:OFILe <OFile>

This command defines the output file name for the multi carrier waveform. This file name is used when a waveform is calculated (command `SOUR:BB:ARB:MCAR:CLOad` or `SOUR:BB:ARB:MCAR:CREate`). The file extension is `*.wv`.

Parameters:

<OFile> string

Example: `MMEM:CDIR "D:\user\waveform"`
sets the default directory.
`BB:ARB:MCAR:OFIL 'mcar1_2'`
defines the file name `mcar1_2.wv` for the multi carrier waveform file

Manual operation: See "Output File" on page 117

[:SOURce<hw>]:BB:ARBitrary:MCARrier:PRESet

Sets all ARB multi carrier parameters to their default values.

Example: `SOURce1:BB:ARB:MCARrier:PRESet`

Usage: Event

Manual operation: See "Set to Default" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:SAMPles?

The command queries the resulting file size. The file size is returned in samples.

Return values:

<Samples> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:ARB:MCAR:SAMP?`
queries the file size of the currently calculated multi carrier waveform.

Usage: Query only

Manual operation: See "File Size" on page 117

[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:CATalog?

Queries the available settings files in the specified default directory. The settings files are used to set the ARB multi carrier submenu. Only files with the file extension `*.arb_multcarr` will be listed.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory
`BB:ARB:MCAR:SETT:CAT?`
reads out all the settings files in the default directory.
`Response: mcar1, mcar2`
the directory contains the configuration files `mcar1.arb_multcarr` and `mcar2.arb_multcarr`.

Usage: Query only

Manual operation: See "Save/Recall Frame" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:LOAD <Filename>

Loads the settings file. If a settings file with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.arb_multcarr` will be loaded or created.

Setting parameters:

<Filename> string

Example: `BB:ARB:MCAR:SETT:LOAD 'D:\user\new'`
creates settings file `new.arb_multcarr`.

Usage: Setting only

Manual operation: See "Save/Recall Frame" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:STORe <Filename>

The command stores the current settings of submenu "Multi Carrier" in a file in the specified directory. The file extension may be omitted, the files are stored with the file extension `*.arb_multcarr`.

Setting parameters:

<Filename> string

Example: `BB:ARB:MCAR:SETT:STOR 'D:\user\mcarr2'`
stores settings file `mcarr2.arb_multcarr` in the default directory.

Usage: Setting only

Manual operation: See "Save/Recall Frame" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast> 0 | 1 | OFF | ON
*RST: 1

Manual operation: See "Save/Recall Frame" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:STATe <State>

Activates multi carrier generation and deactivates all the other digital standards and digital modulation modes in the same path.

Parameters:

<State> 0 | 1 | OFF | ON
*RST: 0

Example: `SOURce1:BB:ARBitrary:MCARrier:STATe ON`

Manual operation: See "State Multi Carrier" on page 112

[:SOURce<hw>]:BB:ARBitrary:MCARrier:TIME <Time>

Sets the user-defined signal period. This setting is only possible for Signal Period Mode User (`BB:ARB:MCAR:TIME:MODE USER`).

Parameters:

<Time> float

Range: 0 to 1E9
Increment: 1E-9
*RST: 0
Default unit: s

Example: `BB:ARB:MCAR:TIME:MODE USER`
selects Signal Period Mode User.
`BB:ARB:MCAR:TIME 10 s`
sets a signal period of 10 seconds

Manual operation: See "Signal Period" on page 114

[:SOURce<hw>]:BB:ARBitrary:MCARrier:TIME:MODE <Mode>

Selects the mode for calculating the resulting signal period of the multi carrier waveform. The resulting period is always calculated for all carriers in the carrier table irrespective of their state (ON/OFF).

Parameters:

<Mode> LONG | SHORt | LCM

LONG
The resulting signal period is defined by the longest I/Q file in the carrier table. Shorter I/Q files are periodically repeated.

SHORt
The resulting signal period is defined by the shortest I/Q file in the carrier table. Only the first part of longer I/Q files is used.

LCM
The output file duration is the least common multiple of all input file durations.

*RST: LONG

Example: `SPOURce1:BB:ARBitrary:MCARrier:TIME:MODE LONG`
selects signal period mode long

Manual operation: See "Signal Period Mode" on page 114

12.8 SOURce:BB:ARB:WSEG Subsystem

The `WSEGment` subsystem contains the commands for setting the Multi Segment Waveform Generator.

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:BLANk:APPend
<SampCount>, <Frequency>

Adds a blank segment to the multi segment file.

Setting parameters:

<SampCount> float

Specifies the number of samples.

Range: 512 to 1E7
Increment: 1
*RST: 1000

<Frequency> float

Determines the clock rate.

Range: 400 Hz to Max (depends on instrument type)
Increment: 0.001
*RST: 1E8

Example: `SOUR:BB:ARB:WSEG:CONF:SEL "MSegFile"`
selects a multi segment file.
`SOUR:BB:ARB:WSEG:CONF:BLAN:APP 1000,100000000`
adds a blank segment with 1000 samples and 100 MHz clock rate to the selected multi segment file

Usage: Setting only

Manual operation: See "Blank Segment" on page 131

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CATalog?

Queries the available configuration files in the specified default directory. The configuration files are used to create multi segment waveform files.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:CONF:CAT?`
reads out all the configuration files in the default directory.
`Response: mulit1, multi2`
the directory contains the configuration files `multi1.inf_mswv` and `multi2.inf_mswv`.

Usage: Query only

Manual operation: See "Load List" on page 129

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CLOCk <Clock>

This command defines the clock rate used for multi segment waveform output in case of Clock Mode "User" (`:BB:ARB:WSEG:CONF:CLOCk:MODE USER`).

Parameters:

<Clock> float

Increment: 1E-3
*RST: max SampleRate

Example: `BB:ARB:WSEG:CONF:CLOC:MODE USER`
selects Clock Mode User.
`BB:ARB:WSEG:CONF:CLOC 50MHz`
defines a clock rate of 50 MHz.

Manual operation: See "User Clock Rate" on page 133

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CLOCk:MODE <Mode>

Selects the clock rate mode for the multi segment waveform.

Parameters:

<Mode> UNCHanged | HIGHest | USER

UNCHanged
The segments are output with the clock rate defined in the waveform file.

HIGHest
The segments are output at the highest available clock rate.

USER
The segments are output with the clock rate defined with command `[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:CLOCk`

*RST: UNCHanged

Example: `BB:ARB:WSEG:CONF:CLOC:MODE UNCH`
selects clock mode unchanged. The segments are output with the clock rate defined in the waveform file.

Manual operation: See "Clock" on page 132

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:COMMent <Comment>

This command enters a comment for the configuration file. The configuration file must be specified with command `:BB:ARB:WSEG:CONF:SEL`.

Parameters:

<Comment> string

Example: `BB:ARB:WSEG:CONF:COMM <3gpp_up>`
enters comment "3gpp_up".

Manual operation: See "Comment" on page 129

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:DELete <Filename>

Deletes the configuration file. The configuration files are used to create multi segment waveform files.

Setting parameters:

<Filename> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:CONF:DEL 'multi1'`
deletes configuration file `multi1.inf_mswv`.

Usage: Setting only

Manual operation: See "Multi Segment Table, Append/Delete/Shift Seg." on page 130

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:LEVel[:MODE] <Mode>

This command selects the level mode for the multi segment waveform.

Parameters:

<Mode> UNCHanged | ERMS

UNCHanged
The segments are output exactly as defined in the files.

ERMS
The segments are output so that all segments have the same rms value.

*RST: UNCHanged

Example: `BB:ARB:WSEG:CONF:LEV:MODE UNCH`
selects level mode unchanged. The segments are output as defined in the waveform file.

Manual operation: See "Level" on page 132

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:ESEGment <Mode>

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

Parameters:

<Mode> OFF | MRK1 | MRK2 | MRK3 | MRK4

OFF
No additional marker is generated.

MRK1|MRK2|MRK3|MRK4
Generates a restart marker signal at the beginning of each segment for the corresponding marker.
The segment begin is defined by the low-high slope of the marker. This applies for switching between two segments as well as in case of segment replay.

*RST: OFF

Example:

```
BB:ARB:WSEG:CONF:SEL 'ConfComm'
BB:ARB:WSEG:CONF:MARK:ESEG MRK3
```

Manual operation: See "Segment Restart" on page 133

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:FSEGment <Mode>

Enables/disables the generation of an additional marker restart signal in the output waveform file.

If additional marker generation is enabled, the existing marker signals in the individual segment waveform files are not considered.

Parameters:

<Mode> OFF | MRK1 | MRK2 | MRK3 | MRK4

OFF
No additional marker is generated.

MRK1|MRK2|MRK3|MRK4
Generates a restart marker signal at the beginning of the first segment for the corresponding marker.
Use this setting to generate a restart marker for the complete sequence.

*RST: OFF

Example:

```
BB:ARB:WSEG:CONF:SEL 'ConfComm'
BB:ARB:WSEG:CONF:MARK:FSEG MRK3
```

Manual operation: See "Sequence Restart" on page 133

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:MARKer:MODE <Mode>

Defines the way the marker information within the separate segments is processed.

Parameters:

<Mode> IGNore | TAKE

IGNore
The marker information carried in the individual segment waveform files is not considered.

TAKE
The output waveform file contains the marker information as configured in the individual waveform files.

*RST: TAKE

Example:
`BB:ARB:WSEG:CONF:SEL 'ConfComm'`
`BB:ARB:WSEG:CONF:MARK:MODE TAKE`

Manual operation: See "Segment Marker" on page 133

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:OFILe <OFile>

Defines the file name for the multi segment waveform. The file extension is `*.wv`.

Parameters:

<OFile> string

Example:
`MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:CONF:SEL 'multi1'`
creates the configuration file `multi1.inf_mswv` in default directory.
`BB:ARB:WSEG:CONF:OFIL 'mseg1_2'`
defines the file name `mseg1_2.wv` for the multi segment waveform file created using configuration file `multi1.inf_mswv`.

Manual operation: See "Save List/Save List As..." on page 129

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SEGMent:APPend <Waveform>

Appends the specified waveform to the configuration file.

Setting parameters:

<Waveform> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory
`BB:ARB:WSEG:CONF:SEL 'new'`
creates the configuration file `new.inf_mswv` in the default directory.
`BB:ARB:WSEG:CONF:SEGM:APP 'arb1'`
appends waveform `arb1.wv` to configuration file `new`. Waveform `arb1` will be the first segment of a multi segment waveform created with configuration file `new`.

Usage: Setting only

Manual operation: See "Multi Segment Table, Append/Delete/Shift Seg." on page 130

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SEGMent:CATalog?

Queries the segments of the currently selected configuration file.

Return values:
<Catalog> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:CONF:SEL 'multi_sin'`
selects the configuration file `multi_sin.inf_mswv`.
`BB:ARB:WSEG:CONF:SEGM:CAT?`
queries the segments of the selected configuration file.
`Response: arb4, arb2`
The configuration file includes the segments `arb4.wv` and `arb2.wv`.

Usage: Query only

Manual operation: See "Multi Segment Table, Append/Delete/Shift Seg." on page 130

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CONFigure:SELect <Filename>

Selects the configuration file in the default directory.

A path can also be specified, in which case the files in the specified directory are selected. If a configuration file with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.inf_mswv` will be selected or created.

Parameters:
<Filename> string

Example: `MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:CONF:SEL 'new'`
create configuration file `new.inf_mswv` in the default directory.

Manual operation: See "New List" on page 129

[:SOURce<hw>]:BB:ARBitrary:WSEGment:CREate <FilenameInput>

Creates a multi segment waveform using the current settings of the specified configuration file. The multi segment waveform is saved into the file defined in the configuration file. The file extension is `*.wv`.

Setting parameters:

<FilenameInput> string

Example:

`MMEM:CDIR 'D:\user\'`
sets the default directory
`BB:ARB:WSEG:CONF:SEL 'new'`
creates the configuration file `new.inf_mswv` in the default directory.
`BB:ARB:WSEG:CONF:SEGM:APP 'arb1.wv'`
'includes waveform `arb1.wv` as segment 1 in the configuration file. The waveform must be available in the default directory.
`BB:ARB:WSEG:CONF:OFIL 'multi_wv1'`
defines the file name `multi_wv1.wv` for the multi segment waveform.
`BB:ARB:WSEG:CRE 'D:\user\new.inf_mswv'`
creates multi segment waveform `multi_wv1.wv` using the settings of the configuration file `new.inf_mswv`.

Usage: Setting only

[:SOURce<hw>]:BB:ARBitrary:WSEGment:NEXT <Next>

Selects the segment to be output.

Parameters:

<Next> float

Range: 0 to 1023
*RST: 0

Example:

`MMEM:CDIR 'D:\user\waveform'`
sets the default directory.
`BB:ARB:WSEG:NEXT 2`
selects segment 2 to be output.
`BB:ARB:TRIG:SMOD NEXT`
selects extended trigger mode next, i.e. the segment specified with command `:BB:ARB:WSEG:NEXT` will be output.
`BB:ARB:SEQ AUTO`
selects trigger mode Auto.
`BB:ARB:WAV:SEL 'multi_wv1'`
loads multi segment waveform `multi_wv1.wv`. Generation of segment 2 starts.
`BB:ARB:WSEG:NEXT 3`
switched at once to output of segment 3.

[:SOURce<hw>]:BB:ARBitrary:WSEGment:NEXT:EXECute

Triggers manually switchover to the subsequent segment in the mutli segment file. A manual trigger can be executed only when an internal next segment source (`BB:ARB:WSEG:NEXT:SOUR INT`) has been selected.

To perform a switchover to any segment within the multi segment file, select the next segment with the command `BB:ARB:WSEG:NEXT`.

This command is disabled, if a sequencing play list is enabled.

Example: `BB:ARB:WSEG:NEXT:SOUR INT`
selects internal next segment source.
`BB:ARB:WSEG:NEXT 2`
selects segment 2 to be output.
`BB:ARB:WSEG:NEXT:EXEC`
executes a switchover to the next segment.

Usage: Event

[:SOURce<hw>]:BB:ARBitrary:WSEGment:SEQuence:APPend <State>, <Segment>, <Count>, <Next>

Appends a new segment to the selected sequencing play list.

Setting parameters:

<State> ON | OFF
activates/deactivates the appended segment
*RST: ON

<Segment> integer
indicates the number of the segment as in the multi segment waveform file
Range: 0 to SegmentCount - 1

<Count> integer
defines how many times this segment is repeated
Range: 1 to 65535

<Next> NEXT | BLANk | ENDLess | SEG0 | SEG1 | ... | SEG31 | 0...maxSegment
determines the action after completing the current segment, like for instance which segment is processed after the processing of the current one is finished.

Example: `BB:ARB:WSEG:SEQ:SEL 'D:\user\play_list_1'`
selects the sequencing list `play_list_1.wvs`.
`BB:ARB:WSEG:SEQ:APP ON,3,15,BLANK`
appends the segment number 3 as a new segment to the sequencing list; this segment is activated and will be repeated 15 times and followed by a blank segment.

Usage: Setting only

[:SOURce<hw>]:BB:ARBitrary:WSEGment:SEQuence:SELect <Filename>

Selects the sequencing list (files with extension `*.wvs`)

Parameters:

<Filename> string

Example: `BB:ARB:WSEG:SEQ:SEL 'D:\user\play_list_1'`
selects the sequencing list `play_list_1.wvs`.

[:SOURce<hw>]:BB:ARBitrary:WSEGment:STATe <State>

Enables/disables multi segment generation.

Parameters:

<State> 0 | 1 | OFF | ON
*RST: 0

Example: `SOURce1:BB:ARBitrary:WSEGment:STATe ON`

Manual operation: See "State" on page 129

12.9 SOURce:BB:DM Subsystem

The commands in the `SOURce:BB:DM` subsystem are described in two sections, separated into configuring digital modulation and lists for digital modulation.

12.9.1 DM General Remote-Control Commands

The following section contains the commands for generating the digital modulation signal.

12.9.1.1 Common Settings

[:SOURce<hw>]:BB:DM:PATTern <Pattern>

The command selects the data pattern for the internal data when PATTern is selected as the data source. The maximum length is 64 bits.

Parameters:

<Pattern>	string	
	Range:	#B0,1 to #B111...1,64
	*RST:	#B0,1

Example: `BB:DM:SOUR PATT`
selects Pattern as the data source for digital modulation.
`BB:DM:PATT #B01110111010101010,17`
generates the user-defined sequence of 0/1 data.

[:SOURce<hw>]:BB:DM:PRBS[:LENGth] <Length>

The command defines the length of the pseudo-random sequence in accordance with the following equation:

Length = (2^`Length`) - 1

Parameters:

<Length> 9 | 11 | 15 | 16 | 20 | 21 | 23 | PN9 | PN11 | PN15 | PN16 | PN20 | PN21 | PN23

*RST: 9

Example: `BB:DM:SOUR PRBS`
The internal pseudo-random generator is used as the data source.
`BB:DM:PRBS 9`
An internal pseudo-random sequence of 511 bits will be generated.

Manual operation: See "Data Source" on page 71

[:SOURce<hw>]:BB:DM:PRESet

Sets the default settings for digital modulation (`*RST` values specified for the commands).

Not affected is the state set with the command `SOURce<hw>:BB:DM:STATe`

Example: `SOURce1:BB:DM:PRESet`

Usage: Event

Manual operation: See "Set To Default" on page 68

[:SOURce<hw>]:BB:DM:SOURce <Source>

The command selects the data source.

Parameters:

<Source> ZERO | ONE | PRBS | PATTern | DLISt

ZERO
An internally generated 0 data sequence is used.

ONE
An internally generated 1 data sequence is used.

PRBS
The pseudo-random sequence generator is used as the data source.The length of the random sequence is defined with the aid of command `SOURce:BB:DM:PRBS`.

PATTern
Internally generated data with a fixed pattern is used. The data pattern is dafined using command `SOURce:BB:DM:PATTern`.

DLISt
Data from the selected data list is used.

*RST: PRBS

Example: `BB:DM:SOUR DLIS`
the internal data generator is used.
`BB:DM:DLIS:SEL 'test'`
the data list `test.dm_iqd` is used.

Manual operation: See "Data Source" on page 71

[:SOURce<hw>]:BB:DM:SRATe <SRate>

Sets the symbol rate. The value range is dependent on the selected modulation type. On changing to an FSK modulation type, excessively high values are automatically limited to the maximum value that can be set for FSK (see data sheet). The symbol rate can be entered in Hz/kHz/MHz or Symb/s / kSymb/s and MSymb/s.

When a standard is selected (`DM:STANdard`), the symbol rate is automatically set to the appropriate default value.

The value range is dependent on the selected modulation type (`:BB:DM:FORM`). When a standard is selected (`:BB:DM:STAN`), the symbol rate is set to the default value.

Parameters:

<SRate> float

Range: 400 to depends on hardware
Increment: 0.001
*RST: 270833.333
Default unit: Hz

Example: `BB:DM:SRAT 10 MHz`
sets a symbol rate of 10 MHz.

Manual operation: See "Symbol Rate" on page 69

[:SOURce<hw>]:BB:DM:STANdard <Standard>

Selects the standard.

After selection, modulation parameters "Modulation Type" (`:BB:DM:FORMat`), "Symbol Rate" (`:BB:DM:SRATe`), "Filter" (`:BB:DM:FILTer:TYPE` and `:BB:DM:FILTer:PARameter:...`) and "Coding" (`:BB:DM:CODing`) are automatically set in accordance with the standard.

The `USER` parameter cannot be set. A query returns this value if a user-defined Custom Dig Mod setting was loaded or if one of the associated settings was changed subsequent to the selection of a standard. The user defined settings are stored and loaded with commands `:BB:DM:STAN:ULIS:...` (see Chapter 12.9.2, "DM Lists", on page 340).

Parameters:

<Standard> USER | BLUetooth | DECT | ETC | GSM | GSMEdge | NADC | PDC | PHS | TETRa | W3GPp | TDSCdma | CFORward | CREVerse | WORLdspace | TFTS | APCOPH1C4fm | APCOPH1CQpsk | APCOPH2HCpm | APCOPH2HDQpsk | APCOPH2HD8PSKW | APCOPH2HD8PSKN | APCOPH1Lsm | APCOPH1Wcqpsk

*RST: GSM

Example: `BB:DM:STAN DECT`

Selects digital modulation according to the DECT standard.

Manual operation: See "Set acc. Standard" on page 69

[:SOURce<hw>]:BB:DM:STATe <State>

Enables/disables digital modulation. Switching on digital modulation turns off all the other digital standards in the same signal path.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `SOURce1:BB:DM:STATe ON`

Manual operation: See "State" on page 68

[:SOURce<hw>]:BB:DM:SLENgth <Slength>

Sets the sequence length of the signal in symbols. The signal is calculated in advance, saved as waveform file, and output in the selected arbitrary waveform generator.

Note: Sequence Length * Oversampling must not exceed the maximum number of samples in the arbitrary waveform generator.

Parameters:

<Slength> integer

Range: 1 to max
*RST: 10000

Example: `BB:DM:SLEN 500`
sets the sequence length to 500 symbols.

Manual operation: See "Sequence Length" on page 68

[:SOURce<hw>]:BB:DM:WAVeform:CREate <Filename>

This command creates a waveform using the current settings of the "Digital Modulation" menu. The file name is entered with the command. The file is stored with the predefined file extension `*.wv`. The file name and the directory it is stored in are user-definable.

Setting parameters:

<Filename> string

Example: `MMEM:CDIR 'd:/user/waveform'`
sets the default directory to `d:/user/waveform`.
`BB:DM:WAV:CRE 'dm_1`
creates the waveform file `dm_1.wv` in the default directory.

Usage: Setting only

Manual operation: See "Generate Waveform File" on page 68

12.9.1.2 Marker Settings

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:MODE <Mode>

Defines the signal for the selected marker output.

Parameters:

<Mode> CLISt | PULSe | PATTern | RATio

CLISt
A marker signal that is defined in the selected control list is generated.

PULSe
A pulsed marker signal is generated. The pulse frequency (= symbol rate/divider) is defined with the `SOUR:BB:DM:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:DM:TRIG:OUTP:PULSe:FREQuency?` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the aid of command `SOURce:BB:DM:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:DM:TRIGger:OUTPut:OFFT` and `SOURce:BB:DM:TRIGger:OUTPut:ONT` is generated.

*RST: RATio

Example: `BB:DM:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

Manual operation: See "Marker Mode" on page 70

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:OFFTime <OffTime>

Sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:DM:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

Parameters:

<OffTime> integer

Range: 1 to 16777215
*RST: 1
Default unit: symbol

Example: `BB:DM:TRIG:OUTP2:OFFT 20`
sets an OFF time of 20 symbols for marker signal 2.

Manual operation: See "Marker Mode" on page 70

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:ONTime <OnTime>

Sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:DM:TRIGger:OUTPut:MODE RATio` on the marker outputs is ON.

Parameters:

<OnTime> integer

Range: 1 to 16777215
*RST: 1
Default unit: symbol

Example: `BB:DM:TRIG:OUTP2:ONT 20`
sets an ON time of 20 symbols for marker 2

Manual operation: See "Marker Mode" on page 70

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PATTern <Pattern>

Sets the bit pattern used to generate the marker signal in the setting `SOURce:BB:DM:TRIGger:OUTPut:MODE PATTern`. 0 is marker off, 1 is marker on.

Parameters:

<Pattern> integer

Range: #B0,1 to #B111...1,32
*RST: #B,1

Example: `BB:DM:TRIG:OUTP2:PATT #B000000011111111,15`
sets a bit pattern.
`BB:DM:TRIG:OUTP2:MODE PATT`
activates the marker signal according to a bit pattern on output MARKER 2.

Manual operation: See "Marker Mode" on page 70

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

Sets the divider for Pulse marker mode (`SOUR:BB:DM:TRIGr:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider> integer

Range: 2 to 1024
*RST: 2

Example: `BB:DM:TRIG:OUTP2:PULS:DIV 2`
sets the divider to 2 for the marker signal on output MARKER 2.
`BB:DM:TRIG:OUTP2:FREQ?`
queries the resulting pulse frequency of the marker signal.
`Response: 66 000`
the resulting pulse frequency is 66 kHz.

Manual operation: See "Marker Mode" on page 70

[:SOURce<hw>]:BB:DM:TRIGger:OUTPut<ch>:PULSe:FREQuency?

Queries the pulse frequency of the pulsed marker signal in the setting `SOURce:BB:DM:TRIGger:OUTPut:MODE PULSe`. The pulse frequency is derived by dividing the symbol rate by the divider.

Return values:

<Frequency> float

Example:
`BB:DM:TRIG:OUTP2:PULS:DIV 2`
sets the divider marker signal on output MARKER 2 to the value 2.
`BB:DM:TRIG:OUTP2:MODE PULS`
enables the pulsed marker signal.
`BB:DM:TRIG:OUTP2:PULS:FREQ?`
queries the pulse frequency of the marker signal.
`Response: 33 000`
the resulting pulse frequency is

Usage: Query only

Manual operation: See "Marker Mode" on page 70

12.9.1.3 Filter Settings

[:SOURce<hw>]:BB:DM:FILTer:ILENgth <ILength>

The command sets the impulse length (number of filter tabs).

Parameters:

<ILength> integer

Range: 1 to 128
*RST: 10

Example: `BB:DM:FILT:ILEN 10`
sets the number of filter tabs to 10.

Manual operation: See "Impulse Length" on page 75

[:SOURce<hw>]:BB:DM:FILTer:ILENgth:AUTO <Auto>

The command activates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

Parameters:

<Auto> 0 | 1 | OFF | ON
*RST: 1

Example: `BB:DM:FILT:ILEN:AUTO ON`
The most sensible parameters are selected automatically.

Manual operation: See "Impulse Length" on page 75

[:SOURce<hw>]:BB:DM:FILTer:OSAMpling <OSampling>

The command sets the upsampling factor.

Parameters:

<OSampling> integer
Range: 1 to 32
*RST: 32

Example: `BB:DM:FILT:OSAM 32`
sets the upsampling factor to 32.

Manual operation: See "Oversampling" on page 75

[:SOURce<hw>]:BB:DM:FILTer:OSAMpling:AUTO <Auto>

The command activates/deactivates the upsampling factor state. If activated, the most sensible parameter values are selected. The value depends on the coherence check. If deactivated, the values can be changed manually.

Parameters:

<Auto> 0 | 1 | OFF | ON
*RST: 1

Example: `BB:DM:FILT:OSAM:AUTO ON`
The most sensible parameters are selected automatically.

Manual operation: See "Oversampling" on page 75

[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine[:ROLLoff] <Cosine>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:GAUSs <Gauss>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASs <LPass>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:LPASSEVM <LPassEvm>

[:SOURce<hw>]:BB:DM:FILTer:PARameter:PGAuss <PGauss>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:RCOSine <RCosine>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:SPHase <SPhase>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25 <Apco25>

The command sets the roll-off factor for filter type APCO25.

On selecting filter APCO25, the filter parameter is set to the default value.

Parameters:

<Apco25> float

Range: 0.05 to 0.99
Increment: 0.01
*RST: 0.2

<Cosine> float

Range: 0.05 to 1.00
Increment: 0.01
*RST: 0.35

<Gauss> float

Range: 0.15 to 100000
Increment: 0.01
*RST: 0.3

<LPass> float

Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<LPassEvm> float

Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<PGauss> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.3

<RCosine> float

Range: 0.05 to 1.00
Increment: 0.01
*RST: 0.35

<SPhase> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

Example: `BB:DM:FILT:PAR:APCO25 0.2`
sets the roll-off factor to 0.2 for filter type APCO25.

Manual operation: See "Filter Parameter" on page 75

[:SOURce<hw>]:BB:DM:FILTer:PARameter:COSine:BANDwidth <FiltParm>

Sets the bandwidth of the cosine filter.

Parameters:

<FiltParm> float

Range: 400 to 25E6
Increment: 1E-3
*RST: 270833.333

Example:
```
:BB:DM:FILT:TYPE COS
:BB:DM:FILT:PAR:COS:BAND 7200
```

Manual operation: See "Bandwidth" on page 75

[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:GAUSs <FiltParm>
[:SOURce<hw>]:BB:DM:FILTer:PARameter:APCO25Lsm:LOWPass <FiltParm>

Sets the cut off frequency for the lowpass/ gauss filter.

Parameters:

<FiltParm> float

Range: 400 to 25E6
Increment: 1E-3
*RST: 270833.333

Example:
```
:BB:DM:FILT:TYPE APCO25Lsm
:BB:DM:FILT:PAR:APCO25L:LOWP 5166.667
:BB:DM:FILT:PAR:APCO25L:GAUS 1700
```

Manual operation: See "Cut Off Frequency Factor" on page 75

[:SOURce<hw>]:BB:DM:FILTer:TYPE <Type>

The command selects the filter type.

When a standard is selected (`:BB:DM:STAN`), the filter type and filter parameter are set to the default value.

Parameters:

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 | COEQualizer | COFequalizer | C2K3x | APCO25 | SPHase | RECTangle | USER | PGAuss | LPASs | DIRac | ENPShape | EWPShape | LTEFilter | LPASSEVM | APCO25Hcpm | APCO25Lsm

*RST: GAUSs

Example:
```
BB:DM:FILT:TYPE COS
```
selects the Cosine filter type.

Manual operation: See "Filter" on page 74

[:SOURce<hw>]:DM:FILTer:PARameter <Parameter>

Sets the filter parameter of the currently selected filter type.

To set the filter type, use command `[:SOURce<hw>]:BB:DM:FILTer:TYPE` on page 334.

Parameters:

<Parameter>	float	
	Range:	0.05 to 2.5
	Increment:	0.01
	*RST:	0.35

Example:
```
BB:DM:FILT:TYPE COS
DM:FILT:PAR 0.5
```
sets 0.5 roll-off factor for the cosine filter.

12.9.1.4 Modulation Settings

[:SOURce<hw>]:BB:DM:AQPSk:ANGLe <Angle>

For AQPSK modulation, sets the angle alpha between the point (0,0) and the I axis.

Parameters:

<Angle>	float	
	Range:	0 to 180
	Increment:	0.01
	*RST:	0
	Default unit:	Deg

Example:
```
BB:DM:FORM AQPS
```
selects modulation type AQPSK.
```
BB:DM:AQPS:ANGL 45
```

Manual operation: See "Angle Alpha" on page 73

[:SOURce<hw>]:BB:DM:ASK:DEPTh <Depth>

The command sets the ASK modulation depth when modulation type ASK is selected.

Parameters:

<Depth> float

Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: PCT

Example: `BB:DM:FORM ASK`
selects the ASK modulation type.
`BB:DM:ASK:DEPT 50 PCT`
sets a modulation depth of 50 percent.

Manual operation: See "ASK Depth" on page 73

[:SOURce<hw>]:BB:DM:CODing <Coding>

The command selects the modulation coding.

If the modulation type selected (`:BB:DM:FORM`) is not possible with the coding that has been set, it is automatically set to OFF (`:BB:DM:COD OFF`).

When a standard is selected (`:BB:DM:STAN`), the coding is set to the default value.

Parameters:

<Coding> OFF | DIFF | DPHS | DGRay | GRAY | GSM | NADC | PDC | PHS | TETRa | APCO25 | PWT | TFTS | INMarsat | VDL | EDGE | APCO25FSK | ICO | CDMA2000 | WCDMA | APCO258PSK

DPHS
Phase Difference

DGRay
Difference + Gray

*RST: INTernal

Example: `BB:DM:COD GRAY`
selects GRAY coding. This coding is valid for all modulation types.

Manual operation: See "Coding" on page 69

[:SOURce<hw>]:BB:DM:FORMat <Format>

Selects the modulation type.

If the coding that is set (`:BB:DM:COD`) is not possible with the modulation type selected, it is automatically set to OFF (`:BB:DM:COD OFF`).

When a standard is selected (`:DM:STAN`), the modulation type is set to the default value.

Parameters:

<Format>	ASK | BPSK | P2DBpsk | QPSK | QPSK45 | OQPSk | P4QPsk | P4DQpsk | PSK8 | P8D8psk | P8EDge | QAM16 | QAM32 | QAM64 | QAM256 | QAM1024 | MSK | FSK2 | FSK4 | USER | FSKVar | QAM128 | QEDGe | QAM16EDge | QAM32EDge | AQPSk | QAM4096 *RST: MSK

Example: `BB:DM:FORM QPSK`
selects modulation type QPSK.

Manual operation: See "Modulation Type" on page 72

[:SOURce<hw>]:BB:DM:FSK:DEViation <Deviation>

Sets the frequency deviation when FSK modulation is selected. The range of values depends on the symbol rate that is set (see data sheet) and the maximum deviation is 10 MHz.

If the symbol rate that is set exceeds the maximum possible value for the chosen frequency deviation, it is suitably adapted (`:BB:DM:SRAT`).

Parameters:

<Deviation>	float Range: 1 to 40E6 Increment: 0.5 *RST: 135416.5

Example: `BB:DM:FORM FSK`
selects FSK modulation.
`BB:DM:FSK:DEV 10 MHz`
sets the frequency deviation to 10 MHz.

Manual operation: See "FSK Deviation" on page 73

[:SOURce<hw>]:BB:DM:FSK:VARiable:SYMBol<ch0>:DEViation <Deviation>

The command sets the deviation of the selected symbol for variable FSK modulation mode.

Suffix:

<ch>	0 .. 16 The number of symbols (and therefore the suffix range) depends on the selected FSK modulation type.

Parameters:

<Deviation>	float The range of values depends on the symbol rate that is set (see data sheet) and the maximum deviation is 10 MHz. Range: -40E6 to 40E6 Increment: 0.5 Default unit: Hz

Example: `BB:DM:FORM FSKV`
selects Variable FSK modulation.
`BB:DM:FSK:VAR:TYPE FSK16`
selects 16FSK modulation.
`BB:DM:FSK:VAR:SYMB0:DEV 135000`
sets the frequency deviation of the least significant symbol to 135 kHz.

Manual operation: See "Deviation xxxx" on page 74

[:SOURce<hw>]:BB:DM:FSK:VARiable:TYPE <Type>

The command selects the modulation type for Variable FSK.

Parameters:
<Type> FSK4 | FSK8 | FSK16
*RST: FSK4

Example: `BB:DM:FORM FSKV`
selects Variable FSK modulation.
`BB:DM:FSK:VAR:TYPE FSK16`
selects 16FSK modulation.

Manual operation: See "FSK Type" on page 74

12.9.1.5 Power Ramp

[:SOURce<hw>]:BB:DM:PRAMp:ATTenuation <Attenuation>

Sets the level attenuation for signal ranges that are flagged with level attribute attenuated by the LEV_ATT control signal.

Parameters:
<Attenuation> float
Range: 0 to 50
Increment: 0.1
*RST: 15
Default unit: dB

Example: `BB:DM:PRAM:ATT 15 dB`
sets a level attenuation of 15 dB.

Manual operation: See "Attenuation" on page 77

[:SOURce<hw>]:BB:DM:PRAMp:FDELay <FDelay>

Sets the delay in the falling edge. A positive value gives rise to a delay and a negative value causes an advance.

Parameters:

<FDelay> float

Range: 0 to 4
Increment: 0.01
*RST: 0
Default unit: symbol

Example: `BB:DM:PRAM:FDEL 1`
The falling edge starts 1 symbol later.

Manual operation: See "Fall Delay" on page 77

[:SOURce<hw>]:BB:DM:PRAMp:RDELay <RDelay>

Sets the delay in the rising edge. A positive value gives rise to a delay and a negative value causes an advance.

Parameters:

<RDelay> float

Range: 0 to 4
Increment: 0.01
*RST: 0
Default unit: symbol

Example: `BB:DM:PRAM:RDEL 1`
The rising edge starts 1 symbol later.

Manual operation: See "Rise Delay" on page 77

[:SOURce<hw>]:BB:DM:PRAMp:SHAPe <Shape>

The command sets the edge shape of the ramp envelope.

Parameters:

<Shape> LINear | COSine

*RST: COSine

Example: `BB:DM:PRAM:SHAP COS`
selects a cosine-shaped rise and fall for the transmitted power edge.

Manual operation: See "Ramp Function" on page 77

[:SOURce<hw>]:BB:DM:PRAMp:TIME <Time>

Sets the power ramping rise time and fall time for a burst.

Parameters:

<Time> float

Range: 0.25 to 16
Increment: 0.01
*RST: 1
Default unit: symbol

Example: `BB:DM:PRAM:TIME 2`
sets a time of 2 symbols for the edges to rise and fall.

Manual operation: See "Ramp Time" on page 77

[:SOURce<hw>]:BB:DM:PRAMp[:STATe] <State>

The command enables or disables power ramping.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:DM:PRAM:STAT ON`
switches power ramping on.

Manual operation: See "State" on page 76

12.9.2 DM Lists

The following section brings together the commands for defining and managing the data lists and control lists for digital modulation.

Lists are stored as files with specific file extensions in a user-definable directory (see Table 12-1). To specify the default directory, use the command `:MMEMory:CDIRectory`. To access files in this directory, enter their file name; the path and extension are optional.

Table 12-1: Overview of the used list types

List type	Content	File extension
Data List	Digital modulation data	`*.dm_iqd`
Control List	Digital modulation control data	`*.dm_iqc`
User Standards	User settings of digital modulation	`*.dm_stu`
Usr Filter	Usr filter settings	`*.vaf`
User Mapping	User mapping settings	`*.vam`

It is not possible to use other file extensions with the commands. Attempting to do so will cause an error message. If the file extension is changed in any other way (e.g. by directly accessing the file system) the lists are no longer recognized and therefore invalid.

The CLISt commands always create a binary control list, even if ASCII format is specified.

Therefore, to create a *.wv file with R&S WinIQSIM2 using a control list, use the input control list in ASCII format. Refer to the example in "To create a control list in binary format" on page 275 for more information.

List of Commands

[:SOURce<hw>]:BB:DM:CLISt:CATalog?
[:SOURce<hw>]:BB:DM:DLISt:CATalog?

[:SOURce<hw>]:BB:DM:FLISt:CATalog?
[:SOURce<hw>]:BB:DM:MLISt:CATalog?

Queries the user mapping lists present in the default directory. The default directory is set using command `MMEM:CDIRectory`. When the names of the files are returned they are separated by commas.

The command only reads out files with the `*.vam` extension (see also Table 12-1).

Return values:

<Catalog> string

Example: `BB:DM:FORM USER`
selects the User modulation type.
`MMEM:CDIR 'D:\user\mapping_List'`
sets the default directory for the user mapping lists.
`BB:DM:MLIS:CAT?`
queries the user mapping lists in the directory.

Usage: Query only

Manual operation: See "Load User Mapping" on page 73

[:SOURce<hw>]:BB:DM:CLISt:FREE?
[:SOURce<hw>]:BB:DM:DLISt:FREE?
[:SOURce<hw>]:BB:DM:FLISt:FREE?
[:SOURce<hw>]:BB:DM:MLISt:FREE?

Queries the user modulation mapping list free memory.

Return values:

<Free> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:DM:FORM USER`
selects the User modulation type.
`BB:DM:FLIS:FREE?`
queries the free memory.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:COPY <Filename>
[:SOURce<hw>]:BB:DM:DLISt:COPY <Filename>

The command copies the selected data list into the data list specified by <list name>. If a data list with the specified name already exists, it is overwritten. If it does not yet exist, it is created.

The source file has to be available in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can be specified, in which case the source file is copied into the file <list name> in the specified directory. The file extension may be omitted.

Only files with the file extension `*.dm_iqc` will be copied (see also Table 12-1).

Setting parameters:
<Filename> string

Example: `BB:DM:DLIS 'd_list1'`
selects data list `d_list1`.
`BB:DM:DLIS:COPY 'd_list2'`
copies the content of data list `d_list1` into data list `d_list2`.
Any existing content in data list `d_list2` is overwritten.

Usage: Setting only

Manual operation: See "Select Data List" on page 71

[:SOURce<hw>]:BB:DM:CLISt:TAG?
[:SOURce<hw>]:BB:DM:DLISt:TAG?

The command queries the content of the specified tag in the selected file.

Return values:
<Tag> <control list>,<tag name>

Example: `BB:DM:DLIS:TAG 'D_list1','date'`
queries the Date tag in control list `D_list1`.
Response:`10.10.2008`
the data list was created on 10.10.2008.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:DATA <Data>

Sends the data to the currently selected control list. If the list already contains data, it is overwritten. This command only writes data into the data section of the file.

The values for the control signals are sent, arranged in an 8-bit value as follows:

Signal	Order	Decimal value of bits
Marker 1 Marker 2	LSBit	1 2
Burst =	LSBit	16
LevAtt1	LSBit	32
CWMod	LSBit	64
Hop	MSBit	128

You can also send the data as a binary block. Each binary block is a 2-byte value in which the 16 bits represent the binary values (16-bit unsigned integer, 2 bytes, LSB first).

For query purposes, the command `:FORMat ASCii | PACKed` can be used to switch between the formats. The byte sequence is defined in the IEC bus standard as 'most significant byte first'

`*RST` has no effect on data lists. This command is without query.

Setting parameters:

<Data> string

Example: `BB:DM:CLIS:SEL 'c_list1'`
Selects the control list.
`BB:DM:CLIS:DATA 0,0,0,0,8,8,8,0,0,0,0...`
Enters the control values in list c_list1. In the example, only ramps for marker 4 are set.

Usage: Setting only

Manual operation: See "Select Ramp to Edit" on page 82

[:SOURce<hw>]:BB:DM:CLISt:DELete <Filename>

The command deletes the specified control list from the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are deleted. The file extension may be omitted. Only files with the file extension `*.dm_iqc` will be deleted.

Setting parameters:

<Filename> <list name>

Example: `BB:DM:CLIS:DEL 'c_list3'`
deletes control list c_list3.

Usage: Setting only

Manual operation: See "Select Control List" on page 71

[:SOURce<hw>]:BB:DM:CLISt:POINts?

The command queries the number of lines (2 bytes) in the currently selected list.

Return values:

<Points> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:DM:CLIS:SEL "c_list1"`
selects control list c_list1.
`BB:DM:CLIS:POIN?`
queries the number of lines in the control list.
Response: `20`
the control list consists of 20 lines.

Usage: Query only

[:SOURce<hw>]:BB:DM:CLISt:SELect <Filename>

The command selects the control list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a control list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.dm_iqc` will be selected or created.

Parameters:

<Filename>	<list name>
Example:	`BB:DM:CLIS:SEL 'c_list1'` selects control list `c_list1`.
Manual operation:	See "Select Control List" on page 71

[:SOURce<hw>]:BB:DM:DLISt:DATA <Data>
[:SOURce<hw>]:BB:DM:DLISt:DATA? [<Start>[, <Count>]]

The **Setting** command sends the bit data to the data list selected with the command `:BB:DM:DLISt:SELect`. Any existing content in the data list is overwritten. This command only writes data into the data section of the file.

This command sends the bit data to the selected data list, which is overwritten.

The **query** reads out the data part of the data list. If the query is expanded by using the two parameters <start> and <length>, the list is read out in smaller sections. Start and Length are expressed in bits. Without the parameters, the total length is always read out starting from address 1. The command `:FORMat ASCii | PACKed` can be used to select the data format. The byte sequence is defined in the IEC bus standard (read/write most significant byte first). `*RST` has no effect on data lists.

Parameters:

<Data>	integer

Query parameters:

<Start>	integer Range: 1 to 2147483647
<Count>	integer Range: 1 to 2147483647

Example:	`BB:DM:DLIS:SEL 'dlist1'`
	Selects data list dlist1. If the file does not yet exist, it is created.
	`BB:DM:DLIS:DATA 1,1,1,0,0,0,1,1,0,1...`
	Sends the specified data to file dlist1. Any data already present is overwritten.

Example:	`BB:DM:DLIS:SEL 'dlist1'`
	Selects data list dlist1.
	`FORM ASCI`
	Selects ASCII data transmission format.
	`BB:DM:DLIS:DATA? 2048,1024`
	Queries the data starting at bit 2048 for 1024 bits.

[:SOURce<hw>]:BB:DM:DLISt:DATA:APPend <Bits>

The command appends the bit data onto the end of the existing data in the selected data list. This means that existing content in the data list is not overwritten. By this means very long data lists can be built up piecemeal. The data format is as specified in command `SOURce:BB:DM:DLISt:DATA`.

The command cannot be used with an empty data list, such as one that has just been created, for example. In this case the command `SOURce:BB:DM:DLISt:DATA` must first be used to enter modulation data in the list.

`*RST` has no effect on data lists.

Setting parameters:

<Bits> 0 | 1 {,0 | 1 } | block data

Example: `BB:DM:DLIS:SEL ' d_list2'`
selects data list `d_list2`.
`FORM ASC`
selects ASCII data transmission format.
`BB:DM:DLIS:DATA:APP 1,1,1,0,0,0,1,1,0,1...`
adds the specified numeric data to the existing data in data list `d_list2`.

Usage: Setting only

[:SOURce<hw>]:BB:DM:DLISt:DELete <Filename>

The command deletes the specified data list. from the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are deleted. The file extension may be omitted. Only files with the file extension `*.dm_iqd` will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:DLIS:DEL ' d_list2'`
deletes data list d_list2.

Usage: Setting only

Manual operation: See "Select Data List" on page 71

[:SOURce<hw>]:BB:DM:DLISt:POINts <Points>

The command defines the number of bits in the selected data list to be utilized. When a list is being filled with block data, this data is only ever sent in multiples of 8 bits. However the exact number of bits to be exploited can be set to a different figure. The superfluous bits in the list are then ignored.

Parameters:

<Points> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:DM:DLIS:POIN 234`
defines the number of bits in the data list to be utilized as 234 bits. If the list was filled with block data, at least the last 6 bits will be ignored.

[:SOURce<hw>]:BB:DM:DLISt:SELect <Select>

The command selects the data list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a data list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.dm_iqd` will be selected or created.

The modulation data in this data list is used when the data source is selected using the command `SOURce:BB:DM:SOURce DLISt`.

Parameters:

<Select> <list name>

Example: `BB:DM:DLIS:SEL 'd_list2'`
selects data list `d_list2`.

Manual operation: See "Data Source" on page 71

[:SOURce<hw>]:BB:DM:FLISt:DELete <Filename>

Deletes the secified user filter file. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.vaf`*.vaf will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`MMEM:CDIR 'D:\user\Filter_List'`
sets the default directory for the user-defined filters.
`BB:DM:FLIS:DEL user_filter3`
deletes the user-defined filter.

Usage: Setting only

Manual operation: See "Load User Filter" on page 76

[:SOURce<hw>]:BB:DM:FLISt:POINts?

Queries the user filter list length.

Return values:

<Points> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`BB:DM:FLIS:POIN?`
queries the list length.

Usage: Query only

[:SOURce<hw>]:BB:DM:FLISt:SELect <Filename>

Selects the user-defined filter (`*.vaf`).

The directory applicable to the following command is defined with the command `MMEM:CDIRectory`. To access the files in this directory, only the file name is required, without the path and the file extension.

Parameters:

<Filename> string

Example: `BB:DM:FILT:TYPE USER`
selects the User filter type.
`MMEM:CDIR 'D:\user\Filter_List'`
sets the default directory for the user-defined filters.
`BB:DM:FLIS:SEL user_filter3`
selects the user-defined filter.

Manual operation: See "Load User Filter" on page 76

[:SOURce<hw>]:BB:DM:MLISt:DELete <Filename>

Deletes the secified user mapping file. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.vam` will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:FORM USER`
selects the User modulation type.
`MMEM:CDIR 'D:\user\mapping_List'`
sets the default directory for the user mapping lists.
`BB:DM:MLIS:DEL user_mapping3`
deletes the user mapping list.

Usage: Setting only

Manual operation: See "Load User Mapping" on page 73

[:SOURce<hw>]:BB:DM:MLISt:POINts?

Queries the user modulation mapping list length.

Return values:

<Points> integer

Range: 0 to INT_MAX
*RST: 0

Example: `BB:DM:FORM USER`
selects the User modulation type.
`BB:DM:MLIS:POIN?`
queries the list length.

Usage: Query only

[:SOURce<hw>]:BB:DM:MLISt:SELect <Filename>

Selects the user mapping list in the default directory. The default directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are selected. If a user mapping list with the specified name does not yet exist, it is created. The file extension may be omitted. Only files with the file extension `*.vam` will be selected or created.

Parameters:

<Filename> string

Example: `BB:DM:MLIS:SEL 'c_list1'`
selects the user mapping list `c_list1`.

Manual operation: See "Load User Mapping" on page 73

[:SOURce<hw>]:BB:DM:SETTing:CATalog?

Reads out the files with Custom Digital Modulation settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.dm` will be listed.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:\user\dig_mod'`
sets the default directory.
`BB:DM:SETT:CAT?`
reads out all the files with Custom Digital Modulation settings in the default directory.
Response: `'DM_1'`
the file `DM_1` with Custom Digital Modulation settings is available.

Usage: Query only

Manual operation: See "Save/Recall" on page 68

[:SOURce<hw>]:BB:DM:SETTing:DELete <Filename>

This command deletes the selected file with Custom Digital Modulation settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm` will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:DEL 'DM_1'`
deletes file `DM_1`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 68

[:SOURce<hw>]:BB:DM:SETTing:LOAD <Filename>

This command loads the selected file with Custom Digital Modulation settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm` will be loaded.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:LOAD 'DM_1'`
loads file `DM_1`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 68

[:SOURce<hw>]:BB:DM:SETTing:STORe <Filename>

This command stores the current Custom Digital Modulation settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. User Standards are stored as files with the specific file extensions `*.dm`.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:STOR 'DM_QAM'`
stores the current Custom Digital Modulation settings into file `DM_QAM`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 68

[:SOURce<hw>]:BB:DM:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast> 0 | 1 | OFF | ON

*RST: 1

[:SOURce<hw>]:BB:DM:STANdard:ULISt:CATalog?

Reads out the files with Digital Standard settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.dm_stu` will be listed.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:\user\dig_mod'`
sets the default directory.
`BB:DM:STAN:ULIS:CAT?`
reads out all the files with Digital Standard settings in the default directory.
Response: `'DM_QAM'`
the file `DM_QAM` with Digital Standard settings is available.

Usage: Query only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:DELete <Filename>

Deletes the selected file with Digital Standard settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm_stu` will be deleted.

Setting parameters:

<Filename> string

Example: `BB:DM:STAN:ULIS:DEL 'DM_QAM'`
deletes file 'DM_QAM'.

Usage: Setting only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:LOAD <Filename>

Loads the selected file with Digital Standard settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.dm_stu` will be loaded.

Setting parameters:

<Filename>	string
Example:	`BB:DM:STAN:ULIS:LOAD 'DM_QAM'` loads file 'DM_QAM'.
Usage:	Setting only

[:SOURce<hw>]:BB:DM:STANdard:ULISt:STORe <Filename>

Stores the current Digital Standard settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. User Standards are stored as files with the specific file extensions `*.dm_stu`.

Setting parameters:

<Filename>	string
Example:	`BB:DM:STAN:ULIS:STOR 'DM_QAM'` stores the current Digital Standard settings into file 'DM_QAM'.
Usage:	Setting only

12.10 SOURce:BB:IMPort Subsystem

The `IMPort` subsystem contains the commands for importing IQ signal data files via TCP/IP into R&S WinIQSIM2.

12.10.1 General Commands

[:SOURce<hw>]:BB:IMPort:PRESet

Sets the default settings for IQ data import.

Example: `SOURce1:BB:IMPort:PRESet`

Usage: Event

Manual operation: See "Set to Default" on page 138

[:SOURce<hw>]:BB:IMPort:SERVer:ID?

Queries the connected import server. If no import server is connected, a message "not connected" is displayed.

Return values:
<Id> string

Example: `BB:IMP:SERV?`
queries the connected import server.

Usage: Query only

Manual operation: See "Server ID" on page 139

[:SOURce<hw>]:BB:IMPort:SERVer:LOCal[:STATe] <State>

Acivates/deactivates the local host as the server from where the IQ data streams will be dowloaded. If activated, the field for entering a remote host is filled with the entry "localhost" automatically and cannot be edited.

Parameters:
<State> 0 | 1 | OFF | ON
*RST: 1

Example: `BB:IMP:SERV:LOC:STAT ON`
sets the localhost as the connected server.

Manual operation: See "Use Local Server" on page 139

[:SOURce<hw>]:BB:IMPort:SERVer:NAME <Name>

The command enters the name (IP address) of the connected server. To enter the name in this field, the check box "Use Local Server" has to be unchecked.

Parameters:
<Name> string

Example: `BB:IMP:SERV:NAME 12.34.45.78`
enters the IP address of the connected server.

Manual operation: See "Server Name" on page 139

[:SOURce<hw>]:BB:IMPort:SERVer:PORT <Port>

Sets the port number for the server connection.

Parameters:

<Port> integer

Range: 0 to 65536
Increment: 1
*RST: 1000

Example: `BB:IMP:SERV:PORT 1000`
sets port 1000 for the server connection.

Manual operation: See "Port Number" on page 139

[:SOURce<hw>]:BB:IMPort:SERVer:TTOut <TTout>

Sets the duration after which the transmission is aborted in case of a server timeout.

Parameters:

<TTout> integer

Range: 10 to 10000
Increment: 1
*RST: 20

Example: `BB:IMP:SERV:TTO 20`
sets the transmission timeout to 20 seconds.

Manual operation: See "Transmission Timeout (s)" on page 139

[:SOURce<hw>]:BB:IMPort:SETTing:CATalog?

Reads out the files with import settings in the default directory. The default directory is set using command `MMEM:CDIRectory`. Only files with the file extension `*.import` will be listed.

Return values:

<Catalog> string

Example: `MMEM:CDIR 'D:\user\/user/waveform'`
sets the default directory to D:\user\/user/waveform.
`BB:IMP:SETT:CAT?`
reads out all the files with import settings in the default directory.
Response: 'import_1', 'import_2'
the files `import_1` and `import_2` with import settings are available.

Usage: Query only

Manual operation: See "Save/Recall" on page 139

[:SOURce<hw>]:BB:IMPort:SETTing:DELete <Filename>

Deletes the selected file with import settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.import` will be loaded.

Setting parameters:

<Filename> string

Example: `BB:IMP:SETT:DEL 'import_1'`
deletes file `import_1`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 139

[:SOURce<hw>]:BB:IMPort:SETTing:LOAD <Filename>

Loads the selected file with import settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.import` will be loaded.

Setting parameters:

<Filename> string

Example: `BB:IMP:SETT:LOAD 'import_1'`
loads file `import_1`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 139

[:SOURce<hw>]:BB:IMPort:SETTing:STORe <Filename>

Stores the current import settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. Import settings are stored as files with the specific file extensions `*.import`.

Setting parameters:

<Filename> string

Example: `BB:IMP:SETT:STOR 'import_1'`
stores the current import settings into file `import_1`.

Usage: Setting only

Manual operation: See "Save/Recall" on page 139

[:SOURce<hw>]:BB:IMPort:SETTing:STORe:FAST <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

Note: This function is not affected by the "Preset" function.

Parameters:

<Fast> 0 | 1 | OFF | ON

*RST: 1

[:SOURce<hw>]:BB:IMPort:SRATe:VARiation?

Queries the sample rate variation.

Return values:

<Variation> float

Range: 400 to 40E6
Increment: 0.001
*RST: 20E6

Example: `BB:IMPort:SRATe:VARiation?`

Usage: Query only

Manual operation: See "Sample Rate Variation " on page 141

[:SOURce<hw>]:BB:IMPort:STATe <State>

Enables/disables the import of IQ data.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `BB:IMP:STAT ON`
enables the import of IQ data.

Manual operation: See "State" on page 138

12.10.2 Filter/Clipping Settings

[:SOURce<hw>]:BB:IMPort:CLIPping:LEVel <Level>

Sets the limit for level clipping. This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Level clipping is activated with the command `SOUR:BB:IMP:CLIP:STAT ON`.

Parameters:

<Level> integer

Range: 1 to 100
Increment: 1
*RST: 100
Default unit: PCT

Example: `BB:IMP:CLIP:LEV 80PCT`
sets the limit for level clipping to 80% of the maximum level.
`BB:IMP:CLIP:STAT ON`
activates level clipping.

Manual operation: See "Clipping Level " on page 143

[:SOURce<hw>]:BB:IMPort:CLIPping:MODE <Mode>

The command sets the method for level clipping (Clipping).

Parameters:

<Mode> VECTor | SCALar

VECTor
The reference level is the amplitude | i+jq |.

SCALar
The reference level is the absolute maximum of the I and Q values.

*RST: VECTor

Example: `BB:IMP:CLIP:MODE SCAL`
selects the absolute maximum of all the I and Q values as the reference level.
`BB:IMP:CLIP:LEV 80PCT`
sets the limit for level clipping to 80% of this maximum level.
`BB:IMP:CLIP:STAT ON`
activates level clipping.

Manual operation: See "Clipping Mode" on page 144

[:SOURce<hw>]:BB:IMPort:CLIPping:STATe <State>

Activates level clipping (Clipping). The value is defined with the command `[SOURce]:BB:IMPort:CLIPping:LEVel`, the mode of calculation with the command `[SOURce]:BB:IMPort:CLIPping:MODE`.

Parameters:
<State> 0 | 1 | OFF | ON
*RST: 0

Example: `BB:IMP:CLIP:STAT ON`
activates level clipping.

Manual operation: See "Clipping State" on page 143

[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth <ILength>

Sets the impulse length (number of filter tabs).

Parameters:
<ILength> integer
Range: 1 to 128
Increment: 1
*RST: 40

Example: `BB:IMP:FILT:ILEN 10`
sets the number of filter tabs to 10.

Manual operation: See "Impulse Length" on page 142

[:SOURce<hw>]:BB:IMPort:FILTer:ILENgth:AUTO <Auto>

Acivates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

Parameters:
<Auto> 0 | 1 | OFF | ON
*RST: 1

Example: `BB:IMP:FILT:ILEN:AUTO ON`
the most sensible parameters are selected automatically.

Manual operation: See "Impulse Length" on page 142

[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling <OSampling>

Sets the upsampling factor.

Parameters:

<OSampling> integer

Range: 1 to 32
Increment: 1
*RST: 2

Example: `BB:IMP:FILT:OSAM 32`
sets the upsampling factor to 32.

Manual operation: See "Oversampling" on page 142

[:SOURce<hw>]:BB:IMPort:FILTer:OSAMpling:AUTO <Auto>

Acivates/deactivates the upsampling factor state. If activated, the most sensible parameter values are selected. The value depends on the coherence check. If deactivated, the values can be changed manually.

Parameters:

<Auto> 0 | 1 | OFF | ON

*RST: 1

Example: `BB:IMP:FILT:OSAM:AUTO ON`
the most sensible parameters are selected automatically.

Manual operation: See "Oversampling" on page 142

[:SOURce<hw>]:BB:IMPort:FILTer:TYPE <Type>

Selects the filter type.

Parameters:

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 | COEQualizer | COFequalizer | C2K3x | APCO25 | SPHase | RECTangle | PGAuss | LPASs | DIRac | ENPShape | EWPShape | LPASSEVM

*RST: GAUSs (if layer mode OFDM), COSine (if layer mode CCK or PBCC)

Example: `BB:IMP:FILT:TYPE COS`
selects the Cosine filter type.

Manual operation: See "Filter" on page 141

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine <Cosine>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:GAUSs <Gauss>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASs <LPass>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:LPASSEVM <LPassEvm>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:PGAuss <PGauss>

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:RCOSine <RCosine>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:SPHase <SPhase>
[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:APCO25 <Apco25>

Sets the corresponding filter parameter.

Parameters:

<Apco25> float
Range: 0.05 to 0.99
Increment: 0.01
*RST: 0.2

<Cosine> float
Range: 0 to 1.0
Increment: 0.01
*RST: 0.1

<Gauss> float
Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.5

<LPass> float
Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<LPassEvm> float
Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

<PGauss> float
Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.5

<RCosine> float
Range: 0 to 1.0
Increment: 0.01
*RST: 0.22

<SPhase> float
Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

Example: `BB:IMP:FILT:PAR:APCO25 0.2`
sets the roll-off factor to 0.2 for filter type APCO25.

Manual operation: See "Roll Off Factor or BxT" on page 141

[:SOURce<hw>]:BB:IMPort:FILTer:PARameter:COSine:COFS <Cofs>

Sets the "cut of frequency shift" value for the Cosine filter type.

Parameters:

<Cofs> float

Range: -1 to 1
Increment: 0.01
*RST: -0.1

Example: `BB:IMP:FILT:PAR:COS:COFS 0.04`
the "cut of frequency shift" value is set to 0.04.

Manual operation: See "Cut Off Frequency Shift" on page 141

12.10.3 Marker Settings

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:MODE <Mode>

The command defines the signal for the selected marker output.

Parameters:

<Mode> RESTart | PULSe | PATTern | RATio

RESTart
A brief marker signal is generated at the start of the waveform.

PULSe
A pulsed marker signal is generated. The pulse frequency (= symbol rate/divider) is defined with the `SOUR:IMP:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:IMP:TRIG:OUTP:PULSe:FREQuency` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the command `SOURce:BB:IMP:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:IMP:TRIGger:OUTPut:OFFT` and `SOURce:BB:IMP:TRIGger:OUTPut:ONT` is generated.

*RST: RESTart

Example: `BB:IMP:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

Manual operation: See "Marker Mode" on page 144

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:ONTime <OnTime>
[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:OFFTime <OffTime>

Sets the number of symbols in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:IMPort:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

Parameters:

<OffTime> integer

Range: 1 to 16777215
Increment: 1
*RST: 1

Example: `BB:IMP:TRIG:OUTP2:OFFT 20`
sets an OFF time of 20 symbols for marker signal 2.

Manual operation: See "Marker Mode" on page 144

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PATTern <Pattern>

The command defines the bit pattern used to generate the marker signal in the setting `SOURce:BB:IMP:TRIGger:OUTPut:MODE PATTern`. 0 is marker off, 1 is marker on.

Parameters:

<Pattern> integer

Range: #B0,1 to #B111
*RST: #B0,1

Example: `BB:IMP:TRIG:OUTP2:PATT #B000000011111111,15`
sets a bit pattern.
`BB:IMP:TRIG:OUTP2:MODE PATT`
activates the marker signal according to a bit pattern on output MARKER 2.

Manual operation: See "Marker Mode" on page 144

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

Sets the divider for the Pulse marker mode (`SOUR:BB:IMP:TRIG:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider> integer

Range: 2 to 1024
Increment: 1
*RST: 2

Example: `BB:IMP:TRIG:OUTP2:PULS:DIV 2`
sets the divider to 2 for the marker signal on output MARKER 2.
`BB:IMP:TRIG:OUTP2:FREQ?`
queries the resulting pulse frequency of the marker signal.
Response:
`66 000`
the resulting pulse frequency is 66 kHz.

Manual operation: See "Marker Mode" on page 144

[:SOURce<hw>]:BB:IMPort:TRIGger:OUTPut<ch>:PULSe:FREQuency?

Queries the pulse frequency of the pulsed marker signal in the setting `SOURce:BB:IMP:TRIGger:OUTPut:MODE PULSe`. The pulse frequency is derived by dividing the symbol rate by the divider.

Return values:

<Frequency> float

Range: 0 to max
Increment: 0.001
*RST: 10E6

Example: `BB:IMP:TRIG:OUTP2:PULS:DIV 2`
sets the divider marker signal on output MARKER 2 to the value 2.
`BB:IMP:TRIG:OUTP2:MODE PULS`
enables the pulsed marker signal.
`BB:IMP:TRIG:OUTP2:PULS:FREQ?`
queries the pulse frequency of the marker signal.
Response:
`33 000`
the resulting pulse frequency is 33 kHz.

Usage: Query only

Manual operation: See "Marker Mode" on page 144

12.11 SOURce:BB:MCCW Subsystem

This subsystem contains the commands for setting the Multi Carrier CW signals.

12.11.1 Suffixes

SOURce<hw>

For one-path instruments, the keyword SOURce is optional and can be omitted.

The numeric suffix to `SOURce` distinguishes between signal generation for path A and path B in the case of two-path instruments:

- `SOURce[1]` = path A
  The keyword `SOURce` is optional and can be omitted
- `SOURce2` = path B
  The keyword `SOURce` is mandatory, i.e. the command must contain the keyword with suffix 2.

OUTPut<ch>

The numeric suffix to `OUTPut` distinguishes between the available markers.

12.11.2 General Settings and Carrier Setup Settings

[:SOURce<hw>]:BB:MCCW:STATe <State>

Activates the multi carrier CW signal generation and deactivates all the other digital standards and digital modulation modes in the same path.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 0

Example: `SOURce1:BB:MCCW:STATe ON`

Manual operation: See "State" on page 100

[:SOURce<hw>]:BB:MCCW:PRESet

Sets the multi carrier signal parameters to their default values (`*RST` values specified for the commands).

Not affected is the state set with the command `SOURce<hw>:BB:MCCW:STATe`

Example: `SOURce1:BB:MCCW:PRESet`

Usage: Event

Manual operation: See "Set to Default" on page 101

[:SOURce<hw>]:BB:MCCW:CARRier:COUNt <Count>

Sets the number of carriers in the Multi Carrier CW signal. The total bandwidth is calculated as (Number of carriers - 1) * Carrier spacing and must not exceed the system bandwidth of the instrument (see data sheet).

The carrier spacing (`:BB:MCCW:CARRier:SPACing`) is reduced if the total bandwidth is not respected when entering the number of carriers.

The number of carriers entered therefore defines the maximum carrier spacing (`:BB:MCCW:CARRier:SPACing`).

Parameters:

<Count> integer

Range: 1 to 8192

*RST: 64

Example: `BB:MCCW:CARR:COUN 10`

sets 10 CW carriers for the multi carrier signal.

Manual operation: See "Number of Carriers" on page 101

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe <Phas0[,Phas1..]>]
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:PHASe? [<Start>[, <Count>]]

Sets the start phase of the carrier with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:

<Phas0[,Phas1..]> float

Range: 0 to 360
Increment: 0.01
*RST: 0
Default unit: DEG

Query parameters:

<Start> integer

Range: 0 to lastCarrier

<Count> integer

Range: 1 to lastCarrier

Return values:

<Phas[,Phas..]> float

Example: `BB:MCCW:CARR:LIST:PHAS 1,20,30,40,50,60`
sets a start phase for carriers 0, 1, 2, 3 and 4.
`BB:MCCW:CARR:LIST:PHAS? 2,3`
queries the phase of carrier 1, 2 and 3.
Response: `20,30,40`

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer <Pow0[,Pow1..]>]
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:POWer? <Start>, <Count>

Sets the power of the carrier with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:

<Pow0[,Pow1..]> float

Increment: 0.01

Query parameters:

<Start> integer

Range: 0 to lastCarrier

<Count> integer

Range: 1 to lastCarrier

Return values:

<Pow[,Pow..]> float

Example: `BB:MCCW:CARR:LIST:POW -65 dB, -30 dB, -50 dB,...`
sets the power of carrier 0 to -65 dB, carrier 1 to -30 dB and so on.
`BB:MCCW:CARR:LIST:POW? 2,3`
queries the power of carrier 1 and 2.
Response: `-30,-50`

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe <Stat0[,Stat1..]>
[:SOURce<hw>]:BB:MCCW:CARRier:LIST:STATe? <Start>, <Count>

Switches the carrier on or off with the aid of a value list. The first value in the list is assigned to the carrier with index 0, the second value to the carrier with index 1, etc. The maximum length corresponds to the maximum number of multi carriers. There is no need to enter all the values every time. Values not set by the value list are set with the default values provided they have already been explicitly set by a previous command. If this is the case, the values continue to apply until overwritten.

Setting parameters:
<Stat0[,Stat1..]> ON | OFF[,ON | OFF,...]

Query parameters:
<Start> integer
Range: 0 to lastCarrier

<Count> integer
Range: 1 to lastCarrier

Return values:
<State[,State..]> select

Example: `BB:MCCW:CARR:LIST:STAT 1,0,1,1,0,0....`
switches carrier 0 on, carrier 1 off, etc.
`BB:MCCW:CARR:LIST:STAT? 2,4`
queries the states of carrier 2, 3, 4 and 5.
Response: `0,1,1,0`

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CARRier:PHASe <CarrierIndex>, <Phase>

Sets the start phase of the selected carrier.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:
<CarrierIndex> integer
Range: 0 to lastCarrier

<Phase> float

Sets the start phase of the selected carrier.

Range: 0 to 359.99
Increment: 0.01
*RST: 0
Default unit: DEG

Example: `BB:MCCW:CARR:PHAS 15,90`
sets a start phase of 90 DEG for carrier 15.

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CARRier:POWer <CarrierIndex>, <Power>

Sets the power of the selected carrier.

Parameters:

<CarrierIndex> integer

Range: 0 to lastCarrier

<Power> float

Sets the power of the selected carrier.

Range: -80 to 0
Increment: 0.01
*RST: 0

Example: `BB:MCCW:CARR:POW 15, -50 dB`
sets the power of carrier 15 to -50 dB.

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CARRier:SPACing <Spacing>

The command sets the carrier spacing.

The carriers are generated symmetrically around the center carrier. The total bandwidth is calculated as ("Number of carriers" - 1) * "Carrier spacing "and must not exceed the system bandwidth of the instrument (see data sheet).

The maximum carrier spacing that can be set is dependent on the chosen number of carriers

The maximum carrier spacing is automatically reduced so that the maximum total bandwidth is not exceeded on entering the number of carriers (`:BB:MCCW:CARRier:COUNt`).

Parameters:

<Spacing> float

Range: 0 to 120E6
Increment: 0.01
*RST: 10E3

Example: `BB:MCCW:CARR:SPAC 10 MHz`
sets a carrier spacing of 10 MHz.

Manual operation: See "Carrier Spacing" on page 101

[:SOURce<hw>]:BB:MCCW:CARRier:STATe <CarrierIndex>, <State>

Switches the selected carrier on or off.

The counting in remote control differs from the numbers in the carrier table. Index 0 corresponds to number 1 (first line) in the table. Therefore, switching the state of the channel via remote control always switches the state of <channel index> + 1 in the table.

Parameters:

<CarrierIndex> integer

Range: 0 to lastCarrier

<State> select

*RST: 0

Example: `BB:MCCW:CARR:STAT 15, ON`
switches carrier 16 on.

Manual operation: See "Carrier Table" on page 104

[:SOURce<hw>]:BB:MCCW:CFACtor <CFactor>

Sets the desired crest factor for the multi carrier signal on selection of the command `SOUR:BB:MCCW:CFAC:MODE SLOW`.

Parameters:

<CFactor> float

Range: 0 to 30
Increment: 0.01
*RST: 3

Example: `BB:MCCW:CFAC:MODE SLOW`
sets the Target Crest optimization mode.
`BB:MCCW:CFAC 10 dB`
sets the desired crest factor to 10 dB.

Manual operation: See "Desired Crest Factor" on page 102

[:SOURce<hw>]:BB:MCCW:CFACtor:ACTual?

Queries the actual Crest Factor for `BB:MCCW:CFAC:MODE SLOW`.

Return values:

<Actual> float

Range: 0 to 100
Increment: 0.01
*RST: 3

Example: `SOUR:BB:MCCW:CFAC:MODE SLOW`
`BB:MCCW:CFAC:ACT?`
Response: `3`

Usage: Query only

[:SOURce<hw>]:BB:MCCW:CFACtor:MODE <Mode>

Sets the mode by which automatic settings will minimize the crest factor or hold it at a chosen value.

Parameters:

<Mode> OFF | CHIRp | SLOW

OFF
Crest factor optimization is switched off. The carrier PHASE setting has an effect.

CHIRp
The crest factor is very rapidly optimized to < 3 dB for multi carrier signals so that all carriers are switched on and have the same amplitude. The computation time is independent of the number of carriers. In other carrier configurations the achievable crest factor is worse.

SLOW
The crest factor entered using `SOURceBB:MCCW:CFACtor` is maintained for all carrier configurations by means of automatic settings. The computation time depends on the number of carriers and on the crest factor that has been set. Computation time increases only when the number of carriers exceeds 256 and the crest factor is above 4 dB.

*RST: CHIRp

Example: `BB:MCCW:CFAC:MODE OFF`
switches off automatic crest factor optimization.
The setting `SOURBB:MCCW:CARR:PHAS` has an effect.

Manual operation: See "Optimize Crest Factor Mode" on page 101

[:SOURce<hw>]:BB:MCCW:CLOCk?

Queries the output clock rate. The output clock rate depends on the number of carriers an the selected carrier spacing.

Return values:

<Clock> float

Range: 0 to Max
Increment: 1E-3
*RST: 0

Example: `BB:MCCW:CLOC?`
queries the output clock rate.
Response: `256 000 000`
the output clock rate is 256 MHz.

Usage: Query only

Manual operation: See "Clock Frequency" on page 101

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:EXECute

Adopts the settings for the carrier range which has been defined using the `:BB:MCCW:EDIT:CARR:...` commands.

Example: `BB:MCCW:EDIT:CARR:STAR 4`
the carrier range starts at carrier 4.
`BB:MCCW:EDIT:CARR:STOP 400`
the carrier range stops at carrier 400.
`BB:MCCW:EDIT:CARR:STAT ON`
sets all the carriers in the carrier range to ON.
`BB:MCCW:EDIT:CARR:EXEC`
adopts the settings for all the carriers in the carrier range.
`BB:MCCW:STAT`
starts generation of the multi carrier signal. Carriers 4 to 400 are in the ON state.

Usage: Event

Manual operation: See "Accept" on page 104

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe:STEP <Step>

Sets the step width by which the start phases of the carriers in the defined carrier range will be incremented.

The phase settings are only valid if optimization of the crest factor is disabled (`:SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:

<Step> float

Range: -359.99 to 359.99
Increment: 0.01
*RST: 0

Example: `BB:MCCW:EDIT:CARR:PHAS 90 DEG`
sets a start phase of 90° for the carriers in the carrier range.
`BB:MCCW:EDIT:CARR:PHAS:STEP 1 DEG`
the start phase is incremented by 1° for each carrier, i.e. the first carrier has a start phase of 90°, the second a start phase of 91°, etc.

Manual operation: See "Phase Step" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:PHASe[:STARt] <Start>

Sets the start phase for the individual carriers in the defined carrier range. If the command `:BB:MCCW:EDIT:CARR:PHAS:STEP` is used to define a step width, the phase entered here applies only to the starting carrier. The phases of the remaining carriers are stepped up or down by the phase value specified in the `:BB:MCCW:EDIT:CARR:PHAS:STEP` command.

The phase settings are only valid if optimization of the crest factor is disabled (`SOURce:BB:MCCW:CFACtor:MODE OFF`).

Parameters:

<Start> float

Range: 0 to 359.99
Increment: 0.01
*RST: 0

Example: `BB:MCCW:EDIT:CARR:PHAS 90 DEG`
sets a start phase of 90° for the carriers in the carrier range.

Manual operation: See "Phase Start" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer:STEP <Step>

Sets the step width by which the starting power of the carriers in the defined carrier range will be incremented.

Parameters:

<Step> float

Range: -80 to 80
Increment: 0.01
*RST: 0

Example: `BB:MCCW:EDIT:CARR:POW -80dB`
sets a power of -80 dB for the carriers in the carrier range.
`BB:MCCW:EDIT:CARR:POW:STEP 1 dB`
the power phase is incremented by dB for each carrier, i.e. the first carrier has -80 dB, the second -79 dB, etc.

Manual operation: See "Power Step" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:POWer[:STARt] <Start>

Sets the power for the individual carriers in the defined carrier range. If the command `:BB:MCCW:EDIT:CARR:POW:STEP` is used to define a step width, the power entered here applies only to the starting carrier. The power of the remaining carriers is stepped up or down by the power specified in the `:BB:MCCW:EDIT:CARR:POW:STEP` command.

Parameters:

<Start> float

Range: -80 to 0
Increment: 0.01
*RST: 0

Example: `BB:MCCW:EDIT:CARR:POW -50 dB`
sets the power of the carrier to -50 dB.

Manual operation: See "Power Start" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STARt <Start>

Selects the first carrier in the carrier range to which the settings with the `:BB:MCCW:EDIT:CARR:...` commands shall apply.

Parameters:

<Start> integer

Range: 0 to 8191
*RST: 0

Example: `BB:MCCW:EDIT:CARR:STAR 4`
the carrier range starts at carrier 4.

Manual operation: See "Carrier Start/Stop" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STATe <State>

Switches all the carriers in the selected carrier range on or off.

Parameters:

<State> 0 | 1 | OFF | ON

*RST: 1

Example: `BB:MCCW:EDIT:CARR:STAT ON`
sets all the carriers in the carrier range to ON.

Manual operation: See "Carrier State" on page 103

[:SOURce<hw>]:BB:MCCW:EDIT:CARRier:STOP <Stop>

Selects the last carrier in the carrier range to which the settings with the `:BB:MCCW:EDIT:CARR:...` commands shall apply.

Parameters:

<Stop>	integer Range: 0 to 8191 *RST: 0
Example:	`BB:MCCW:EDIT:CARR:STOP 40` the carrier range stops at carrier 40.
Manual operation:	See "Carrier Start/Stop" on page 103

12.11.3 Marker Settings

This section lists the remote control commands, necessary to configure the markers.

The marker delay settings are available for R&S SMx and R&S AMU instruments only.

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:MODE <Mode>

Defines the signal for the selected marker output.

Parameters:

<Mode> RESTart | PULSe | PATTern | RATio

RESTart
A brief marker signal is generated at the start of the waveform.

PULSe
A pulsed marker signal is generated. The pulse frequency (= sample rate/divider) is defined with the `SOUR:BB:MCCW:TRIG:OUTP:PULSe:DIVider` command and can be queried with the `SOUR:BB:MCCW:TRIG:OUTP:PULSe:FREQuency?` command.

PATTern
A marker signal is generated with the aid of a user-definable bit pattern. The bit pattern is entered with the aid of command `SOURce:BB:MCCW:TRIGger:OUTPut:PATTern`. The bit pattern is a maximum of 32 bits long.

RATio
A regular marker signal corresponding to the Time Off / Time On specifications in the commands `SOURce:BB:MCCW:TRIGger:OUTPut:OFFT` and `SOURce:BB:MCCW:TRIGger:OUTPut:ONT` is generated.

*RST: RESTart

Example: `BB:MCCW:TRIG:OUTP2:MODE PULS`
selects the pulsed marker signal on output MARKER 2.

Manual operation: See "Marker Mode" on page 106

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:OFFTime <OffTime>
[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:ONTime <OnTime>

Sets the number of samples in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:MCCW:TRIGger:OUTPut:MODE RATio` on the marker outputs is ON.

Parameters:

<OnTime> integer

Range: 1 to 16777215
*RST: 1

Example: `BB:MCCW:TRIG:OUTP2:ONT 20`
sets an ON time of 20 samples for marker 2.

Manual operation: See "Marker Mode" on page 106

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PATTern <Pattern>

Sets the bit pattern used to generate the marker signal in the setting `SOURce:BB:MCCW:TRIGger:OUTPut:MODE PATTern` 0 is marker off, 1 is marker on.

Parameters:

<Pattern>	<32 bit pattern> *RST: 0

Example:
`BB:MCCW:TRIG:OUTP2:PATT #HE0F52,20`
sets a bit pattern.
`BB:MCCW:TRIG:OUTP2:MODE PATT`
activates the marker signal according to a bit pattern on output MARKER 2.

Manual operation: See "Marker Mode" on page 106

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:DIVider <Divider>

Sets the divider for Pulse marker mode (`SOUR:BB:MCCW:TRIG:OUTP:MODE PULSe`). The resulting pulse frequency is derived by dividing the symbol rate by the divider.

Parameters:

<Divider>	integer Range: 2 to 1024 *RST: 2

Example:
`BB:MCCW:TRIG:OUTP2:PULS:DIV 2`
sets the divider for the marker signal on output MARKER 2 to the value 2.
`BB:MCCW:TRIG:OUTP2:FREQ?`
queries the resulting pulse frequency of the marker signal
`Response: 66 000`
the resulting pulse frequency is 66 kHz.

Manual operation: See "Marker Mode" on page 106

[:SOURce<hw>]:BB:MCCW:TRIGger:OUTPut<ch>:PULSe:FREQuency?

Queries the pulse frequency of the pulsed marker signal in the setting `SOUR:BB:MCCW:TRIG:OUTP:MODE PULS`. The pulse frequency is derived by dividing the symbol rate by the divider. The divider is defined with command `SOUR:BB:MCCW:TRIG:OUTP:PULS:DIV`.

Return values:

<Frequency>	float Increment: 1E-3

Example:
`BB:MCCW:TRIG:OUTP2:PULS:DIV 4`
sets the divider for the marker signal on output MARKER 2 to the value 4.
`BB:MCCW:TRIG:OUTP2:MODE PULS`
enables the pulsed marker signal
`BB:MCCW:TRIG:OUTP2:PULS:FREQ?`
queries the pulse frequency for the marker signal.
`Response: "33 000"`
the resulting pulse frequency is 33 kHz.

Usage: Query only

Manual operation: See "Marker Mode" on page 106

12.12 SOURce:BB:PROGress Subsystem General Commands

In the R&S WinIQSIM2, some calculation processes may take longer time. While operating the instrument manually, you can observe the status of an initiated process by the busy indicator. The following commands fulfill the same task in the remote control operation.

Example: Querying the status of the Create Waveform file process

The following is an example on how to use these commands to retrieve information about how many percent of the initiated process are completed.

```
:SOURce1:BB:EUTRa:SETTing:TMOD:DL "E-TM1_1__15MHz"
:SOURce1:BB:EUTRa:SLENgth 100
:SOURce1:BB:PROGress:MCODer?
// 100 (task compleated)
:SOURce1:BB:EUTRa:STATe ON
:BB:PROGress:MCODer?
// 67 (task in progress)
:SOURce1:BB:EUTRa:WAVeform:CREate
:SOURce1:BB:PROGress:MCODer?
// 25 (task in progress)
// :SOURce1:BB:PROGress:MCODer:DM:FILTer?
// 100
```


[:SOURce<hw>]:BB:PROGress:MCODer?
[:SOURce<hw>]:BB:PROGress:MCODer:ARBitrary:MCARrier?
[:SOURce<hw>]:BB:PROGress:MCODer:ARBitrary:WSEGment?

Queries the status of an initiated process, like for example the calculation of a signal in accordance to a digital standard, or the calculation of a multi carrier or multi segment waveform file.

Return values:

<WSegment> integer

Indicates the task progress in percent

Range: 0 to 100
*RST: 100

Example: see Example "Querying the status of the Create Waveform file process" on page 377

Usage: Query only

[:SOURce<hw>]:BB:PROGress:MCODer:DM:FILTer?

Queries the status of an applied offline filtering, like for example during the calculation of a waveform and a multi carrier waveform file.

Return values:

<Filter> integer

Indicates the task progress in percent

Range: 0 to 100
*RST: 100

Example: see Example "Querying the status of the Create Waveform file process" on page 377

Usage: Query only

12.13 STATus Subsystem

This system contains the commands for the status reporting system. See also Chapter A.1.1.5, "Status Reporting System", on page 410 for detailed information.

`*RST` on page 254 has no effect on the status registers.

Value ranges

- Queries return the current value of the respective register, which permits a check of the device status.
  Return values: A decimal value in the range 0 to 32767 (=2^{15}-1)
- The configuration commands set the respective register thus determining which status changes of the R&S WinIQSIM2 cause the status registers to be changed.
  Setting values: A decimal value in the range 0 to 32767 (=2^{15}-1)

:STATus:OPERation:CONDition <Condition>

Sets the content of the CONDition part of the STATus:OPERation register. This part contains information on the action currently being performed in the instrument. The content is not deleted after being read out because it indicates the current hardware status.

Parameters:

<Condition> string

Example: `:STATus:OPERation:CONDition?`
queries the Status:Operation:Condition register.

:STATus:OPERation:ENABle <Enable>

Sets the bits of the ENABle part of the STATus:OPERation register. This setting determines which events of the Status-Event part are forwarded to the sum bit in the status byte. These events can be used for a service request.

Parameters:

<Enable> string

Example: `:STAT:OPER:ENAB 32767`
all events are forwarded to the sum bit of the status byte.

:STATus:OPERation[:EVENt] <Event>

Queries the content of the EVENt part of the STATus:OPERation register. This part contains information on the actions performed in the instrument since the last readout. The content of the EVENt part is deleted after being read out.

Parameters:

<Event> string

Example: `:STAT:OPER:EVEN?`
queries the STATus:OPERation:EVENt register.

:STATus:OPERation:NTRansition <Ntransition>

Sets the bits of the NTRansition part of the STATus:OPERation register. If a bit is set, a transition from 1 to 0 in the condition part causes an entry to be made in the EVENt part of the register. The disappearance of an event in the hardware is thus registered, for example the end of an adjustment.

Parameters:

<Ntransition> string

Example: `:STAT:OPER:NTR 0`
a transition from 1 to 0 in the condition part of the Status:Operation register does not cause an entry to be made in the EVENt part.

:STATus:OPERation:PTRansition <Ptransition>

Sets the bits of the PTRansition part of the STATus:OPERation register. If a bit is set, a transition from 0 to 1 in the condition part causes an entry to be made in the EVENt part of the register. A new event in the hardware is thus registered, for example the start of an adjustment.

Parameters:

<Ptransition> string

Example: `:STAT:OPER:PTR 32767`
all transitions from 0 to 1 in the condition part of the Status:Operation register cause an entry to be made in the EVENt part.

:STATus:PRESet <Preset>

Resets the status registers. All PTRansition parts are set to FFFFh (32767), i.e. all transitions from 0 to 1 are detected. All NTRansition parts are set to 0, i.e. a transition from 1 to 0 in a CONDition bit is not detected. The ENABle parts of STATus:OPERation and STATus:QUEStionable are set to 0, i.e. all events in these registers are not passed on.

Parameters:

<Preset> string

Example: `STAT:PRES`
resets the status registers.

:STATus:QUEStionable:CONDition <Condition>

Queries the content of the CONDition part of the STATus:QUEStionable register. This part contains information on the action currently being performed in the instrument. The content is not deleted after being read out since it indicates the current hardware status.

Parameters:

<Condition> string

Example: `:STATus:QUEStionable:CONDition?`
queries the Status:Questionable:Condition register.

:STATus:QUEStionable:ENABle <Enable>

Sets the bits of the ENABle part of the STATus:QUEStionable register. This setting determines which events of the Status-Event part are enabled for the sum bit in the status byte. These events can be used for a service request.

Parameters:

<Enable> string

Example: `STAT:OPER:ENAB 1`
problems when performing an adjustment cause an entry to be made in the sum bit.

:STATus:QUEStionable[:EVENt] <Event>

Queries the content of the EVENt part of the `STATus:QUEStionable` register. This part contains information on the actions performed in the instrument since the last readout. The content of the EVENt part is deleted after being read out.

Parameters:

<Event> string

Example: `STAT:QUES:EVEN?`
queries the Status:Questionable:Event register.

:STATus:QUEStionable:NTRansition <Ntransition>

Sets the bits of the NTRansition part of the STATus:QUEStionable register. If a bit is set, a transition from 1 to 0 in the condition part causes an entry to be made in the EVENt part of the register.

Parameters:

<Ntransition> string

Example: `STAT:OPER:NTR 0`
a transition from 1 to 0 in the condition part of the Status:Questionable register does not cause an entry to be made in the EVENt part

:STATus:QUEStionable:PTRansition <PTransition>

Sets the bits of the NTRansition part of the STATus:QUEStionable register. If a bit is set, a transition from 1 to 0 in the condition part causes an entry to be made in the EVENt part of the register.

Parameters:

<PTransition> string

Example: `:STAT:OPER:PTR 32767`
all transitions from 0 to 1 in the condition part of the Status:Questionable register cause an entry to be made in the EVENt part

:STATus:QUEue[:NEXT]?

Queries the oldest entry in the error queue and then deletes it. Positive error numbers denote device-specific errors, and negative error numbers denote error messages defined by SCPI. If the error queue is empty, 0 ("No error") is returned.

The command is identical to `:SYSTem:ERRor[:NEXT]?` on page 384.

Return values:

<Next> string

Example: `:STATus:QUEue?`
queries the oldest entry in the error queue.
`Response: 0, 'no error'`
no errors have occurred since the error queue was last read out

Usage: Query only

12.14 SYSTem Subsystem

The `SYSTem` subsystem contains a series of commands for general functions which do not directly affect signal generation.

:SYSTem:ERRor:ALL?

Queries the error/event queue for all unread items and removes them from the queue. The response is a comma separated list of error number and a short description of the error in FIFO order.

Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.

Return values:
<All> string
List of: Error/event_number,"Error/event_description>[;Device-dependent info]"
If the queue is empty, the response is `0,"No error"`

Example: `SYST:ERR:ALL?`
queries all entries in the error queue.
`Response: 0, 'no error'`
No errors have occurred since the error queue was last read out.

Usage: Query only

Manual operation: See "Clear History" on page 226

:SYSTem:ERRor:CODE:ALL?

Queries all entries in the error queue and then deletes them. Only the error numbers are returned and not the entire error text.

Return values:

<All> string

0
"No error", i.e. the error queue is empty

positive value
Positive error numbers denote device-specific errors

negative value
Negative error numbers denote error messages defined by SCPI.

Example: `SYST:ERR:CODE:ALL`
queries all entries in the error queue.
`Response: 0`
no errors have occurred since the error queue was last read out.

Usage: Query only

:SYSTem:ERRor:CODE[:NEXT]?

Queries the oldest entry in the error queue and then deletes it. Only the error number is returned and not the entire error text.

Return values:

<Next> string

0
"No error", i.e. the error queue is empty

positive value
Positive error numbers denote device-specific errors

negative value
Negative error numbers denote error messages defined by SCPI.

Example: `SYST:ERR:CODE`
queries the oldest entry in the error queue.
`Response: 0`
No errors have occurred since the error queue was last read out.

Usage: Query only

:SYSTem:ERRor:COUNt?

Queries the number of entries in the error queue. If the error queue is empty, '0' is returned.

Return values:

<Count> string

Example: `SYST:ERR:COUN`
queries the number of entries in the error queue.
`Response: 1`
One error has occurred since the error queue was last read out.

Usage: Query only

:SYSTem:ERRor[:NEXT]?

Queries the error/event queue for the oldest item and removes it from the queue. The response 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.

Return values:

<Next> string

Error/event_number,"Error/event_description>[;Device-dependent info]"
If the queue is empty, the response is `0,"No error"`

Example: `SYST:ERR?`
queries the oldest entry in the error queue.
`Response: 0, 'no error'`
No errors have occurred since the error queue was last read out.

Usage: Query only

:SYSTem:SERRor?

This command returns a list of all errors existing at the time when the query is started. This list corresponds to the display on the info page under manual control.

Return values:

<StaticErrors> string

Example: `SYSTem:SERRor?`
queries all errors existing in the error queue.
`Response: -221, 'Settings conflict', 153, 'Input voltage out of range'`
the two returned errors have occurred since the error queue was last queried.

Usage: Query only

:SYSTem:VERSion?

Queries the SCPI version the instrument's command set complies with.

Return values:

<Version> string

Example: `SYST:VERS`
queries the SCPI version.
Response: `"1996"`
The instrument complies with the SCPI version from 1996.

Usage: Query only

:SYSTem:WAIT <TimeMs>

Delays the execution of the subsequent remote command by the specified time.

This function is useful, for example to execute an SCPI sequence automatically but with a defined time delay between some commands.

Setting parameters:

<TimeMs> integer

Wait time in ms

Range: 0 to 10000
*RST: 0

Example:

```
:SYSTem:WAIT 10000
// waits 10s before resetting the instrument
*RST
```

Usage: Setting only

12.15 List of Commands

13 Troubleshooting and Error Messages

R&S WinIQSIM2 distinguishes between a variety of different messages such as status messages, error messages, warnings, or information that are displayed in the "Info" line on the screen, and also entered in the error/event queue of the status reporting system.

This section describes the types of error messages and warnings. The information and status messages concerning the operating status of the application, see Chapter A.3.1, "Status Information Displayed in the Info Line", on page 421. The status reporting system is described in detail in Chapter A.1.1.5, "Status Reporting System", on page 410.

You can also access an "Info" window with detailed information on all messages in a history list. For details, see Chapter 10.2, "Querying Error Messages & Info Key", on page 224

13.1 Error Messages

Error messages indicate inconsistencies of a configuration or process. They are displayed in different colors depending on their importance and display duration. Errors (e.g. Settings conflicts) are displayed in red, information (e.g. file not found) and warnings in black. Warnings indicate less significant errors (e.g. if the settings exceed the permissible range).

Some error messages require that the error must be corrected before further operation can be performed properly. To access the "Info" dialog with a list of current messages and a detailed description of each message, select the "Info" button.

In the remote control mode, error messages are entered in the error/event queue of the status reporting system and can be queried with the command `SYSTem:ERRor?`. If the error queue is empty, 0 ("No error") is returned.

13.1.1 Volatile Messages

Volatile messages report automatic settings (e.g. switching off of incompatible types of modulation) or on illegal entries that are not accepted by the application (e.g. range violations). They are displayed in the info line on a yellow background. They are displayed on top of status information or permanent messages.

Volatile messages do not normally demand user actions and disappear automatically after a brief period of time. They are stored in the history, however.

SCPI command: `:SYSTem:ERRor:ALL?` or `:SYSTem:ERRor:CODE[:NEXT]?`

13.1.2 Permanent Messages

Permanent messages are displayed if an error occurs that impairs further operation. The error signaled by a permanent message must be eliminated before correct operation can be continued.

The message is displayed until the error is eliminated. It covers the status display in the info line. After error elimination, the message automatically disappears and is also recorded in the history.

SCPI command: `:SYSTem:SERRor?`

13.2 SCPI-Error Messages

The SCPI error messages are the same in all SCPI applications. Detailed information and an overview of all error messages as defined in SCPI standard can be found in the corresponding documentation.

The errors are assigned negative numbers. The error text being entered into the error/ event queue or being displayed is printed in bold face on the left together with the error code. Below the error text, there is an explanation as to the respective error.

13.3 Device-Specific Error Messages

The following table contains all error messages specific for the instrument in alphabetical order, as well as an explanation of the error situation. The positive error codes mark the errors specific to the instrument.

The device-specific error messages set bit 3 in the ESR register.

The index provides a list of the error messages sorted according to their error codes.

Error Code	Error	Description	Remedy
140	This modulation forces other modulations off	A modulation has been switched on which cannot be used at the same time as an already active modulation. The previous modulation has been switched off. Example: Enabling modulation GSM/ EDGE switches any active digital modulation off.	
241	No current list	There is no list selected. To execute the required operation, a list has to be selected in the related menu. If no list is available, a new list must be created.	
242	Unknown list type specified	The list type selected is not valid for the required operation. For instance, the file extension for waveform list files is *.wv. It is not possible to enter another file extension when selecting a list.	Check the selected list type.
261	Waveform Protected	The selected waveform file cannot be transferred to a controller. The waveform is produced with simulation software R&S WinIQSIM2 and is protected.	
460	Cannot open file	The selected file cannot be opened.	Check the path and file name.
461	Cannot write file	The file cannot be written.	Check if the file is read-only.
462	Cannot read file	The file cannot be read.	Check if the file contents are compatible with the file type.
463	Filename missing	The required operation cannot be executed because the file name is not specified.	Enter a file name when creating a new list.
464	Invalid filename extension	The file extension is not valid for the required operation.	Check the file extension. For instance, the file extension for waveform list files is `*.wv`. It is not possible to enter another file extension when storing a list.
465	File contains invalid data	The selected file contains data that is not valid for the file type. The file extension determines the data that is valid for this file type. If the file extension is changed, the lists are no longer recognized and the data is therefore invalid. Example: the extension of a waveform file (= `*.wv`) was changed to `*.txt`	Check the file extension.

13.4 Resolving Network Connection Failures

Several issues can cause failures in the network connection. This section lists the most likely reasons and the recommended solutions.

Common reasons for network connection failures

- Network connecting cables and cable connectors of poor quality
- Incompatibility between the network interface of the R&S WinIQSIM2 PC and certain switches or routers available on the market
- Invalidity of the assigned IP address

Possible solutions

NOTICE

Risk of connection errors

Before configuring the network, changing IP addresses or exchanging hardware, consult your network administrator.

Connection errors can affect the entire network.

- Check the network infrastructure. Exchange connecting cables if obvious damage is visible.
- Observe the link status LED next to the LAN connector.
  If a link failure is detected, connect the computer to a different network port or to a different network device.
- Check whether the IP address of the computer is within the network's address range.
  IP addresses that are set manually can be invalid.

13.5 Obtaining Technical Support

If problems occur, R&S WinIQSIM2 generates error messages which usually are sufficient for you to detect the cause of an error and find a remedy. Error message types are described in Chapter 13.1, "Error Messages", on page 392.

In addition, our customer support centers are there to assist you in solving any problems that you may encounter with your R&S WinIQSIM2. We will find solutions more quickly and efficiently if you provide us with the following information:

- **Software options** accessed via "File > Setup > Software/Option " provide information on the status of R&S WinIQSIM2 software components installed on your computer.
- **System messages:** displayed in the "Info" line provide information on any errors that have occurred.

Collect the error information and contact your Rohde & Schwarz Customer Support Center for technical support, see http://www.customersupport@rohde-schwarz.com.

Annex

A Annex

A.1 Reference Information on Remote Control

A.1.1 Additional Basics on Remote Control

This section provides basic information using the remote control.

A.1.1.1 Messages

The messages transferred on the data lines are divided into the following categories:

- Interface messages
  Interface messages are transmitted to the instrument on the data lines, with the attention line being active (LOW). They are used to communicate between the controller and the instrument. Interface messages can only be sent by instruments that have GPIB bus functionality. For details see the sections for the required interface.
- Instrument messages
  Instrument messages are employed in the same way for all interfaces, if not indicated otherwise in the description. Structure and syntax of the instrument messages are described in Chapter A.1.1.3, "SCPI Command Structure", on page 398. A detailed description of all messages available for the instrument is provided in the chapter "Remote Control Commands".
  There are different types of instrument messages, depending on the direction they are sent:
  – Commands
  – Instrument responses

Commands

Commands (program messages) are messages the controller sends to the instrument. They operate the instrument functions and request information. The commands are subdivided according to two criteria:

- According to the effect they have on the instrument:
  – **Setting commands** cause instrument settings such as a reset of the instrument or setting the frequency.
  – **Queries** cause data to be provided for remote control, e.g. for identification of the instrument or polling a parameter value. Queries are formed by directly appending a question mark to the command header.
- According to their definition in standards:

– **Common commands**: their function and syntax are precisely defined in standard IEEE 488.2. They are employed identically on all instruments (if implemented). They refer to functions such as management of the standardized status registers, reset and self-test.
– **Instrument control commands** refer to functions depending on the features of the instrument such as frequency settings. Many of these commands have also been standardized by the SCPI committee. These commands are marked as "SCPI confirmed" in the command reference chapters. Commands without this SCPI label are device-specific; however, their syntax follows SCPI rules as permitted by the standard.

Instrument responses

Instrument responses (response messages and service requests) are messages the instrument sends to the controller after a query. They can contain measurement results, instrument settings and information on the instrument status.

A.1.1.2 LAN Interface Messages

In the LAN connection, the interface messages are called low–level control messages. These messages can be used to emulate interface messages of the GPIB bus.

Command	Long term	Effect on the instrument
&ABO	Abort	Aborts processing of the commands just received.
&DCL	Device Clear	Aborts processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
&GTL	Go to Local	Transition to the "local" state (manual control). (The instrument automatically returns to remote state when a remote command is sent UNLESS `&NREN` was sent before.)
&GTR	Go to Remote	Enables automatic transition from local state to remote state by a subsequent remote command (after `&NREN` was sent).
&GET	Group Execute Trigger	Triggers a previously active instrument function (e.g. a sweep). The effect of the command is the same as with that of a pulse at the external trigger signal input.
&LLO	Local Lockout	Disables transition from remote control to manual control by means of the front panel keys.
&NREN	Not Remote Enable	Disables automatic transition from local state to remote state by a subsequent remote command. (To re-activate automatic transition use `&GTR`.)
&POL	Serial Poll	Starts a serial poll.

A.1.1.3 SCPI Command Structure

SCPI commands consist of a so-called header and, in most cases, 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 may consist of several mnemonics (keywords). Queries are formed by appending a question mark directly to the header.

The commands can be either device-specific or device-independent (common commands). Common and device-specific commands differ in their syntax.

Syntax for Common Commands

Common (=device-independent) commands consist of a header preceded by an asterisk (*) and possibly one or more parameters.

Examples:

*RST	RESET	Resets the instrument.
*ESE	EVENT STATUS ENABLE	Sets the bits of the event status enable registers.
*ESR?	EVENT STATUS QUERY	Queries the contents of the event status register.
*IDN?	IDENTIFICATION QUERY	Queries the instrument identification string.

Syntax for Device-Specific Commands

Not all commands used in the following examples are necessarily implemented in the instrument.

For demonstration purposes only, assume the existence of the following commands for this section:

- DISPlay[:WINDow<1...4>]:MAXimize <Boolean>
- FORMat:READings:DATA <type>[,<length>]
- HCOPy:DEVice:COLor <Boolean>
- HCOPy:DEVice:CMAP:COLor:RGB <red>,<green>,<blue>
- HCOPy[:IMMediate]
- HCOPy:ITEM:ALL
- HCOPy:ITEM:LABel <string>
- HCOPy:PAGE:DIMensions:QUADrant[<N>]
- HCOPy:PAGE:ORIentation LANDscape | PORTrait
- HCOPy:PAGE:SCALe <numeric value>
- MMEMory:COPY <file_source>,<file_destination>
- SENSE:BANDwidth|BWIDth[:RESolution] <numeric_value>
- SENSe:FREQuency:STOP <numeric value>
- SENSe:LIST:FREQuency <numeric_value>{,<numeric_value>}

Long and short form

The mnemonics feature a long form and a short form. The short form is marked by upper case letters, the long form corresponds to the complete word. Either the short form or the long form can be entered; other abbreviations are not permitted.

Example:

`HCOPy:DEVice:COLor ON` is equivalent to `HCOP:DEV:COL ON`.

Case-insensitivity

Upper case and lower case notation only serves to distinguish the two forms in the manual, the instrument itself is case-insensitive.

Numeric suffixes

If a command can be applied to multiple instances of an object, e.g. specific channels or sources, the required instances can be specified by a suffix added to the command. Numeric suffixes are indicated by angular brackets (<1...4>, <n>, <i>) and are replaced by a single value in the command. Entries without a suffix are interpreted as having the suffix 1.

Example:

Definition: `HCOPy:PAGE:DIMensions:QUADrant[<N>]`

Command: `HCOP:PAGE:DIM:QUAD2`

This command refers to the quadrant 2.

Different numbering in remote control

For remote control, the suffix may differ from the number of the corresponding selection used in manual operation. SCPI prescribes that suffix counting starts with 1. Suffix 1 is the default state and used when no specific suffix is specified.

Some standards define a fixed numbering, starting with 0. If the numbering differs in manual operation and remote control, it is indicated for the corresponding command.

Optional mnemonics

Some command systems permit certain mnemonics to be inserted into the header or omitted. These mnemonics are marked by square brackets in the description. The instrument must recognize the long command to comply with the SCPI standard. Some commands are considerably shortened by these optional mnemonics.

Example:

Definition: `HCOPy[:IMMediate]`

Command: `HCOP:IMM` is equivalent to `HCOP`

Optional mnemonics with numeric suffixes

Do not omit an optional mnemonic if it includes a numeric suffix that is relevant for the effect of the command.

Example:

Definition:`DISPlay[:WINDow<1...4>]:MAXimize <Boolean>`

Command: `DISP:MAX ON` refers to window 1.

In order to refer to a window other than 1, you must include the optional `WINDow` parameter with the suffix for the required window.

`DISP:WIND2:MAX ON` refers to window 2.

Parameters

Parameters must be separated from the header by a "white space". If several parameters are specified in a command, they are separated by a comma (,). For a description of the parameter types, refer to "SCPI Parameters" on page 402.

Example:

Definition:`HCOPy:DEVice:CMAP:COLor:RGB <red>,<green>,<blue>`

Command:`HCOP:DEV:CMAP:COL:RGB 3,32,44`

Special characters

|	**Parameters** A vertical stroke in parameter definitions indicates alternative possibilities in the sense of "or". The effect of the command differs, depending on which parameter is used. **Example**: Definition:HCOPy:PAGE:ORIentation LANDscape | PORTrait Command HCOP:PAGE:ORI LAND specifies landscape orientation Command HCOP:PAGE:ORI PORT specifies portrait orientation **Mnemonics** A selection of mnemonics with an identical effect exists for several commands. These mnemonics are indicated in the same line; they are separated by a vertical stroke. Only one of these mnemonics needs to be included in the header of the command. The effect of the command is independent of which of the mnemonics is used. **Example**: DefinitionSENSE:BANDwidth|BWIDth[:RESolution] <numeric_value> The two following commands with identical meaning can be created: SENS:BAND:RES 1 SENS:BWID:RES 1
[ ]	Mnemonics in square brackets are optional and may be inserted into the header or omitted. **Example**: HCOPy[:IMMediate] HCOP:IMM is equivalent to HCOP
{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted. **Example**: SENSe:LIST:FREQuency <numeric_value>{,<numeric_value>} The following are valid commands: SENS:LIST:FREQ 10 SENS:LIST:FREQ 10,20 SENS:LIST:FREQ 10,20,30,40

SCPI Parameters

Many commands are supplemented by a parameter or a list of parameters. The parameters must be separated from the header by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). Allowed parameters are:

- Numeric values
- Special numeric values
- Boolean parameters
- Text
- Character strings
- Block data

The parameters required for each command and the allowed range 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 may comprise up to 255 characters, the exponent must lie inside the value range -32000 to 32000. 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. Allowed unit prefixes are G (giga), MA (mega), MOHM and MHZ are also allowed), K (kilo), M (milli), U (micro) and N (nano). If the unit is missing, the basic unit is used.

Example: `SENS:FREQ:STOP 1.5GHz` = `SENS:FREQ:STOP 1.5E9`

Units

For physical quantities, the unit can be entered. Allowed unit prefixes are:

- G (giga)
- MA (mega), MOHM, MHZ
- K (kilo)
- M (milli)
- U (micro)
- N (nano)

If the unit is missing, the basic unit is used.

Example:

`SENSe:FREQ:STOP 1.5GHz` = `SENSe:FREQ:STOP 1.5E9`

Some settings allow relative values to be stated in percent. According to SCPI, this unit is represented by the `PCT` string.

Example:

`HCOP:PAGE:SCAL 90PCT`

Special numeric values

The texts listed below are interpreted as special numeric values. In the case of a query, the numeric value is provided.

- **MIN/MAX**
  MINimum and MAXimum denote the minimum and maximum value.
- **DEF**
  DEFault denotes a preset value which has been stored in the EPROM. This value conforms to the default setting, as it is called by the *RST command.
- **UP/DOWN**
  UP, DOWN increases or reduces the numeric value by one step. The step width can be specified via an allocated step command for each parameter which can be set via UP, DOWN.
- **INF/NINF**

INFinity, Negative INFinity (NINF) represent the numeric values 9.9E37 or -9.9E37, respectively. INF and NINF are only sent as instrument responses.

- **NAN**
  Not A Number (NAN) represents the value 9.91E37. NAN is only sent as a instrument response. This value is not defined. Possible causes are the division of zero by zero, the subtraction of infinite from infinite and the representation of missing values.

Example:

Setting command: `SENSe:LIST:FREQ MAXimum`

Query: `SENS:LIST:FREQ?, Response: 3.5E9`

Queries for special numeric values

The numeric values associated to `MAXimum/MINimum/DEFault` can be queried by adding the corresponding mnemonics to the command. They must be entered following the quotation mark.

Example: `SENSe:LIST:FREQ? MAXimum`

Returns the maximum numeric value as a result.

Boolean Parameters

Boolean parameters represent two states. The "ON" state (logically true) is represented by "ON" or a numeric value 1. The "OFF" state (logically untrue) is represented by "OFF" or the numeric value 0. The numeric values are provided as the response for a query.

Example:

Setting command: `HCOPy:DEV:COL ON`

Query: `HCOPy:DEV:COL?`

Response: `1`

Text parameters

Text parameters observe the syntactic rules for mnemonics, 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: `HCOPy:PAGE:ORIentation LANDscape`

Query: `HCOP:PAGE:ORI?`

Response: `LAND`

Character strings

Strings must always be entered in quotation marks (' or ").

Example:

`HCOP:ITEM:LABel "Test1"` or `HCOP:ITEM:LABel 'Test1'`

Block data

Block data is a format which is suitable for the transmission of large amounts of data. A command using a block data parameter has the following structure:

Example:

`FORMat:READings:DATA #45168xxxxxxxx`

The ASCII character # introduces the data block. The next number indicates how many of the following digits describe the length of the data block. In the example the 4 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.

#0 specifies a data block of indefinite length. The use of the indefinite format requires a `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

The following table provides an overview of the syntax elements:

:	The colon separates the mnemonics 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 (both single and double quotation marks are possible).
#	The hash symbol introduces binary, octal, hexadecimal and block data. • Binary: #B10110 • Octal: #O7612 • Hexa: #HF3A7 • Block: #21312
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

Structure of a command line

A command line may consist of one or several commands. It is terminated by one of the following:

- a <New Line>
- a <New Line> with EOI

- 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:

`MMEM:COPY "Test1","MeasurementXY";:HCOP:ITEM ALL`

This command line contains two commands. The first command belongs to the MMEM system, the second command belongs to the HCOP 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:

`HCOP:ITEM ALL;:HCOP:IMM`

This command line contains two commands. Both commands are part of the `HCOP` command system, i.e. they have one level in common.

When abbreviating the command line, the second command begins with the level below HCOP. The colon after the semicolon is omitted. The abbreviated form of the command line reads as follows:

`HCOP:ITEM ALL;IMM`

A new command line always begins with the complete path.

Example:

`HCOP:ITEM ALL`

`HCOP:IMM`

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. According to SCPI, the responses to queries are partly subject to stricter rules than in standard IEEE 488.2.

- The requested parameter is transmitted without a header.
  Example: `HCOP:PAGE:ORI?`, Response: `LAND`
- Maximum values, minimum values and all other quantities that are requested via a special text parameter are returned as numeric values.
  Example: `SENSe:FREQuency:STOP? MAX`, Response: `3.5E9`
- Numeric values are output without a unit. Physical quantities are referred to the basic units or to the units set using the `Unit` command. The response `3.5E9` in the previous example stands for 3.5 GHz.
- Truth values (Boolean values) are returned as `0` (for OFF) and `1` (for ON).
  Example:

Setting command: `HCOPy:DEV:COL ON`
Query: `HCOPy:DEV:COL?`
Response: `1`

- Text (character data) is returned in a short form.
Example:
Setting command: `HCOPy:PAGE:ORIentation LANDscape`
Query: `HCOP:PAGE:ORI?`
Response: `LAND`

A.1.1.4 Command Sequence and Synchronization

IEEE 488.2 defines a distinction between overlapped and sequential commands:

- A sequential command always completes executing before the next command starts. Commands that are processed quickly are defined as sequential commands. They are not implemented in the instrument. However, the execution time of most of the commands is so short that they act as sequential commands, if they are sent in separate command lines.
- An overlapping command is still running when the next command starts. Usually, an overlapping command takes a certain time to process its task, and thus allows the program to execute other tasks, while it is still running. If overlapping commands have to follow a specific order, for example to avoid incorrect measurement readings, they must be executed in sequence. This is called synchronization between the controller and the instrument.

Several setting commands within a command line are not necessarily processed in the order they are received. Even if they are implemented as sequential commands. To follow a particular sequence, send each command in a separate line.

As a rule, send commands and queries in different program messages.

Preventing Overlapping Execution

To prevent an overlapping execution of commands, one of the commands *OPC, *OPC? or *WAI can be used. All three commands cause a certain action only to be carried out after the hardware has been set. By suitable programming, the controller can be forced to wait for the corresponding action to occur.

*Table A-1: Synchronization using *OPC, *OPC? and *WAI*

Command	Action	Programming the controller
*OPC	Sets the Operation Complete bit in the ESR after all previous commands have been executed.	• Setting bit 0 in the ESE • Setting bit 5 in the SRE • Waiting for service request (SRQ)
*OPC?	Stops command processing until 1 is returned. This is only the case after the Operation Complete bit has been set in the ESR. This bit indicates that the previous setting has been completed.	Sending *OPC? directly after the command whose processing should be terminated before other commands can be executed.
*WAI	Stops further command processing until all commands sent before *WAI have been executed.	Sending *WAI directly after the command whose processing should be terminated before other commands are executed.

Command synchronization using *WAI or *OPC? is a good choice if the overlapped command takes only little time to process. The two synchronization commands simply block overlapped execution of the command. Append the synchronization command to the overlapping command, for example:

```
SINGle; *OPC?
```

For time consuming overlapped commands you can allow the controller or the instrument to do other useful work while waiting for command execution. Use one of the following methods:

*OPC with a service request

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1
2. Set bit no. 5 in the SRE: *SRE 32 to enable ESB service request.
3. Send the overlapped command with *OPC
4. Wait for a service request

   The service request indicates that the overlapped command has finished.

*OPC? with a service request

1. Set bit no. 4 in the SRE: *SRE 16 to enable MAV service request.
2. Send the overlapped command with *OPC?
3. Wait for a service request

   The service request indicates that the overlapped command has finished.

Event Status Register (ESE)

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1
2. Send the overlapped command without *OPC, *OPC? or *WAI

3. Poll the operation complete state periodically (by means of a timer) using the sequence: *OPC; *ESR?

   A return value (LSB) of 1 indicates that the overlapped command has finished.

Examples to Command Sequence and Synchronization

See the following examples to command sequences and synchronization. Some examples given illustrate possible constellations for overlapping tasks.

Example: Commands and queries in one message

The response to a query combined in a program message with commands that affect the queried value is not predictable.

The following commands always return the specified result:

`:FREQ:STAR 1GHZ;SPAN 100 :FREQ:STAR?`

Result:

`1000000000` (1 GHz)

Whereas the result for the following commands is not specified by SCPI:

`:FREQ:STAR 1GHz;STAR?;SPAN 1000000`

The result could be the value of `STARt` before the command was sent since the instrument might defer executing the individual commands until a program message terminator is received. The result could also be 1 GHz if the instrument executes commands as they are received.

Example: Overlapping command with *OPC

The instrument implements `*RST` as an overlapped command. Assuming that `*RST` takes longer to execute than `*OPC`, sending the following command sequence results in initiating a reset and, after some time, setting the `OPC` bit in the `ESR`:

`*RST; *OPC`

Sending the following commands still initiates a reset:

`*RST; *OPC; *CLS`

However, since the operation is still pending when the instrument executes `*CLS`, forcing it into the "Operation Complete Command Idle" State (OCIS), `*OPC` is effectively skipped. The `OPC` bit is not set until the instrument executes another `*OPC` command.

Example: Overlapped command followed by non-conflicting commands

Suppose that the instrument is switched on to provide a real time test signal that requires some calculation time. At the same time, some settings for the configuration of a different signal are made which do not interact with the generated signal (for example the signal may be used later on). The signal generation and the signal configuration are independent from each other, so there is no need to synchronize the following overlapped commands:

`SOUR:BB:3GPP:STAT ON`

`SOUR:BB:GSM:FORM FSK2`

Example: Overlapped command followed by conflicting commands

Suppose that the generator is switched on to provide a real time test signal that requires some calculation time. This signal is to be added to a waveform from the second baseband generator. In this case, the application program has to make sure that the real signal is available in the added signal before further action is started. This involves an appropriate synchronization technique for the first command (the following sequence assumes an appropriate routing):

`SOUR:BB:3GPP:STAT ON`

The instrument waits until command has finished.

`SOUR2:BB:GSM:STAT ON`

Depending on the selected synchronization techniques, non-conflicting commands can be executed while waiting until the synchronized overlapped command has finished.

Example: Polling the progress of the zeroing process

Suppose that you start the zeroing for a connected power sensor via the remote control command `SENS1:ZERO`. This process blocks the processing of further tasks during execution. The query for completeness is performed with the `*OPC?` command. It returns a `1` in the output buffer when the process is completed.

`SENS:ZERO;*OPC?`

Instead of waiting via `*OPC?`, you can perform alternative tasks while the zeroing is running, as for example updating the GUI or adjusting other instruments. Synchronize the commands by querying the progress of the zeroing process periodically via the event status register `*ESR?`:

`*SRE 32`

Sets the service request enable. The bit is set when an event in the event status register occurs.

`*ESE 1`

Configures the mask of the event status register to "Operation Complete".

`SENS:ZERO;*OPC`

Sets the evaluation via the status byte query. It uses `*OPC?` as the reference.

`*CLS`

Clears all status registers.

Even if the instrument is busy, you can perform this procedure, since the query is executed in a subchannel.

A.1.1.5 Status Reporting System

The status reporting system stores all information on the current operating state of the instrument, and on errors which have occurred. This information is stored in the status registers and in the error queue.

You can query both with the commands of the STATus Subsystem.

Hierarchy of the Status Registers

The Figure A-1 shows the hierarchical structure of information in the status registers (ascending from left to right).

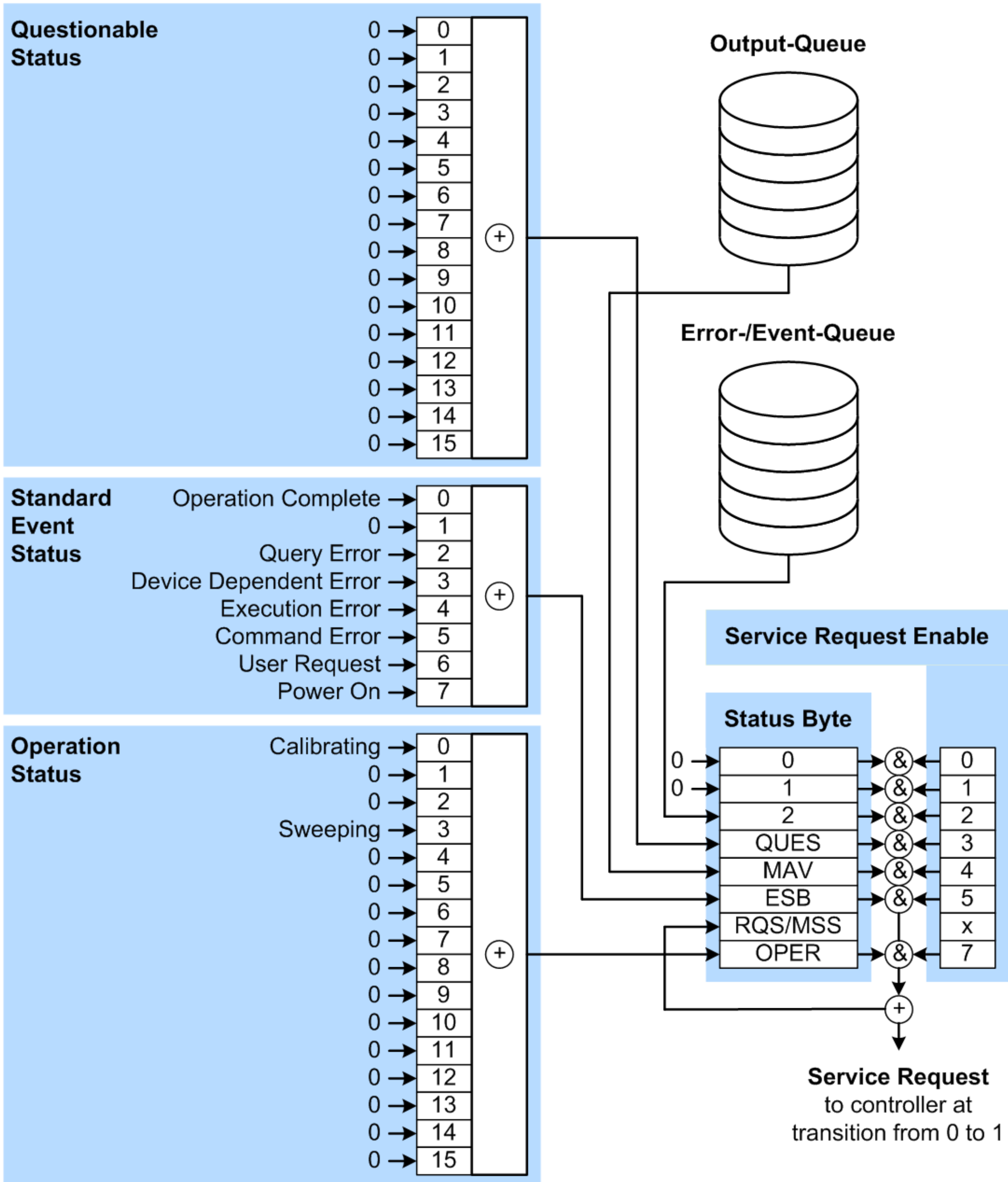


Figure A-1: Graphical overview of the status registers hierarchy

OPER = Operation Status Summary Bit
RQS/MSS = Service Request Generation
ESB = Standard Event Status Summary Bit
MAV = Message Available in Output Queue
QUES = Questionable Status Summary Bit
2 = Error- /Event-Queue
1, 0 = not used

Note: This legend explains the abbreviations to the Status Byte Register.

The R&S WinIQSIM2 uses the following status registers:

- **Status Byte** (STB) and **Service Request Enable** (SRE), see "Status Byte (STB) and Service Request Enable Register (SRE)" on page 414.
- **Standard Event Status**, i.e. the Event status Register (ESR) and the Event Status Enable (ESE), see "Event Status Register (ESR) and Event Status Enable Register (ESE)" on page 415.
- **Questionable Status** and **Operation Status**, the (SCPI status registers, see "Structure of a SCPI Status Register" on page 412, "Questionable Status Register (STATus:QUEStionable)" on page 416 and "Operation Status Register (STATus:OPERation)" on page 416.
- **Output-Queue**
  The output queue 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 and thus is represented in the overview.
- **Error- /Event-Queue**
  The error-/event-queue contains all errors and events that have occurred in the past. When reading the queue, the instrument starts with the first occurred error/event.

All status registers have the same internal structure.

SRE, ESE

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.

Structure of a SCPI Status Register

Each standard SCPI register consists of 5 parts. Each part has a width of 16 bits and has 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 integers.

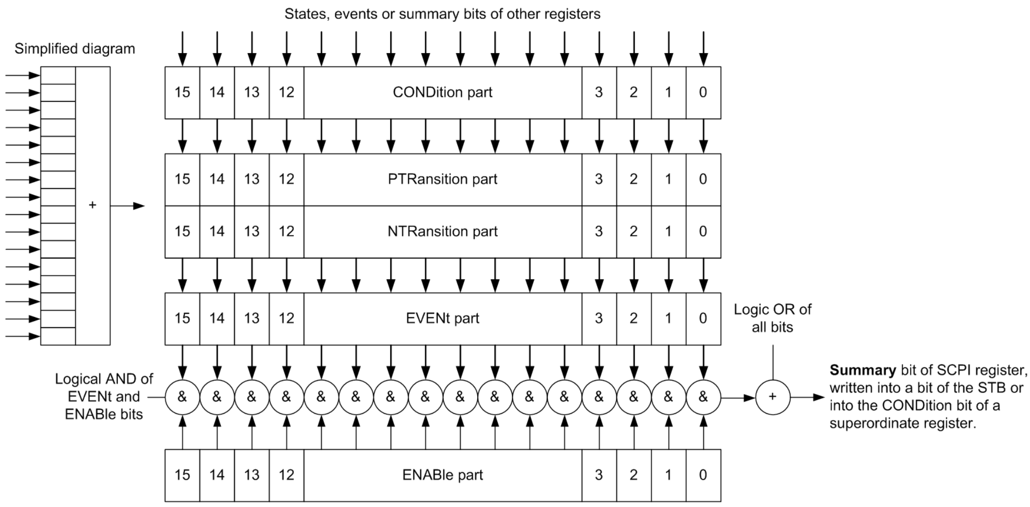


Figure A-2: The status-register model

Description of the five status register parts

The five parts of a SCPI register have different properties and functions:

- **CONDition**
  The `CONDition` part is written into directly by the hardware or the sum bit of the next lower register. Its contents reflect the current instrument status. This register part can only be read, but not written into or cleared. Its contents are not affected by reading.
- **PTRansition / NTRansition**
  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 part can be written into and read as required. Its contents are not affected by reading.
  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 written into and read as required. Its contents are not affected by reading.
- **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. This part can only be

read by the user. 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 (see 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 '+').
  `ENABle` bit = 0: the associated `EVENt` bit does not contribute to the sum bit
  `ENABle` bit = 1: if the associated `EVENt` bit is "1", the sum bit is set to "1" as well.
  This part can be written into and read by the user as required. Its contents are not affected by reading.

Sum bit

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.

Status Byte (STB) and Service Request Enable Register (SRE)

The `STatus Byte` (STB) is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.

The STB is read using the command `*STB?` or a serial poll.

The `STatus Byte` (STB) is linked to the `Service Request Enable` (SRE) register. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. 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. The SRE can be set using the command `*SRE` and read using the command `*SRE?`.

Table A-2: Meaning of the bits used in the status byte

Bit No.	Meaning
0...1	Not used
2	Error Queue not empty The bit is set when an entry is made in the error queue. If this bit is enabled by the SRE, each entry of the error queue generates a service request. Thus an error can be recognized and specified in greater detail by polling the error queue. The poll provides an informative error message. This procedure is to be recommended since it considerably reduces the problems involved with remote control.
3	QUEStionable status register summary bit The bit is set if an `EVENt` bit is set in the `QUEStionable` status register and the associated `ENABle` bit is set to 1. A set bit indicates a questionable instrument status, which can be specified in greater detail by querying the `STATus:QUEStionable` status register.
4	MAV bit (message available) The bit is set if a message is available in the output queue which can be read. This bit can be used to enable data to be automatically read from the instrument to the controller.

Bit No.	Meaning
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 indicates a serious error which can be specified in greater detail by polling the event status register.
6	MSS bit (master status summary bit) The bit is set if the instrument triggers a service request. This is the case if one of the other bits of this registers is set together with its mask bit in the service request enable register SRE.
7	`STATus:OPERation` status register summary bit The bit is set if an `EVENt` bit is set in the `OPERation` status register and the associated `ENABle` bit is set to 1. A set bit indicates that the instrument is just performing an action. The type of action can be determined by querying the `STATus:OPERation` status register.

Event Status Register (ESR) and Event Status Enable Register (ESE)

The ESR is defined in IEEE 488.2. It can be compared with the `EVENt` part of a SCPI register. The event status register can be read out using command *ESR?.

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 STB is set. The ESE register can be set using the command `*ESE` and read using the command `*ESE?`.

Table A-3: Meaning of the bits used in the event status register

Bit No.	Meaning
0	Operation Complete This bit is set on receipt of the command *OPC exactly when all previous commands have been executed.
1	Not used
2	Query Error This bit is set if either the controller wants to read data from the instrument without having sent a query, or if it 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 denotes 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 denotes the error in greater detail, is entered into the error queue.
5	Command Error This bit is set if a command is received, which is undefined or syntactically incorrect. An error message with a number between -100 and -200, which denotes the error in greater detail, is entered into the error queue.

Bit No.	Meaning
6	User Request This bit is set when the instrument is switched over to manual control.
7	Power On (supply voltage on) This bit is set on switching on the instrument.

Questionable Status Register (STATus:QUEStionable)

This register contains information on questionable instrument states. Such states may occur when the instrument is not operated in compliance with its specifications.

To read the register, use the query commands STAT:QUEST:COND? or STAT:QUEST[:EVEN]?.

Table A-4: Meaning of the bits used in the questionable status register

Bit No.	Meaning
0–15	Not used

Operation Status Register (STATus:OPERation)

This condition part contains information on the actions currently being performed by the instrument, while the event part contains information on the actions performed by the instrument since the last readout of the register.

To read the register, use the query commands STAT:OPER:COND? or STAT:OPER[:EVEN]?.

Table A-5: Meaning of the bits used in the operation status register

Bit No.	Meaning
0	Calibrating The bit is set during the calibration phase.
1–2	Not used
3	Sweeping This bit is set during a sweep in automatic or single mode.
4–15	Not used

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. To do this and react appropriately, the controller must receive and evaluate the information of all devices. The following standard methods are used:

- **Service request** (SRQ) initiated by the instrument
- **Serial poll** of all devices in the bus system, initiated by the controller to find out who sent an SRQ and why
- Query of a **specific instrument status** by commands
- Query of the **error queue**

Service Request

Under certain circumstances, the instrument can send a service request (SRQ) to the controller. Usually this service request initiates an interrupt at the controller, to which the control program can react appropriately. An SRQ is always initiated if one or several of bits 2, 4 or 5 of the status byte are set and enabled in the SRE. Each of these bits combines the information of the error queue or the output buffer. To use the possibilities of the service request effectively, all bits should be set to "1" in the enable registers SRE and ESE.

Example:

Use command `*OPC` to generate an SRQ .

`*ESE 1` - set bit 0 of ESE (Operation Complete)

`*SRE 32` - set bit 5 of SRE (ESB).

After its settings have been completed, the instrument generates an SRQ.

The SRQ is the only possibility for the instrument to become active on its own. Each controller program should set the instrument such that a service request is initiated in the case of malfunction. The program should react appropriately to the service request.

Serial Poll

In a serial poll, just as with command `*STB`, the status byte of an instrument is queried. However, the query is realized via interface messages and is thus clearly faster.

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 mainly used to obtain a fast overview of the state of several instruments connected to the controller.

Query of an instrument status

Each part of any status register can be read using queries. There are two types of commands:

- The common commands `*ESR?`, `*IDN?`, `*IST?`, `*STB?` query the higher-level registers.
- The commands of the `STATus` system query the SCPI registers (`STATus:QUEStionable`...)

The returned value is always a decimal number that represents the bit pattern of the queried register. This number is evaluated by the controller program.

Queries are usually used after an SRQ in order to obtain more detailed information on the cause of the SRQ.

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 looked up in the Error Log or queried via remote control using `SYSTem:ERRor[:NEXT]?`. Each call of

SYSTem:ERRor[:NEXT]? provides one entry from the error queue. If no error messages are stored there any more, the instrument responds with 0, "No error".

The error queue should be queried after every SRQ in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

Reset Values of the Status Reporting System

The following table contains the different commands and events causing the status reporting system to be reset. None of the commands, except of *RST and SYSTem:PRESet affect the functional instrument settings. In particular, DCL does not change the instrument settings.

Table A-6: Resetting the status reporting system

Event	**Switching on supply voltage** **Power-On-Status-Clear**		DCL, SDC **(Device Clear, Selected Device Clear)**	*RST **or** SYSTem:PRESet	STATus:PRESet	*CLS
Effect	**0**	**1**				
Clear STB, ESR	-	Yes	-	-	-	Yes
Clear SRE, ESE	-	Yes	-	-	-	-
Clear PPE	-	Yes	-	-	-	-
Clear error queue	Yes	Yes	-	-	-	Yes
Clear output buffer	Yes	Yes	Yes	1)	1)	1)
Clear command processing and input buffer	Yes	Yes	Yes	-	-	-
1) The first command in a command line that immediately follows a <PROGRAM MESSAGE TERMINATOR> clears the output buffer.						

A.1.1.6 General Programming Recommendations

Initial instrument status before changing settings

Manual operation is designed for maximum possible operating convenience. In contrast, the priority of remote control is the "predictability" of the instrument status. Thus, when a command attempts to define incompatible settings, the command is ignored and the instrument status remains unchanged, i.e. other settings are not automatically adapted. Therefore, control programs should always define an initial instrument status (e.g. using the *RST command) and then implement the required settings.

Command sequence

As a general rule, send commands and queries in different program messages. Otherwise, the result of the query may vary depending on which operation is performed first (see also Preventing Overlapping Execution).

Reacting to malfunctions

The service request is the only possibility for the instrument to become active on its own. Each controller program should instruct the instrument to initiate a service request in case of malfunction. The program should react appropriately to the service request.

Error queues

The error queue should be queried after every service request in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

A.2 Extensions for User Files

The following table lists all available file extensions for user files.

Table A-7: List of the automatically assigned file extensions in the instrument

Function/Digital Standard	List type	Contents	File suffix
Instrument State	Settings	Instrument settings	`*.savrcl`
Import Settings	Settings	Settings	`*.import`
"Arbitrary Wave-form Generator"	Waveform	ARB waveforms ARB multi segment waveforms	`*.wv`
	Waveform	ARB multi carrier settings	`*.arb_multcarr`
	Configuration data	Configuration file for creation of multisegment ARB waveforms	`*.inf_mswv`
	Play List	ARB Sequencing List	`*.wvs`
"DM"	Data List	Digital modulation data	`*.dm_iqd` `*.tdm`
	Control List	Data to control digital modulation	`*.dm_iqc`
	Settings	Digital modulation settings	`*.dm`
	User Standard	Digital modulation user standard	`*.dm_stu`
	User Mapping	Digital modulation user mapping	`*.vam`
	User Filter	Digital modulation user filter	`*.vaf`
"GSM/EDGE"	Settings	GSM/EDGE settings	`*.gsm`

Function/Digital Standard	List type	Contents	File suffix
	Slot	User-defined slot data	`*.gsm_slu`
	Frame	User-defined frame data	`*.gsm_fu`
	Slot	Higher symbol rate slot	`*.gsm_hslu`
	Frame	Higher symbol rate frame	`*.gsm_hfu`
"Bluetooth"	Bluetooth Settings	Complete setting of the Bluetooth menu	`*.bto`
"TETRA"	TETRA Settings	Complete setting of the TETRA menu	`*.tetra`
"3GPP FDD"	Settings	Complete setting of the 3GPP (FDD) menu	`*.3g`
	Settings	Channel coding enhanced DPCH channels (downlink)	`*.3g_ccod_dl_s`
	Settings	Channel coding enhanced DPDCH channels (uplink)	`*.3g_ccod_ul`
"CDMA2000"	Settings	Complete setting of the CDMA2000 menu	`*.cdma2k`
"TD-SCDMA2000"	Settings	Complete setting of the TD-SCDMA2000 menu	`*.tdscdma`
	Settings	Test Model for TD-SCDMA2000	`*.tdtmd`
1xEV-DO	Settings	Complete setting of the 1xEV-DO menu	`*.evdo`
"IEEE 802.11 WLAN"	Settings	Complete setting of the IEEE 802.11 WLAN menu	`*.wlan`
"IEEE 802.11 n WLAN"	Settings	Complete setting of the IEEE 802.11n WLAN menu	`*.wlann`
	Settings	Beamforming data	`*.bmf`
"IEEE 802.16 WiMAX"	Settings	Complete setting of the IEEE 802.16 WiMAX menu	`*.wimax`
"EUTRA/LTE"	Settings	Complete setting of the EUTRA/LTE menu	`*.eutra`
	Settings	TDD Settings	`*.lte_tdd`
	Settings	User-defined IQ-File	`*.iqw`

"GNSS"	Settings	Complete setting of the GNSS menu	`*.gps`
			`*.galileo`
			`*.glonas`

	Settings	Almanac Settings	`*.txt`
			`*.alm`
			`*.al3`
			`*.agl`

	Settings	Waypoints File	`*.txt`
	Settings	Generated Almanac File	`*rs_al` `*.rs_yuma`

Function/Digital Standard	List type	Contents	File suffix
	Settings	Ionospheric File	`*.rs_ion`
	Settings	Navigation Data	`*.rs_nav`
	Settings	UTC File	`*.rs_utc` `*.rs_acq`
"DVB"	Settings	Complete setting of the DVB menu	`*.dvb`

	Settings	DVB Transport Stream	`*.gts`
			`*.ts`
			`*.trp`

"DAB/T-DMB"	Settings	Complete setting of the DAB/T-DMB menu	`*.dab`
	Settings	DAB ETI Files	`*.eti` `*.xeti`
NFC	Settings	Complete setting of the NFC menu	`*.nfc`

A.3 Elements and Controls of the Graphical User Interface

To get familiar with the display of the instrument, you will find a summary description in the guide Chapter 3.5.1, "Understanding the Display Information", on page 42.

These are in detail:

- "Menu Bar / Toolbar" on page 43
- "Status bar" on page 44
- "Info Line" on page 45
- "Block diagram" on page 45
- "Taskbar" on page 46
- "Additional display characteristics" on page 46

This overview provides additional details on some of the topics.

A.3.1 Status Information Displayed in the Info Line

The info line shows system messages as e.g. status information on the operating status of R&S WinIQSIM2, but also error messages and warnings. It appears when an event message is generated. Some of the messages are permanently displayed, and the less important only temporarily, as volatile information.

The following section contains the distinctive messages concerning the operating status of the application. For information on the error messages and warnings, refer to Chapter 13, "Troubleshooting and Error Messages", on page 392.

You can also access an info window with detailed information about all messages in a history list. For details, see Chapter 10.2, "Querying Error Messages & Info Key", on page 224.

The following types of messages are distinguished:

- **Error**
  An error indicates that a fault has occurred, e.g. in the current configuration or even in the application.

  Depending on the severity of the error, R&S WinIQSIM2 classifies an error as...
  - **Critical**
    Prevents R&S WinIQSIM2 from working, e.g. a DLL file version does not match. Critical errors are displayed in red.
  - **System**
    Concerns the operating system, such as incorrect file path. System errors are displayed in black.
- **Information**
  Displays system messages, e.g. information ton the operating status.
  Information messages are displayed in black.
- **Warning**
  A warning indicates a less significant error and is displayed in black.
- **Brief message**
  Brief messages report automatic settings in the program, e.g. switching off of incompatible types of modulation, or on illegal entries that are not accepted by the program, such as range violations. These messages are displayed in the info line on a yellow background. They are displayed on top of status information or permanent messages.
  Brief messages usually do not demand user actions and disappear automatically after a short period of time. They are stored in the history, however.
- **Permanent messages**
  Permanent messages are displayed if an error occurs that impairs further operation. The error signaled by a permanent message must be corrected before you can continue.
  The message is displayed until the error is eliminated. It covers the status display in the info line. If the error is corrected, the error message disappears, however, it is recorded in the history.

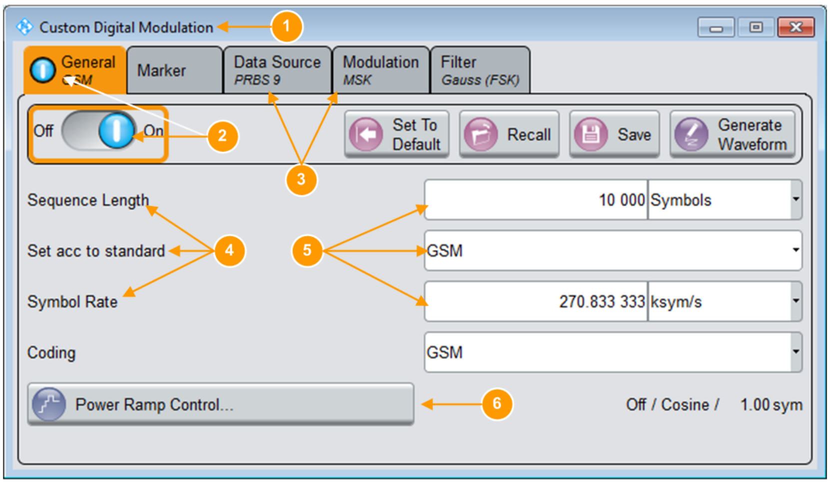
A.3.2 Elements Displayed for Interactions

The signal generator is primarily configured by settings dialogs, but also by active elements in the block diagram and the status bar. Additional interactive elements such as tooltips or on-screen keypads help you to conveniently perform the settings.

Dialogs
It is distinguished between settings dialogs and diagrams.

Basically, a dialog has a parent name corresponding to its content.

"Settings dialog"



1 = Dialog name
2 = Status indicators
3 = Key parameter settings
4 = Parameters
5 = Parameter entry field
6 = Access keys for further settings dialogs

The name(1) of a settings dialog is displayed in the header. It is a generic term that refers to the functionality covered in the dialog. Dialog names are for example a digital standard (EUTRA/LTE).

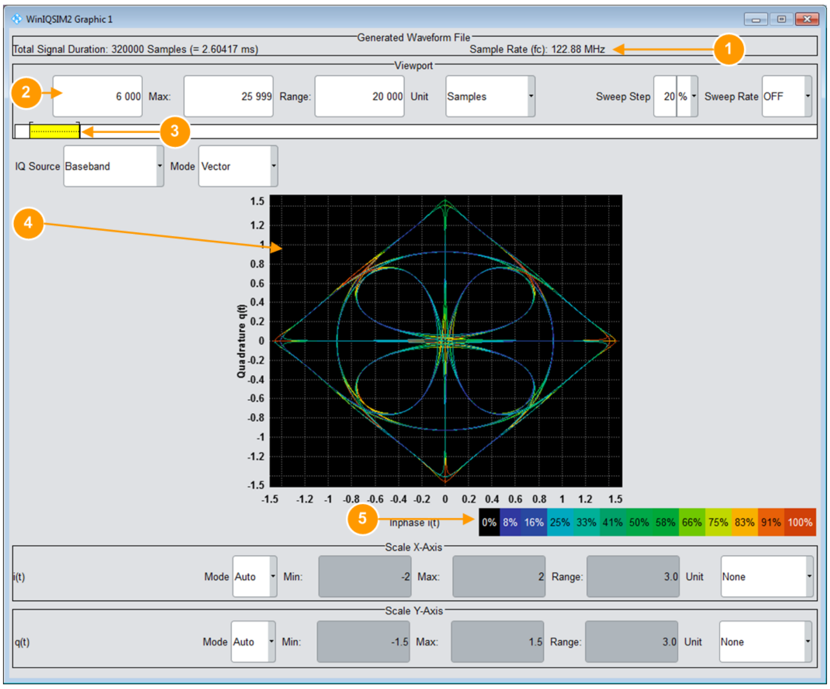
The header of a settings dialog contains the well known buttons to minimize or close the dialog, and a parent name which is usually composed of:

- Parent name
- Generic term of the function

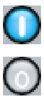
A settings dialog is mostly divided into tabs with logically grouped parameters. The tab name expresses the content and may contain status indicators or the set value of a key parameter.

"Diagram dialogs"

The graphical representation of a diagram dialog contains the graph and functional keys.



1 = Signal characteristics
2 = Setting parameters
3 = Viewport
4 = Graph
5 = Color Map

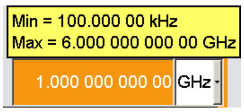


On/Off Switch

Indicates the current state of a function.

The icon is interactive and you can select it to turn on and off.

Tooltip



In edit mode, the tooltip indicates the possible value range of a parameter.

List of Commands

Index

Symbols

A

B

C

D