

R&S® Pulse Sequencer

Pulse Signal Generation Software

User Manual



1176.9512.02 – 04

This document describes the following software options:

- R&S®SMW-K300/-K301
1413.8805.02, 1413.9776.02
- R&S®SMW-K308
1414.1433.02
- R&S®SMBV-K300/K301
1419.2744.02, 1419.2789.02
- R&S®SGT-K300/K301
1419.7652.02, 1419.7700.02

This manual describes software version V1.2 and later of the R&S®Pulse Sequencer.

The software contained in this product makes use of several valuable open source software packages. For information, see the "Open Source Acknowledgement" document, which is available for download from the R&S Pulse Sequencer product page at e.g. <http://www.rohde-schwarz.com/product/SMW200A.html> > "Downloads > Software".

Rohde & Schwarz would like to thank the open source community for their valuable contribution to embedded computing.

© 2016 Rohde & Schwarz GmbH & Co. KG

Mühlendorfstr. 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

Subject to change – Data without tolerance limits is not binding.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.

Trade names are trademarks of the owners.

The following abbreviations are used throughout this manual: R&S®SMW200A is abbreviated as R&S SMW, R&S®SGT100A is abbreviated as R&S SGT, R&S®SMBV100A is abbreviated as R&S SMBV, R&S®Pulse Sequencer is abbreviated as R&S Pulse Sequencer, R&S®WinIQSIM2™ is abbreviated as R&S WinIQSIM2.

Contents

1	Preface	9
1.1	About this Manual	9
1.2	Documentation Overview	10
1.3	Conventions Used in the Documentation	11
1.3.1	Typographical Conventions.....	11
1.3.2	Conventions for Procedure Descriptions.....	11
1.3.3	Notes on Screenshots.....	11
2	Welcome to the R&S Pulse Sequencer	12
2.1	Accessing the Pulse Sequencer	13
3	Getting Started	14
3.1	Required Options	14
3.2	Installing the Software	15
3.3	Starting the R&S Pulse Sequencer for the First Time	17
3.4	Understanding the Displayed Information	18
3.5	Means of Users Interaction	22
3.6	Trying Out the Software	25
3.6.1	Completing the Scenario that was Automatically Created upon Start Up.....	26
3.6.2	Generating an ARB Waveform File.....	27
3.6.3	Verifying the Generated Signal in the Waveform View.....	28
3.6.4	Launching the Built-In Wizard.....	29
3.6.5	Using the Wizard to Create a Complex Scenario.....	30
3.6.6	Transferring the Waveform to and Playing it with a Connected Instrument.....	33
3.6.7	Saving and Recalling Settings.....	34
3.6.8	Advanced Features and Examples.....	34
3.7	Customizing the Software	35
3.7.1	Customizing Your Workspace.....	35
3.7.2	Changing Colors and Default Configuration.....	36
3.8	Getting Information and Help	38
4	About the R&S Pulse Sequencer	39
5	Organizing the Project Data in Repositories	43

5.1	Repository Settings.....	43
5.2	User Account Settings.....	47
5.3	Repository Manager Settings.....	47
5.4	How to Manage the Project Data.....	49
6	Selecting a Suitable Scenario and Creating Scenarios.....	54
6.1	Overview of the Available Scenarios and Their Complexity.....	54
6.2	Scenario Settings.....	55
6.2.1	Common Scenario Settings.....	56
6.2.2	Scenario Specific Settings.....	63
6.3	How to Select and Create a Test Scenario.....	68
7	Creating a Pulse Library.....	70
7.1	Basics on Pulse Signals and Pulse Generation.....	70
7.2	Pulse Settings.....	72
7.2.1	Pulse Name and Comment.....	73
7.2.2	Pulse Timing Settings.....	74
7.2.3	Pulse Level Settings.....	77
7.2.4	Pulse Envelope Settings.....	80
7.2.5	Modulation on Pulse (MOP) Settings.....	83
7.2.6	Built-In Modulation Types and their Settings.....	86
7.2.6.1	AM and ASK Modulation.....	87
7.2.6.2	FM and FSK Modulation.....	89
7.2.6.3	Chirp Modulation.....	91
7.2.6.4	Phase Modulation.....	93
7.2.6.5	QPSK Modulation.....	98
7.2.6.6	Noise.....	98
7.2.6.7	Plugins.....	99
7.2.6.8	Data Source, Coding, Filter.....	100
7.2.7	Envelope and Modulation Graphs.....	101
7.3	How to Create a New Pulse and Adjust Its Settings.....	104
8	Building Pulse Sequences.....	111
8.1	About the Sequencing Principles.....	111
8.2	Sequence Settings.....	113

8.2.1	Sequence Description Settings.....	114
8.2.2	Pulse Repetition Settings.....	118
8.2.3	Loop Settings.....	119
8.2.4	Overlay Settings.....	121
8.2.5	Fillers Settings.....	121
8.2.6	Lists with Multiple Sequences.....	123
8.3	How to Create Sequences and Use the Control Elements.....	124
9	Defining and Enabling Inter-Pulse Modulation Effects.....	134
9.1	IPM Profiles Settings.....	137
9.1.1	Common IPM Settings.....	137
9.1.2	IPM Profiles Settings.....	139
9.1.3	Edit List Settings.....	147
9.1.4	Configure Inter-Pulse Modulation (IPM) Settings.....	148
9.2	How to Create IPM Profiles and Use Them to Vary Pulse Parameters.....	150
10	Defining Antenna Patterns and Antenna Scans.....	161
10.1	Overview of the Main Antenna Parameters.....	161
10.2	Antenna Pattern Settings.....	164
10.3	Antenna Scans Settings.....	179
10.4	How to Create a Library with Antenna Patterns and Scans.....	186
11	Emulating Emitters.....	196
11.1	Emitter Settings.....	198
11.2	3D Emitter Preview Settings.....	202
11.3	Lists with Multiple Emitters.....	203
11.4	How to Create and Configure Emitters.....	205
12	Emulating Receivers.....	211
12.1	Receiver Properties.....	213
12.2	Receiver Settings.....	214
12.3	DF System Configuration Settings.....	217
13	Working with Waveforms and Generating Interfering Signals.....	219
13.1	Waveform Settings.....	223
13.2	How to Create a Waveform Scenario and Work with Waveforms.....	232

14	Creating and Mapping Generator Profiles.....	237
14.1	Profile to Instrument Links Settings.....	237
14.2	Generator Profile Settings.....	240
14.3	How to Create and Map Generator Profiles.....	244
15	Creating Complex 2D Scenarios with Receiver, Emitters and Interferers.....	248
15.1	2D Map Settings.....	248
15.2	3D Scan Pair View Settings.....	250
15.3	Lists with Multiple Emitters and Interferers.....	253
15.4	Lists with Multiple Background Emitters.....	258
15.5	How to Create Scenarios with Emitters, Interferers and a Receiver.....	258
16	Visualizing and Analyzing Waveforms.....	264
16.1	Waveform and Data View Settings.....	264
16.2	Waveform Reference Level Settings.....	267
16.3	How to Analyze the Content of Waveform Files and Files with I/Q Data.....	269
17	Playing the Generated Waveform Files.....	272
17.1	Waveform Generation Settings.....	272
17.2	Assign Signal Source or Receiver Signals to Generator Profile Settings.....	275
17.3	Signal Generator Remote Control Settings.....	280
17.4	Instrument Control Panel Settings.....	281
17.5	How to Create, Transfer and Play the Waveforms of Simple Scenarios.....	284
17.6	How to Play the Waveforms of Emitter-Based 2D Scenarios.....	287
18	Defining the Data Source.....	291
18.1	Data Sources Settings.....	293
18.2	How to Configure the Bit Stream Used by the MOP.....	295
19	Defining Complex Modulation Schemes and IPM Profiles.....	297
20	Defining and Enabling Marker Signals.....	301
20.1	Marker Settings.....	301
20.2	How to Configure and Visualize Markers.....	303
21	Creating Reports and Documenting Measurement Results.....	308
21.1	Reporting Settings.....	309

21.2	How to Create Test Reports.....	312
22	Automation of R&S Pulse Sequencer.....	315
22.1	Remote Settings.....	316
22.2	Script Manager Settings.....	317
22.3	How to Configure and Enable Remote Control of R&S Pulse Sequencer.....	318
23	Remote-Control Commands.....	321
23.1	Conventions used in SCPI Command Descriptions.....	321
23.2	Programming Examples.....	322
23.3	Common Commands.....	322
23.4	Commands with Similar Syntax.....	323
23.5	Antenna Pattern Commands.....	337
23.6	Antenna Scan Commands.....	348
23.7	Dashboard Commands.....	356
23.8	Data Source Commands.....	360
23.9	Signal to Generator Mapping Commands.....	362
23.10	Emitter Commands.....	366
23.11	Generator Profiles and Instruments Commands.....	369
23.12	Inter-Pulse Modulation Commands.....	373
23.13	Marker Commands.....	385
23.14	Plugin and Reporting Commands.....	390
23.15	Pulse Commands.....	395
23.16	Receiver Commands.....	415
23.17	Repository Commands.....	418
23.18	Scenario Commands.....	425
23.19	Sequence Commands.....	445
23.20	Status Commands.....	454
23.21	System Commands.....	455
23.22	Waveform Commands.....	457
23.23	Waveform Viewer Commands.....	461
23.24	List of Remote Commands.....	462
24	Querying Error Messages and Troubleshooting.....	477
	Annex.....	480

A	Supported File Types and File Formats.....	480
A.1	File Format of the Reporting Template.....	481
A.2	Antenna Pattern File Formats.....	485
B	Formula Syntax.....	492
C	Plugin Programming API.....	495
C.1	Common Functions.....	495
C.2	MOP Functions.....	500
C.3	IPM Functions.....	505
C.4	Custom Antenna Pattern Import Functions.....	506
C.5	List of Plugin Functions.....	509
D	Scripting API.....	510
D.1	Global Script Extensions.....	510
D.2	Newable Script Extensions.....	512
D.2.1	QVisa.....	512
D.2.2	QDir.....	515
D.2.3	QFile.....	515
D.2.4	QFileDialog.....	516
D.2.5	QMessageBox.....	517
D.3	Script Examples.....	518
D.4	List of Script Objects and Methods.....	519
E	Common Coding Algorithms.....	521
F	Supported Filter and Impact of the Filter Parameters.....	522
G	References.....	524
	Glossary: Terms and Abbreviations.....	525
	Index.....	528

1 Preface

1.1 About this Manual

This User Manual provides all the information **specific to the R&S Pulse Sequencer** and describes the software and all its options.

This description assumes that all available options are installed. Required options are stated at the beginning of each section or in the parameter description, if one particular parameter required additional option. If your software does not support all options, consider the corresponding parts as not relevant.

The main focus in this manual is on the provided settings and the tasks required to generate a signal. The following topics are included:

- **Welcome to the R&S Pulse Sequencer options R&S SMW/SMBV/SGT-K300/-K301**
Introduction to and getting familiar with the option
- **Getting Started**
Information needed to install and start working with the software, as well as basic operation and typical signal generation examples
- **About the R&S Pulse Sequencer**
Background information on basic terms and principles in the context of the signal generation
- **R&S Pulse Sequencer Configuration and Settings**, logically divided in several topics
A concise description of all functions and settings available to configure signal generation with their corresponding remote control commands
- **How to Generate a Signal with the R&S Pulse Sequencer Options**
The basic procedure to perform signal generation tasks and step-by-step instructions for more complex tasks or alternative methods.
Detailed examples guide you through typical signal generation scenarios and allow you to try out the application immediately
- **Typical Application Examples**
Example signal generation scenarios in which the software is frequently used.
- **Remote Control Commands incl. List of Remote Commands**
Remote commands required to configure and perform signal generation in a remote environment, sorted by tasks
Programming examples demonstrate the use of many commands and can usually be executed directly for test purposes
Includes an alphabetical list of all remote commands described in the manual (Commands required to set up the vector signal generator or to perform common tasks on the instrument are provided in the user manual of the base unit)
- **Annex**
Reference material, such as extensive lists, tables or overviews
- **Glossary**
Alphabetical list of often used terms and abbreviations

- **Index**

1.2 Documentation Overview

The user documentation for R&S Pulse Sequencer consists of the following parts:

- Online help system included in the software
- User manual and online manual provided on the product page
- Release notes provided on the product page
- Data sheet and brochures provided on the product page
- Application notes provided on the Rohde & Schwarz website



You find the R&S Pulse Sequencer user documentation on the product page of the signal generator, for example at:

<http://www.rohde-schwarz.com/product/SMW200A.html> > "Downloads > Manuals".

Further download paths are listed in the abstracts of the documentation types.

Online Help

Offers quick, context-sensitive access to the information needed for operation and programming.

User Manual and Online Manual

Introduces the R&S Pulse Sequencer and describes how to start working with the product. Includes the description of the graphical user interface, basic operations, typical examples, and general information. Contains also the description of all SCPI remote control commands and programming examples.

The **online manual** provides the contents of the user manual for immediate display on the internet.

Data Sheet and Brochures

The data sheet contains the technical specifications of the R&S Pulse Sequencer, see the product page of the signal generator, for example at:

<http://www.rohde-schwarz.com/product/SMW200A.html> > "Brochures and Data Sheets"

Release Notes

Describes the software installation, new and modified features and fixed issues according to the current software version. You find the latest version for example at:

<http://www.rohde-schwarz.com/product/SMW200A.html> > "Downloads > Software".

Application Notes, Application Cards, White Papers

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

1.3 Conventions Used in the Documentation

1.3.1 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.
<i>Input</i>	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.

1.3.2 Conventions for Procedure Descriptions

When describing how to operate the software, several alternative methods may be available to perform the same task. In this case, the procedure using the menu bar is described. Many elements that can be accessed from menus can also be accessed by clicking an icon or using the context menu. Alternative procedures are only described if they deviate from the standard operating procedures.

The term "select" may refer to any of the described methods.

1.3.3 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 R&S Pulse Sequencer

The R&S Pulse Sequencer is a software that generates complex pulse signals by using predefined, configurable test scenarios with different complexity. You can simulate the signals of different emitter and receiver configurations, including antennas and scan types. The signal can be played by the instruments listed below, for example to perform radar receiver tests.

- R&S SMW
- R&S SMBV
- R&S SGT

The R&S Pulse Sequencer software is a stand-alone, PC-based application that creates waveform files. This software is available for free download on the R&S website.

Pulse Sequencer features:

- ARB-based signal generation
- Multi-segment waveform sequencing
- Pulse shape definition with rise and fall time, droop, ripple, overshoot
- Modulation on pulse with all major formats like chirps, Barker Codes, Polyphase Codes, PSKs, AM, FM
- Single pulse, pulse train generation with repetition count per pulse
- Inter-pulse modulation of amplitude, phase, frequency, etc. values from pulse to pulse
- Internal and external data sources for modulation
- Import of waveform files for sequencing with repetition count

Advanced Pulse Sequencer key features:

- ARB-based signal generation and multi-segment waveform sequencing
- Single pulse and pulse train generation with repetition count per pulse
- Powerful sequencing tool with loops, nested loops, subsequences, and overlays
- Antenna diagram definition and antenna scan definition
- Antenna diagrams like pencil beams, cosecant beams, Gaussian diagrams, user defined antenna diagrams, phased array antenna diagrams
- Antenna scan types like helical scans, circular scans, conical scans
- Emitter definition by waveforms, antenna diagram, antenna scan, attitude information, EIRP, and carrier frequency
- Receiver definition by antenna diagram, antenna scan, and attitude information
- Calculation of signal under consideration of one-way free space propagation according to emitter and receiver location on the 2D map
- Import of R&S®WinIQSIM2™ or customer waveforms for interference generation on the 2D map

The R&S SMW-K308 option Direction Finding extends the combination R&S SMW-K300/-K301 with the following features:

- Dedicated direction finding scenario and receiver element with multiple antennas and individual positioning
- Generation of individual waveforms for each receive antenna

This user manual contains a description of the functionality that the software provides, including remote control operation.

Functions of the signal generator are not described here; they are described in the user manual of the corresponding base unit. The latest version is available for download at the instrument's page, for example at [R&S SMW200A product page](#).

Installation

You can find detailed installation instructions in the delivery of the option or in [Chapter 3.2, "Installing the Software"](#), on page 15.

2.1 Accessing the Pulse Sequencer

To open the application

- ▶ On your PC, select "Start > All Programs > R&S Pulse Sequencer > R&S Pulse Sequencer".

The software opens and per default displays the last opened workspace.

3 Getting Started

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

• Required Options	14
• Installing the Software	15
• Starting the R&S Pulse Sequencer for the First Time	17
• Understanding the Displayed Information	18
• Means of Users Interaction	22
• Trying Out the Software	25
• Customizing the Software	35
• Getting Information and Help	38

3.1 Required Options

The R&S Pulse Sequencer software is a stand-alone, PC-based application that creates waveform files. This software is available for free download on the R&S website. Waveforms generated with the software can be played by a subset of instruments, that must fulfill the minimum requirements listed below.

R&S SMW200A

R&S SMW200A base unit equipped with:

- Option Baseband Generator (R&S SMW-B10)
- Option Frequency (R&S SMW-B1xx/-B2xx)
- Option Baseband main module, one I/Q path to RF (R&S SMW-B13)
- Option Pulse Sequencer (R&S SMW-K300) per signal path

Additional non-mandatory options:

- Option Multiple Entities R&S SMW-K76
- Option Advanced Features for Pulse Sequencer (R&S SMW-K301) per signal path
- Option Direction Finding for Pulse Sequencer (R&S SMW-K308) per signal path (requires option R&S SMW-K301 and option R&S SMW-K76)
- Further options depending on the particular application
For example, if waveform files generated with R&S®WinIQSIM2™ is used within the sequences, the corresponding R&S SMW-K2xx options

R&S SMBV100A

R&S SMBV100A base unit equipped with:

- Option Baseband Generator (R&S SMBV-B51)
- Option Memory Extension for ARB (R&S SMBV-K511)
- Option Frequency (R&S SMBV-B1xx)
- Option Pulse Sequencer (R&S SMBV-K300)

Additional non-mandatory options:

- Option RF Bandwidth Extension (R&S SMBV-K521/-K522)
- Option Advanced Features for Pulse Sequencer (R&S SMBV-K301)
- Further options depending on the particular application
For example, if waveform files generated with R&S®WinIQSIM2™ is used within the sequences, the corresponding R&S SMBV-K2xx options

R&S SGT100A

R&S SGT100A base unit equipped with:

- Option Frequency (R&S SGT-KB106)
- Option Baseband Generator (R&S SGT-K510)
- Option Pulse Sequencer (R&S SGT-K300)

Additional non-mandatory options:

- Option Memory Extension for ARB (R&S SGT-K51x)
- Option Advanced Features for Pulse Sequencer (R&S SGT-K301)
- Further options depending on the particular application
For example, if waveform files generated with R&S®WinIQSIM2™ is used within the sequences, the corresponding R&S SGT-K2xx options

3.2 Installing the Software

Software updates and the Release Notes describing the improvements and modifications are available for download at the [product page of the signal generator, e.g. R&S SMW](#).

Software and Hardware Requirements

The following minimum requirements have to be met:

- **Software**
 - Microsoft®Windows 7 Professional 32 Bit (SP1) operating system
 - R&S®VISA (Virtual Instrument Software Architecture) software library other VISA runtime library
The R&S®VISA is provided for download at the Rohde & Schwarz website <http://www.rohde-schwarz.com/rsvisa>.
- **Hardware**

Table 3-1: Hardware requirements

	Minimum requirements	Recommended hardware
AMD or Intel CPU	Dual Core, 2 GHz	Intel Quad-Core i7, Quad-Core Xeon, AMD FX series
RAM	≥ 2 GB	16 GB
Video	NVIDIA Quadro 128 MB or ATI Radeon	NVIDIA Quadro 128 MB or ATI Radeon

	Minimum requirements	Recommended hardware
Video resolution	≥ 1280 x 1024 pixels	≥ 1920 x 1200 pixels
Rendering	OpenGL, shader model 3	OpenGL, shader model 3
Network	LAN 1 GB/s	LAN 1 GB/s

To install the software

Download the R&S Pulse Sequencer software from the R&S website.

The R&S Pulse Sequencer software consists of the file

PS-Install <major>.<minor>.<build>.<release>.msi.

1. Start the PS-Install <major>.<minor>.<build>.<release>.msi file.

The file name follows the naming conventions:

- <major>.<minor> is the software version
- <build> is the build number.
This is the day elapsed since 1 January 2000
- <release> indicates the release on the build date.
This is 1/10 of the number of seconds elapsed since midnight.

2. Depending on your user rights, select one of the following:

- a) Installation for all users (default)

- Requires administrator rights
- Is a per-machine installation
- Project data and program data are common to all users
- Users keep their Individual settings

- b) Installation for current user only

- Does not require administrator rights
- Is a per-user installation
- Installation in %HomePath%

3. Choose the setup type.

4. Follow the installation instructions.

The installation is completed. Per default, a shortcut icon is created on the desktop.

The software creates the folder structure listed in [Table 3-2](#).

Table 3-2: Default file location (software installation for all users)

File type	File location
Program data, SDK files	%PROGRAMFILES(X86)%\Rohde-Schwarz\Pulse Sequencer (Path)
Project data (Repositories)	%PUBLIC%\Public Documents\Rohde-Schwarz\ Pulse Sequencer (DataPath)

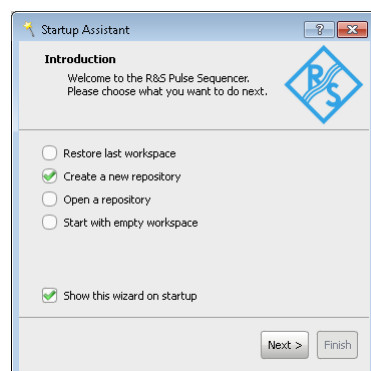
File type	File location
Report files	%HOMEPATH% (HomePath OR ReportPath)
User settings (Workspace, etc.)	%HOMEPATH%\AppData\Roaming\Rohde-Schwarz\ Pulse Sequencer

For information on how to query the file location, the version of the used VISA library, the list of all loaded modules, see ["To access the Debug Information dialog"](#) on page 478.

3.3 Starting the R&S Pulse Sequencer for the First Time

1. On your PC,
 - a) Select "Start > All Programs > R&S Pulse Sequencer > R&S Pulse Sequencer"
 - or
 - b) Double click on the shortcut icon on the desktop.

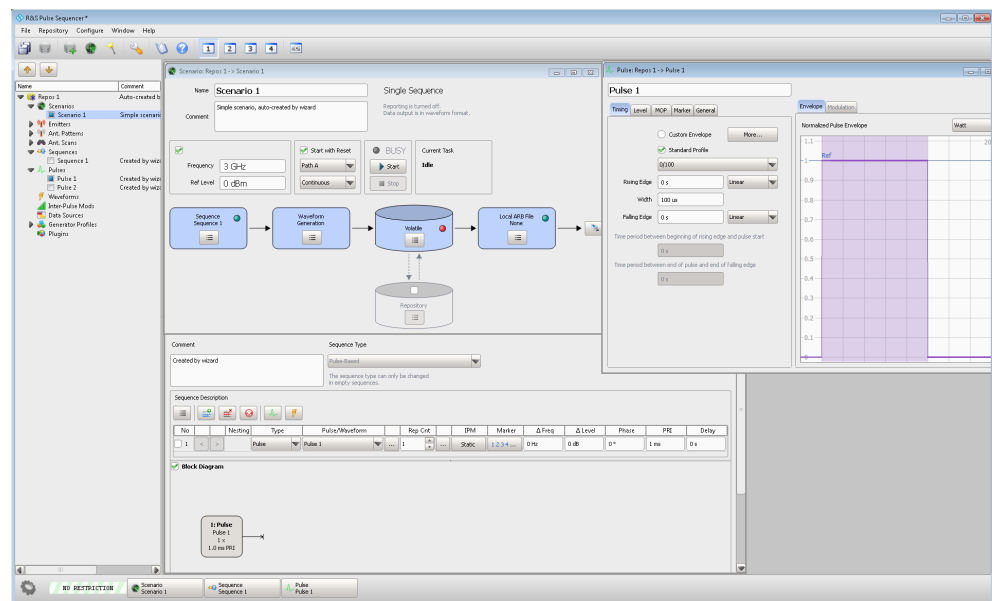
The "Startup Assistant" opens.



2. Select "Create a new Repository" and select "Next".
3. Use the default settings and follow the instructions.

A new repository ("Repos 1") is created that contains a simple scenario ("Scenario 1").

Your workspace should resemble the one shown on the following figure.



See also:

- [Figure 3-1](#) (in [Chapter 3.4, "Understanding the Displayed Information"](#), on page 18)
- [About the R&S Pulse Sequencer](#), section "Introduction to the Software Concept" on page 39
- [Chapter 3.6, "Trying Out the Software"](#), on page 25

3.4 Understanding the Displayed Information

The following figure shows an example test scenario during software operation. All different information areas are labeled. They are explained in more detail in the following sections.

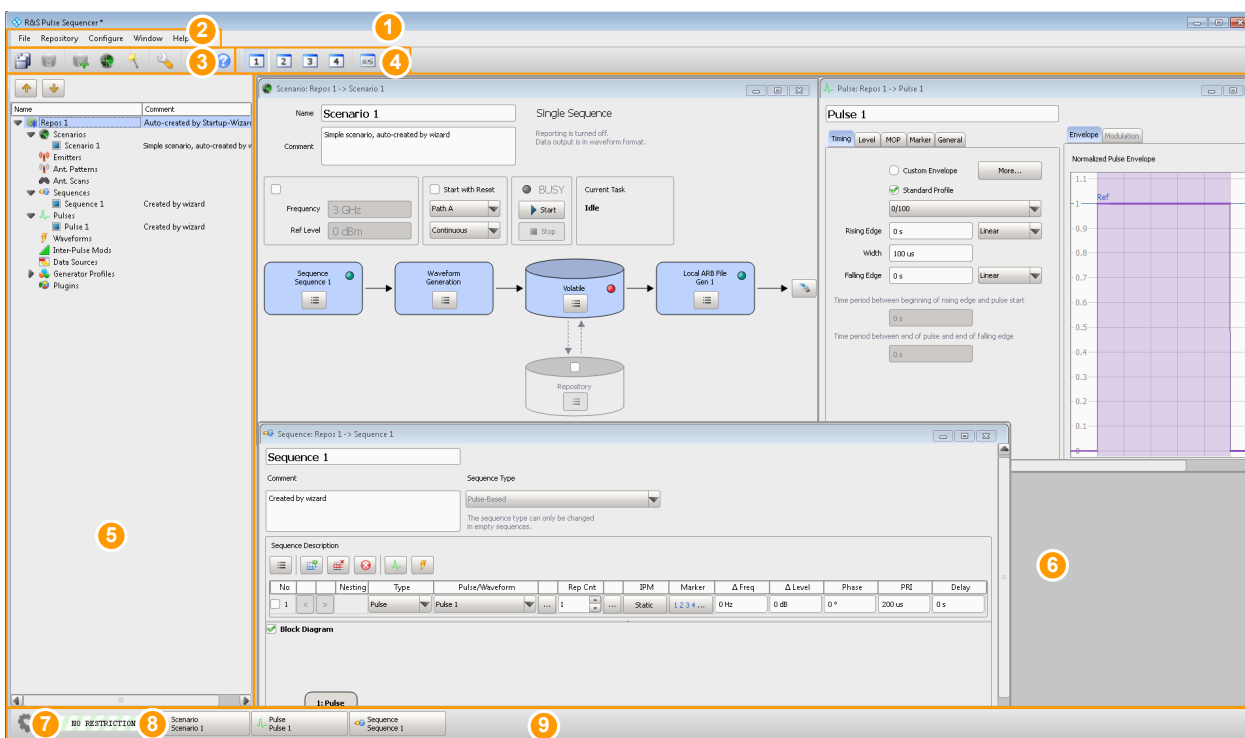


Figure 3-1: R&S Pulse Sequencer default workspace: understanding the displayed information

- 1 = Title bar
- 2 = Menu bar with standard functions, e.g. save/open file etc.
- 3 = Toolbar
- 4 = Screens for individual scenarios
- 5 = Project tree (repository tree)
- 6 = Work area
- 5+6 = Workspace
- 7, 8, 9 = Task bar with indication on enabled remote control, information on the required highest security level (if enabled) and access to active dialogs

The user interface of the R&S Pulse Sequencer offers an intuitive operating concept. Operation is similar to any operating system. All menus and dialogs use known elements, e.g., selection lists, check boxes, and entry fields.

Menu bar

Table 3-3: Content of the menu bar

File	Repository	Configure	Script	Window	Help
<ul style="list-style-type: none"> • New Repository • Load and Manage Repository • Import Repository Archive • Save Repository • Clear Workspace • Open Workspace • Save Workspace • Save Workspace As • Exit 	Quick access to currently loaded repository	<ul style="list-style-type: none"> • Profile to Instrument Links • Remote Control • Settings • Colors 	Manage...	<ul style="list-style-type: none"> • Console • Message Log • Dashboard • Cascade • Minimize All • Close All • Screen 1 .. 4 	<ul style="list-style-type: none"> • About • Online Help • Wizard • Debug Information

Toolbar

Standard functions can be performed via the icons in the toolbar at the top of the screen.



Figure 3-2: Standard functions in the toolbar

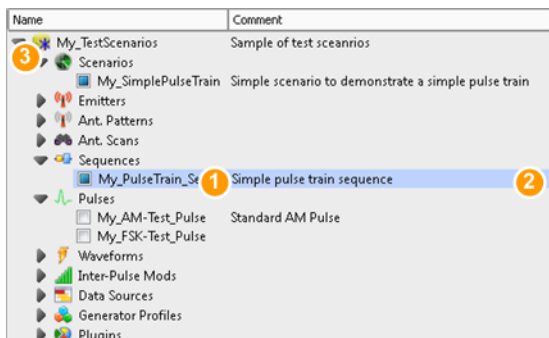
- 1 = Save as: stores current workspace configuration
- 2 = Access the "Manage Repository" dialog for loading, discarding or exporting repositories
- 3 = Creates a repository
- 4 = Creates a scenario
- 5 = Starts a wizard
- 6 = Access the "Program Settings" dialog
- 7 = Opens the message log window
- 8 = Dashboard: "Instrument Control Panel" window
- 9 = Help: displays context-sensitive help topic for currently selected element
- 10 = Screen: switches between the virtual screens
- 11 = Hardcopy: copies current dialog to the clipboard

Virtual screens

Using the R&S Pulse Sequencer you can handle several scenarios with even comprehensive settings at the same time. You can arrange logically the dialogs across multiple screens, so that you easily switch between them. You can for example assign the subset of antenna dialogs to one screen and the pulse and modulation dialogs to another.

Repository tree

The repository tree shows the content of all currently loaded repositories. Repository elements are indicated with the assigned icons, their names, and the first comment line.

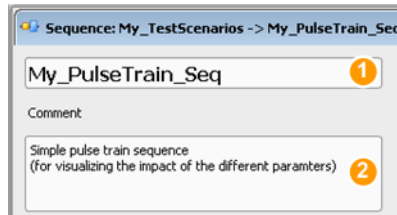


- 1 = Name, see "Name and comment" on page 20
- 2 = First comment line
- 3 = Indicates unsaved changes in the repository

Name and comment

Each repository element is identified with its unique name and optional description entered in the comment field. The description may contain several lines; new line is added by pressing SHIFT+ENTER.

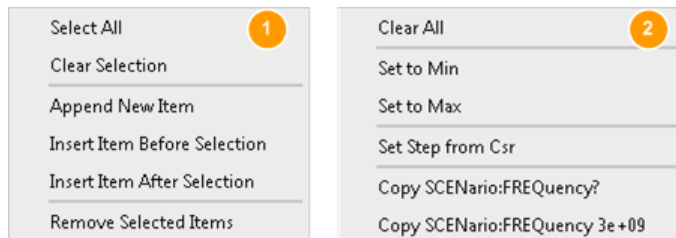
See also "Dialog names" on page 21.



1 = A unique name to indicate the element in the repository
2 = Description; the first line is displayed in the repository tree

Context-sensitive menus

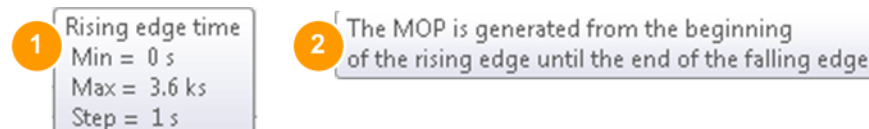
All user interface controls provide a context-sensitive menu. Context-sensitive menus are accessed with a right mouse click on the control.



1 = Context menu of an icon
2 = Context menu of a parameter

Tooltips

Tooltips indicate the possible value range of a parameter or list overview information on current settings.



1 = Tooltip of a setting parameter, indicating the allowed value range
2 = Tooltip with detailed information on a setting

Dialog names

The name of a dialog indicates its content or the functionality this dialog covers. The used naming convention helps you to recognize which data base element you are configuring, to which scenario does it belong and what is its current name.

"<Repository Element>: <Scenario Name> -> <Function Name>"

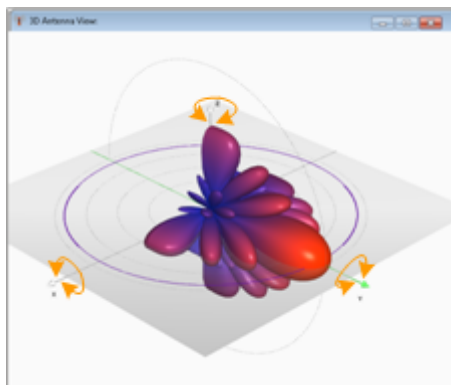


1 = Repository element
2 = Scenario name
3 = Function name

Interactive 3D displays

Several dialogs display the configured settings as 2D or as interactive 3D models or full 3D antenna radiation patterns.

Turning the mouse wheel zooms in and out on these dialogs. Moving the mouse while holding the left mouse key rotates the 3D view around its origin.



3.5 Means of Users Interaction

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

There are two ways to operate the R&S Pulse Sequencer:

- Manual operation:
Run the software on your PC and use the mouse and/or keyboard.
- Remote control:
Create programs to automate settings, tests and measurements. The software is controlled by a program, running on the same or on another computer.
This way of operation is described [Chapter 22, "Automation of R&S Pulse Sequencer"](#), on page 315.

Basic operation concept

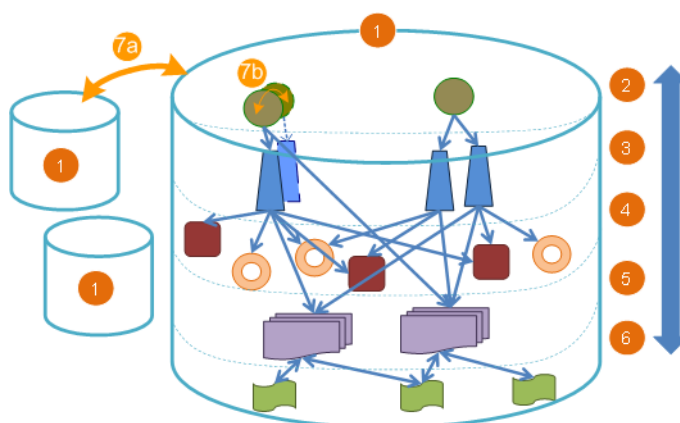
This section lists settings and functions that are common to the dialogs and are performed in the same manner throughout the software. You can use them alternatively or complementary.

As described in "[Introduction to the Software Concept](#)" on page 39, the R&S Pulse Sequencer software is based on a relational database that organizes database elements within repositories. Database elements are indicated by their names; within one repository, element names must be unique (see "[Name and comment](#)" on page 20).

Creating elements

Once created, database elements can be used by other elements within the same repository. Elements can be created in both the bottom-up and the top-down direction, or in a combination of these two methods:

- *Bottom-up*: you can create your pulse and waveform library first, then build sequences, create scenarios, and use the created sequences or assign them to emitters
- *Top-down*: create your scenario, create emitter, define the antenna pattern/scan, define the used sequence, and define the pulses and waveforms



- 1 = Repository, one or more
 2 = Scenario level; scenarios describe one or more emitters and one or more sequences
 3 = Emitter level; emitters are described among others with antenna patterns and antenna scans
 4 = Antenna patterns and antenna scans
 5 = Sequence level; sequences are built of one or more pulses or waveforms
 6 = Pulse and waveform level
 7a, 7b = Creating elements at any level with clone and copy

For information on the naming conventions and description of the database elements, see "[Introduction to the Software Concept](#)" on page 39.

Handling database elements

While creating database elements, consider also the following possibilities:

- **Change**
Files describing elements are stored once but may be used several times. When one file is changed, its parent elements are also updated.
- **Clone**
Elements can be cloned to create a deep copy of the database element
- **Delete**
Elements can be deleted if they are not referenced by other elements
- **Copy**
If two repositories are opened in the project tree, a drag&drop operation creates a copy of the selected element and all referenced elements










Standard function in the context menus



Each dialog with settings belonging to a „higher-level“ element provides standard functions to:

- Assign existing elements
- Create elements
- Edit the existing assigned element

You can access these functions from the context sensitive menus of the icons listed in [Table 3-4](#).

Table 3-4: Overview of the used icons and their functions

Icon	Name	Function/Description
	<Function> Menu (the name changes depending on the situation)	Smart menu with several options, depending on the current element and situation: <ul style="list-style-type: none"> • New (Item) • Append item • Prepend item • Insert Item • Insert Item Before/After • Delete Item • Clear (Selection) • Edit • Select • Rename
	More	Select Edit New
	Pulse, Waveform	Creates repository element
	Antenna Pattern, Antenna Scan, Sequence	Accesses a dialog with detailed settings
	Emitter, interferer, receiver	
	3D, 2D diagram	
	Append item, Remove item, Delete	Appends or removes item, deletes all items
	Left/Right , Up/Down	Reorders items
	Select	Selects a table row or an item for editing
	Enable/Disable	Activates or deactivates settings

Icon	Name	Function/Description
	Hardcopy, Print	Copies current screen or dialog to the clipboard; Prints current diagram
	Split window	Detaches the diagram and opens it in a separate window

Standard functions for elements handling

Supported are the following standard direct interaction functions:

- Double-click
Opens the selected element for editing
- Clone
Creates a copy of the selected element
- Drag&Drop
A drag&drop operation:
 - Creates a copy of the selected element, together with all referenced elements
 - Assigns an element to a 2D view
- Mouse wheel
Turning the mouse wheel zooms in and out

3.6 Trying Out the Software

This chapter introduces the most important functions and often used settings of the R&S Pulse Sequencer step by step. The complete description of all functions and their usage is provided in the corresponding main chapters of this user manual.

Prerequisite

The software is running and started up as described in [Chapter 3.3, "Starting the R&S Pulse Sequencer for the First Time"](#), on page 17.

For the first signal generation tasks, you use the *on-demand Wizard* to generate an ARB signal, so you do not need any external instruments. More complex signal generation tasks, however, may require an external signal generator equipped with the options listed in [Chapter 3.1, "Required Options"](#), on page 14.

The software is manually operated. Try out the following:

- [Completing the Scenario that was Automatically Created upon Start Up](#)..... 26
- [Generating an ARB Waveform File](#)..... 27
- [Verifying the Generated Signal in the Waveform View](#)..... 28
- [Launching the Built-In Wizard](#)..... 29
- [Using the Wizard to Create a Complex Scenario](#)..... 30
- [Transferring the Waveform to and Playing it with a Connected Instrument](#)..... 33
- [Saving and Recalling Settings](#)..... 34
- [Advanced Features and Examples](#)..... 34

3.6.1 Completing the Scenario that was Automatically Created upon Start Up

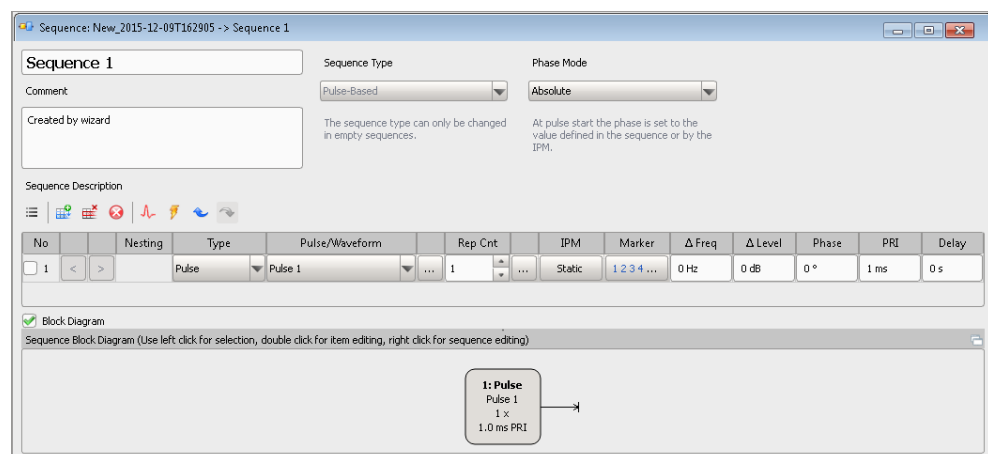
We assume that you have started the software as described in [Chapter 3.3, "Starting the R&S Pulse Sequencer for the First Time"](#), on page 17. Your workspace should resemble the one shown on [Figure 3-1](#) (see section [Chapter 3.4, "Understanding the Displayed Information"](#), on page 18).

Your repository contains:

- "Pulse 1": an unmodulated pulse with pulse width of 100 us and rise and fall times set to zero
- "Sequence 1": a sequence containing one single pulse, "Pulse 1". This pulse is processed once ("Rep. Count = 1") and has a pulse repetition interval "PRI = 3 ms".

To create a sequence with 10 pulses, each having a PRI of 1 ms

1. In the repository tree, select "Scenario > Sequence 1".
2. In the "Sequence Description" table, select:
 - a) "Rep. Count = 10"
 - b) "PRI = 1 ms"



For information on the provided settings, see:

- [Chapter 8.2, "Sequence Settings"](#), on page 113
- [Chapter 7.2.2, "Pulse Timing Settings"](#), on page 74

To assign the sequence to the scenario

1. In the repository tree, select "Scenario > Scenario 1".
The "Scenario" dialog opens and displays a block diagram of the signal flow.
2. In the block diagram, select "Sequence" and select the icon to open the context menu.
3. Select "Select > Sequence 1".
The status indicator in the "Sequence" block is green.

The status indicator in the "Volatile" block is still red.

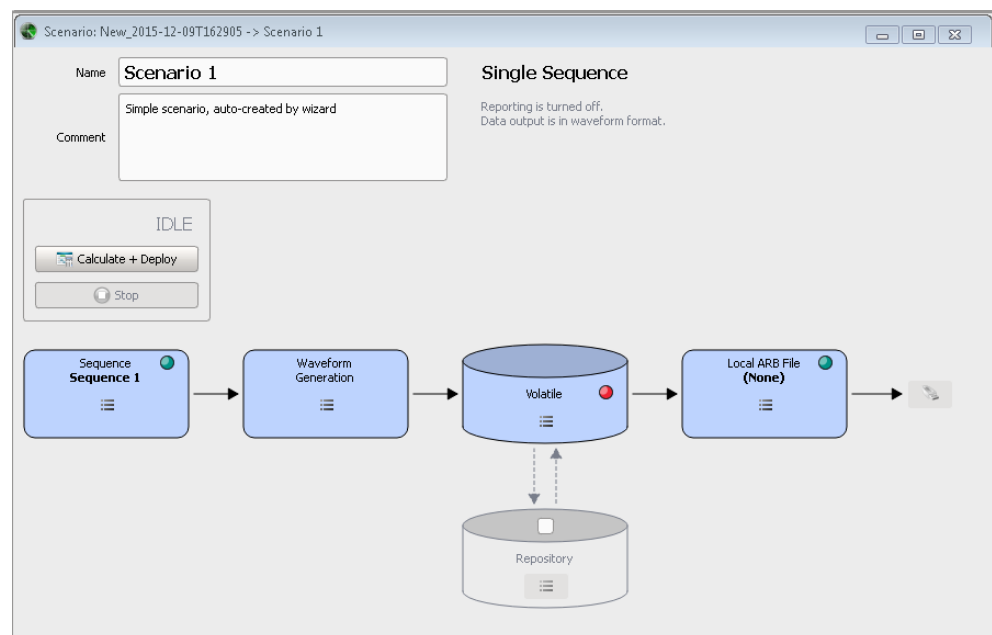
3.6.2 Generating an ARB Waveform File

The initial situation is the configuration described in ["To assign the sequence to the scenario"](#) on page 26.

To store the created ARB file on the file system of your PC

1. In the repository tree, select "Scenario > Scenario 1".
The "Scenario" dialog opens and displays a block diagram of the signal flow.
2. In the block diagram, select "Local ARB File" and select the icon to open the context menu.
3. In the context menu, select "Set Path...".
4. In the Windows Explorer, navigate to the directory in that the file should be stored.

The block diagram shows the settings.

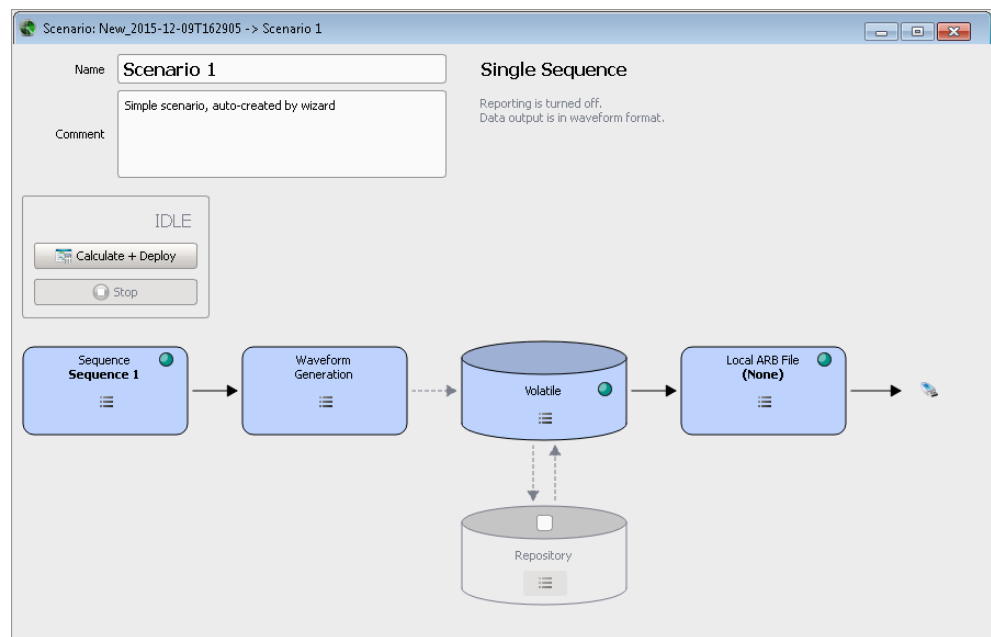


To start the ARB waveform generation

If all preconditions are fulfilled, the "Calculate + Deploy" button is active and you can start signal generation.

- ▶ In the "Scenario" dialog, select "Calculate + Deploy".

The R&S Pulse Sequencer creates the waveform and shows the current progress.



A green indication in the "Volatile" block confirms that the generation is completed.

For information on the provided settings, see:

- [Chapter 6.2, "Scenario Settings"](#), on page 55

3.6.3 Verifying the Generated Signal in the Waveform View

It is often useful to check the generated waveform visually, before you transfer the waveform to the vector signal generator. The R&S Pulse Sequencer provides a built-in function, the "Waveform View", to represent the generated signal in a graphical form.

The initial situation is the configuration described in "[To start the ARB waveform generation](#)" on page 27.

To open the Data View and view the generated signal

- ▶ In the "Scenario" dialog, select "Volatile", open the context menu, and select "View".

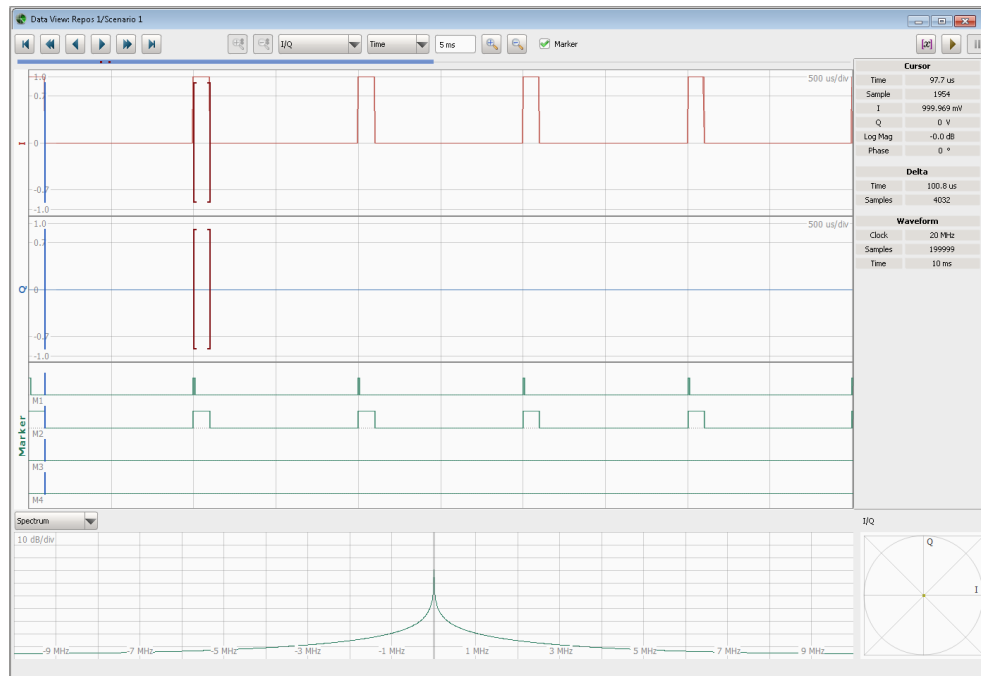


Figure 3-3: Data View: Sequence composed of 10 unmodulated pulses (first 5 pulses shown)

In its default view mode, the "Data View" dialog displays the I and Q data as a function of time, the spectrum of the generated signal and I/Q the constellation diagram.

The displayed settings confirm that:

- Generated is a sequence of 10 pulses but the view port shows only the first 5 pulses
- Each pulse has a PRI = 1 ms
- The waveform is 10 ms long, that is 10*PRI
- Each pulse has a pulse width of 100 us.

For information on the provided settings, see [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264.

3.6.4 Launching the Built-In Wizard

To help you get familiar with the software, the R&S Pulse Sequencer provides a built-in startup assistant, the "Wizard".

You may access this "Wizard":

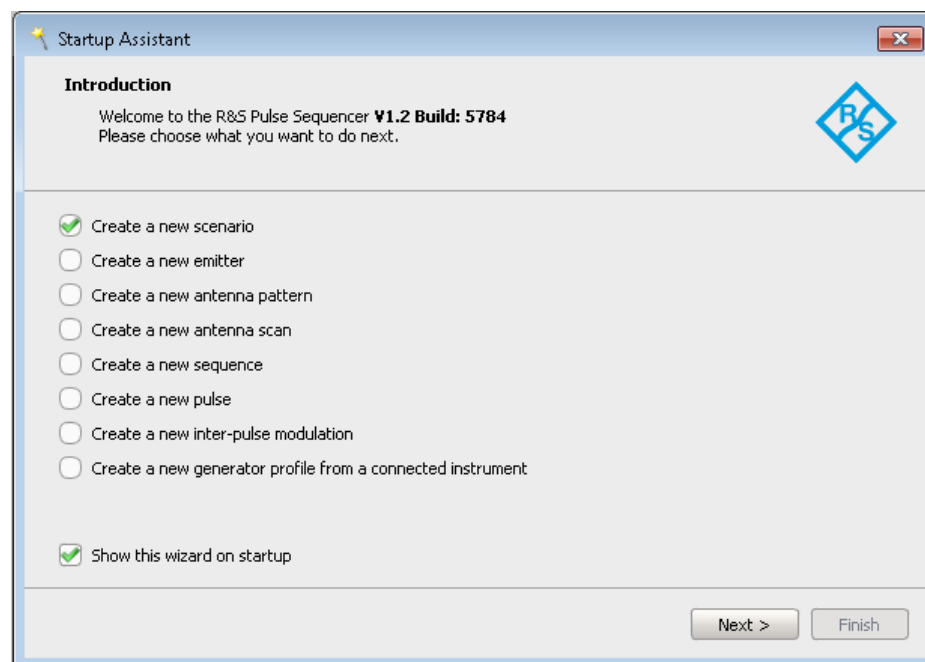
- **On an application start:**
That is each time you start the software
- **On demand:**

That is any time during operation when you need a fast configuration of few settings.

To start the built-in wizard

- ▶ In the tool bar, select "Help > Wizard".

The "Startup Assistant" opens.



3.6.5 Using the Wizard to Create a Complex Scenario

The main application field of the R&S Pulse Sequencer is the generation of pulsed signals. This example uses the "Wizard" to introduce the way you can create and define antenna patterns, antenna scans, emitters, etc. and interact with the software.

To create a new emitter-based scenario

1. Start the "Wizard", see [Chapter 3.6.4, "Launching the Built-In Wizard"](#), on page 29.
2. In the assistant dialog, select "Create New Scenario" and confirm with "Next".
3. Select "Complex Scenarios with emitters, modes, and antennas".
4. Follow the instructions and select:
 - a) Emitter with: "EIRP = 120 dBW", "Frequency = 3 GHz", "Antenna Pattern > On", "Antenna Scan > ON"
 - b) "Antenna Pattern > Gaussian"
 - c) "Scan Pattern > Raster"
 - d) Pulse with "Rising Edge = Falling Edge = 10 us", "Width = 100 us" and enable "Modulation > On".
 - e) "Linear Chirp > On" with "Deviation = 2 MHz"

- f) "PRI = 1 ms" and "Repetition = 5"
5. Confirm with "Finish".

The software creates the scenario automatically with all required elements, e.g. one new emitter (Emitter 1), one new sequence (Sequence 2), new pulse (Pulse 2), etc.

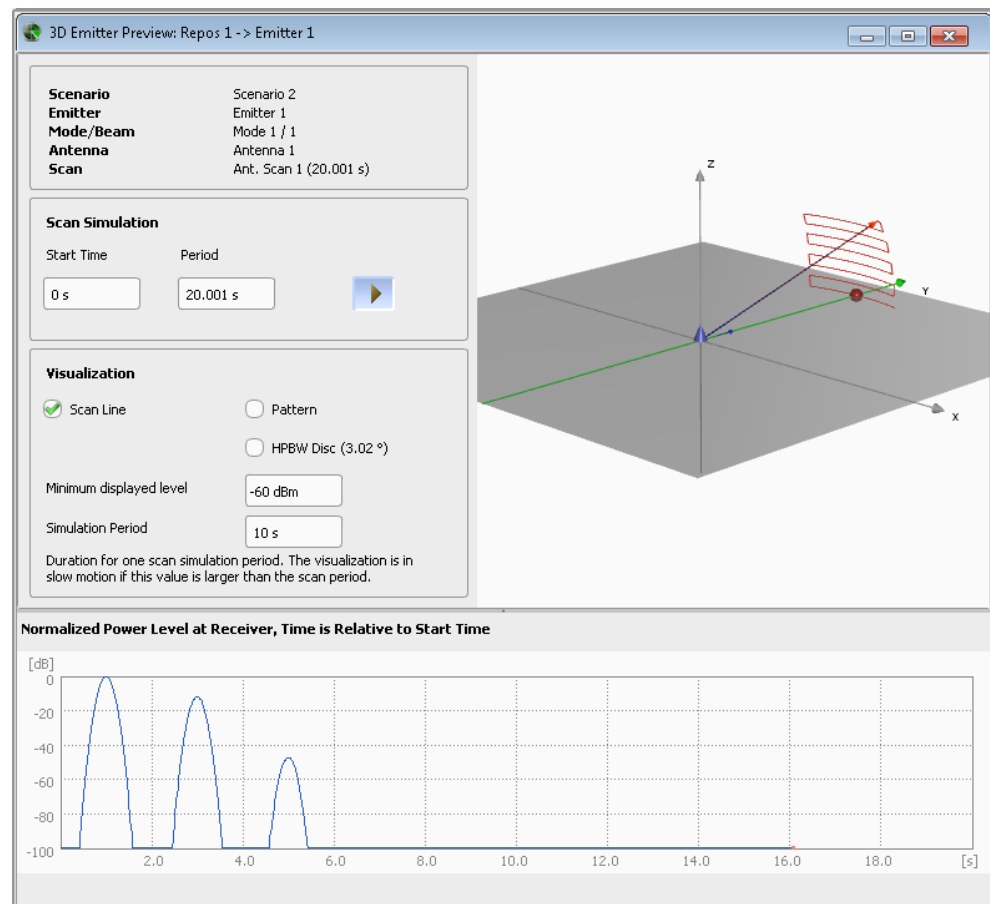
You recognize the created elements by their comment.

For information on the provided settings, see:

- [Chapter 11.1, "Emitter Settings"](#), on page 198
- [Chapter 10.2, "Antenna Pattern Settings"](#), on page 164
- [Chapter 10.3, "Antenna Scans Settings"](#), on page 179
- [Chapter 7.2.5, "Modulation on Pulse \(MOP\) Settings"](#), on page 83

To visualize the signal received by a static receiver

1. Open the automatically created emitter-based scenario ("Scenario 2").
2. In the "Current Emitter" section on the block diagram, select the "3D" icon.



The "3D Single Emitter" dialog displays a live plot of the normalized signal power level at the receiver and an interactive 3D view of the emitter antenna pattern and scan.

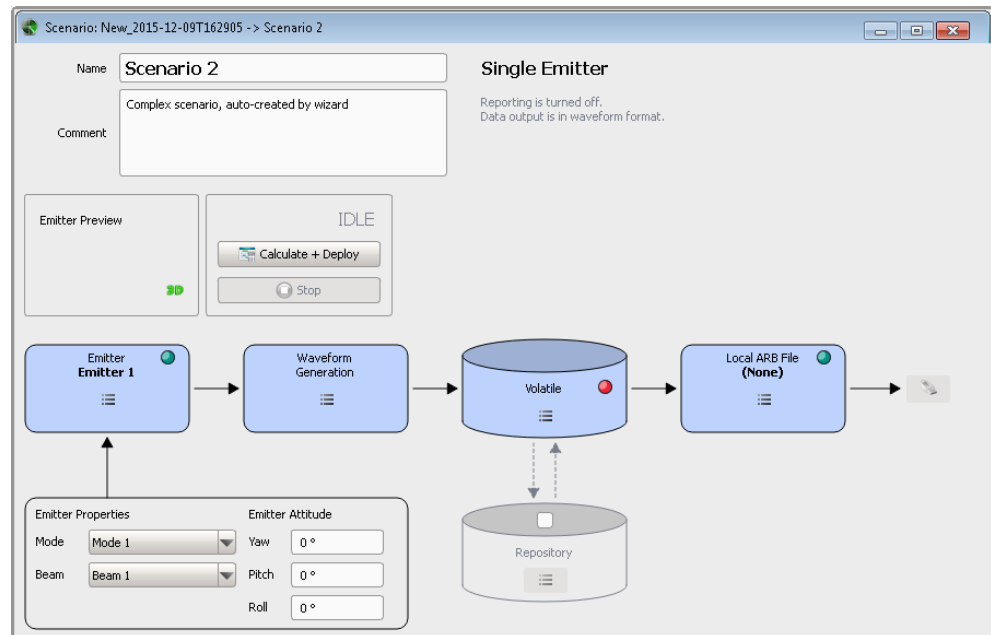
3. Use the mouse wheel to zoom in and out.

- To rotate the 3D view around its origin, left mouse click, keep the key pressed and move the mouse.

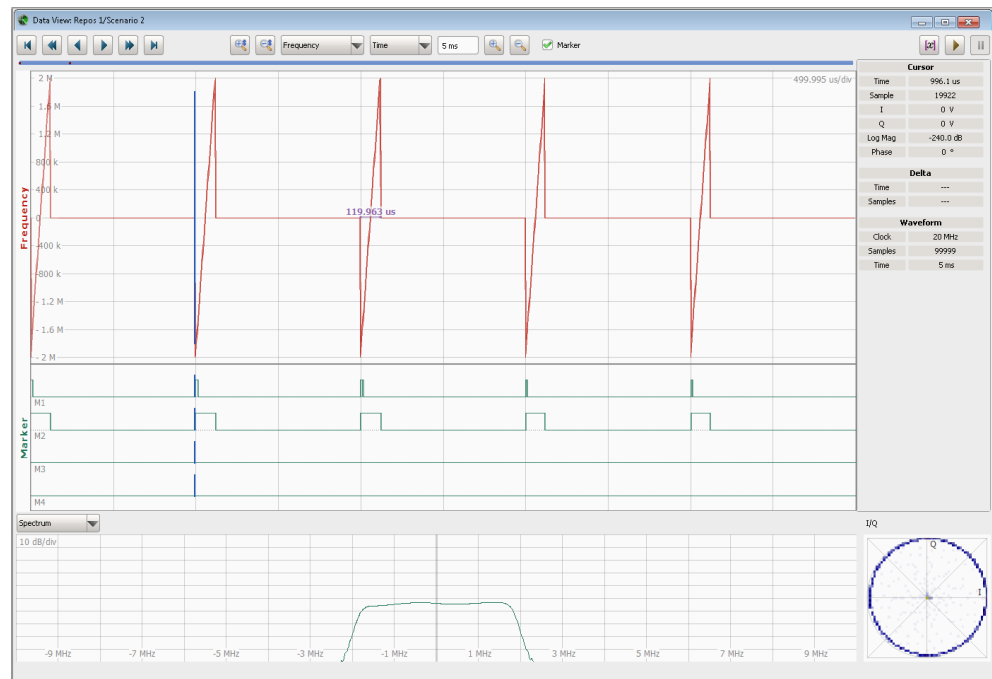
For information, see [Chapter 15, "Creating Complex 2D Scenarios with Receiver, Emitters and Interferers"](#), on page 248

To create and verify the generated ARB waveform

- In the "Scenario" dialog, define the storage location for the ARB files, see [Chapter 3.6.2, "Generating an ARB Waveform File"](#), on page 27.



- Select "Calculate + Deploy" to start signal calculation.
- Select "Volatile > View".
- In the "Waveform View" dialog, select "View Mode > Frequency".
- Use the "Zoom In" button to change the scale on the y axis.



The "Data View" dialog confirms that the generated sequence consists of 5 chirped pulses, each with a PRI = 1 ms. The frequency variation of each pulse has the typical linear chirp shape.

6. Compare the signal spectrum with the spectrum on [Figure 3-3](#).

You can retrieve more information about the waveform, for example observe the "Waterfall" diagram, display a part of the signal in greater details, or enable and display marker traces.

You find details in the following:

- [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264
- [Chapter 20, "Defining and Enabling Marker Signals"](#), on page 301

3.6.6 Transferring the Waveform to and Playing it with a Connected Instrument

The example configurations are theoretical cases, because you will not use the software alone. Usually, the PC with the installed software would be connected via LAN to a vector signal generator.

Transferring of waveforms and playing them by a signal generator is beyond the scope of these first steps.

For description, see [Chapter 17, "Playing the Generated Waveform Files"](#), on page 272.

3.6.7 Saving and Recalling Settings


Repositories are automatically saved when you exist the application. We recommend that you set a meaningful repository name before closing the application.

To rename the repository

Your workspace shows one automatically created repository, "New_yyyy_mm_tttt".

1. In the repository tree, select "Repository > Repos 1".
2. Change the name and optionally enter a comment.
For example, enter "General > Name > K300 and K301 Tests" or a "Comment > Repository with test files".
To enter a new line in the comment field, press SHIFT+ENTER.
3. Close the dialog.

To save the repository

A yellow data base icon  in the repository tree indicates that there are unsaved changes.

- ▶ In the menu bar, select "File > Save Repository > K300 and K301 Tests".
The repository is stored in the data path set during the program installation.

To exit the software

- ▶ Close the application software.
If the current repository contains unsaved changes, you are prompt to store or ignore them.
Note: Unsaved changes are lost.

For more information, see:

- [Chapter 3.7.1, "Customizing Your Workspace"](#), on page 35
- [Chapter 5.4, "How to Manage the Project Data"](#), on page 49

3.6.8 Advanced Features and Examples

With these first steps, you have gained an impression of the provided functionality.

You find a comprehensive description of the full range of capabilities in the following sections:

- [Chapter 7.3, "How to Create a New Pulse and Adjust Its Settings"](#), on page 104
- [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124
- [Chapter 6.3, "How to Select and Create a Test Scenario"](#), on page 68
- [Chapter 14.3, "How to Create and Map Generator Profiles"](#), on page 244

- [Chapter 16.3, "How to Analyze the Content of Waveform Files and Files with I/Q Data"](#), on page 269
- [Chapter 17.5, "How to Create, Transfer and Play the Waveforms of Simple Scenarios"](#), on page 284
- [Chapter 21.2, "How to Create Test Reports"](#), on page 312
- [Chapter 10.4, "How to Create a Library with Antenna Patterns and Scans"](#), on page 186
- [Chapter 11.4, "How to Create and Configure Emitters"](#), on page 205
- [Chapter 15.5, "How to Create Scenarios with Emitters, Interferers and a Receiver"](#), on page 258
- [Chapter 13.2, "How to Create a Waveform Scenario and Work with Waveforms"](#), on page 232
- [Chapter 9.2, "How to Create IPM Profiles and Use Them to Vary Pulse Parameters"](#), on page 150
- [Chapter 20, "Defining and Enabling Marker Signals"](#), on page 301
- [Chapter 18.2, "How to Configure the Bit Stream Used by the MOP"](#), on page 295
- ["How to import and assign user defined plugins"](#) on page 297

3.7 Customizing the Software

You can change default application settings to adapt the user interface to your needs.

- [Customizing Your Workspace](#).....35
- [Changing Colors and Default Configuration](#).....36

3.7.1 Customizing Your Workspace

A workspace comprises the repository tree on the left side with one or more loaded repositories, and the working area with one or more screens. You can create several workspaces with different repositories assigned to them. You can save and load a user-defined workspace and switch between the different workspaces.

To save a workspace

We assume that you have opened or created a repository and opened one or more dialogs in one of the screens.

1. In the menu bar, select "File > Save Workspace/Save Workspace As" or use the save icon in the toolbar.
2. In the explorer, enter a file name, e.g. `MyWorkspace.pswk` and confirm with "Ok".
Per default, the `*.pswk` file is stored in the application data directory.

To load a workspace

1. In the menu bar, select "File > Open Workspace".

Existing default and user-defined workspaces are listed.

2. Select a workspace.

The workspace opens. All dialogs and repositories associated with the workspace are displayed.

To close a workspace

You can close a workspace if you do not need it no longer or you need to switch to a different one.

- ▶ In the menu bar, select "File > Clear Workspace".

The workspace is closed.

Closing the workspace unloads all repositories and closes all dialogs.

You can open a workspace again, see ["To load a workspace"](#) on page 35.

3.7.2 Changing Colors and Default Configuration

You can change the default application settings, for example, change the display colors or change default settings of the R&S Pulse Sequencer.

To customize the display colors

1. In the menu bar, select "Configure > Colors".



1 = Color scheme

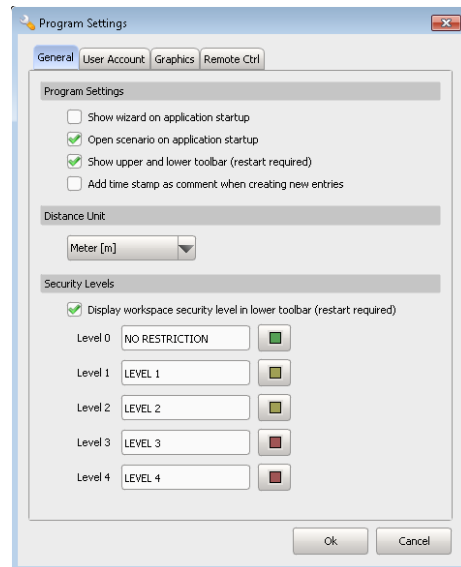
2 = Assigns colors to the different display elements

2. Select a predefined color scheme and if required, adjust the display colors individually.
3. Adjust the display colors of the individual elements, e.g. select "Traces > Magnitude" and change the default color.

4. Select "Save As" to store your custom color profile in a file, e.g. MyColors.col.

To change the general program settings

1. In the menu bar, select "Configure > Settings".

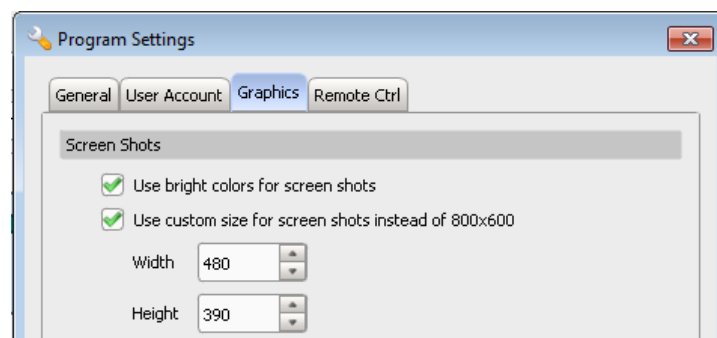


2. Change the settings, for example select the units in that the distance is expressed or select the colors indicating the different security levels.
For information on the security levels, see ["General Repository Settings"](#) on page 43.
3. Confirm with "Ok".

To set the size and color scheme of the hardcopies

For documentation purposes, you can create a hardcopy or screen shot of the current screen or dialog. Hardcopies can be sent to the printer or stored to the clipboard.

1. To adjust the hardcopy size, in the menu bar, select "Configure > Settings > Graphics".



2. Select "Use bright colors for screen shots > On" e.g. for ink saving printing.

Brighter colors are used for all graphical screens instead of the ones set in the current color scheme.

3. Select "Use custom size for screen shots instead of 800x600 > On" and set the "Width" and "Height" in number of pixels.
4. Confirm with "Ok".

The "Program Settings" dialog has two more tabs. The settings provided on these tabs are explained in:

- ["To change the default user used to access repositories"](#) on page 53
- [Chapter 17.3, "Signal Generator Remote Control Settings"](#), on page 280

3.8 Getting Information and Help

If any questions or problems concerning the R&S Pulse Sequencer arise, an extensive online help system is provided in the software and can be consulted at any time.

The integrated help system provides the description of all functions. The help system is context-sensitive and provides information specifically for the current operation or setting to be performed. In addition, general topics provide an overview on complete tasks or function groups as well as background information.

Calling Up Help

- ▶ To display the "Help" dialog for the currently focused screen element, e.g. a setting in an opened dialog, select the "Help" icon on the toolbar or press F1.

The "Help" dialog is displayed. A topic containing information about the focused element is displayed. If no context-specific help topic is available, a more general topic or the "Contents" tab is displayed.

The Help toolbar provides buttons with standard navigation functions. The Help window contains several tabs:

- "View" - shows the selected help topic
- "Contents" - contains a table of help contents
- "Index" - contains index entries to search for help topics
The index is sorted alphabetically. You can browse the list, or search for entries in the list
- "Bookmarks" - contains custom bookmarks
- "Search" - provides text search

To close the Help window

- ▶ Select the "Close" icon in the upper right corner of the help window.

4 About the R&S Pulse Sequencer

The R&S Pulse Sequencer generates complex pulsed signal for use with a signal generator. A typical setup includes a PC with installed software and a connected vector signal generator. See [Figure 4-1](#) for an example of a simplified test setup.

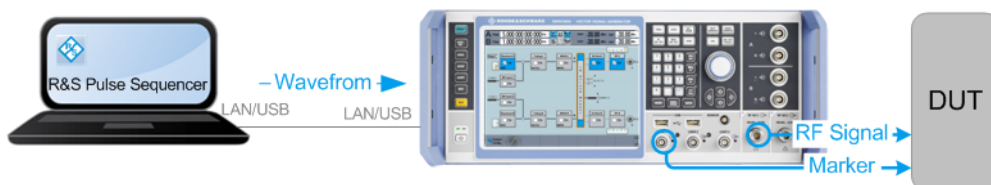


Figure 4-1: Simplified test setup

From the configured settings, like selected emitter and receiver configuration, the R&S Pulse Sequencer verifies the capabilities of the connected instrument. The R&S Pulse Sequencer checks, whether the instruments capabilities fulfill the requirements of the generated waveform. In particular, this affects the bandwidth, the carrier frequency, and the waveform length of the calculated signal. The R&S Pulse Sequencer then calculates a waveform and transfers it to the instrument.

See also [Chapter 17, "Playing the Generated Waveform Files"](#), on page 272.

Introduction to the Software Concept

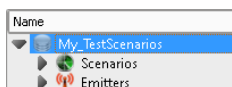
The following is brief introduction to the software concept and the way it handles files and settings.

Repository

The R&S Pulse Sequencer organizes the data in repositories.

A repository is a file-based database located on the local hard drive or on a network storage drive.

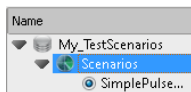
Repositories are entirely managed by the R&S Pulse Sequencer. They can be password protected against unauthorized access. Repositories can be exported as archives and shared among multiple users if they are located on a network drive. Several users can have read permission to load and use the same repository simultaneously. However, only one user can have a write permission at a time.



To move or share repositories with other users, use the "Export/Import Repository Archive" function. Do not directly access the repositories from your windows Explorer.

For more information, see [Chapter 5, "Organizing the Project Data in Repositories"](#), on page 43.

Scenario



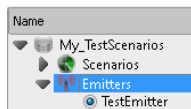
The scenario is the top-level description of the signals to be generated.

The type of the scenario determines which kind of signal is calculated and how this signal is processed. A more complex scenario combines several signals. Scenarios may also contain information about the time variation of the signals.

Starting a scenario creates the I/Q waveform, sends them to the ARBs and configures the signal generators.

For more information, see [Chapter 6, "Selecting a Suitable Scenario and Creating Scenarios"](#), on page 54.

Emitter



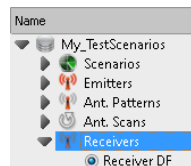
Emitters emulate radar systems. The emitter combines a sequence, an antenna scan, and an antenna pattern to one logical unit.

An emitter can work in more than one operating mode and switch between them. The operating mode is a description of the radar purpose, such as scanning, searching or tracking. For example, an airport surveillance radar can switch between different scan types to observe its airspace.

Each mode can have individual antenna and signal configuration. Multiple beams are possible within a mode.

For more information, see [Chapter 15, "Creating Complex 2D Scenarios with Receiver, Emitters and Interferers"](#), on page 248.

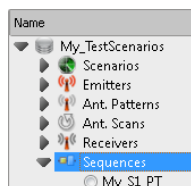
Receiver



Receivers are used to model direction finding systems. They comprise of up to 10 individual antennas with individual patterns and positioning relative to a direction finding system.

For more information, see [Chapter 12, "Emulating Receivers"](#), on page 211.

Sequence

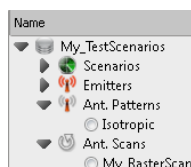


A sequence describes how pulses are arranged to form a waveform. The most simple sequence comprises one pulse that is repeated with a constant PRI (pulse repetition interval). Typical sequences however are rather complex; they may contain pulse definitions, waveforms, and sequencing elements such as repetitions and loops.

The sequence is a logical description. It cannot directly be represented as an I/Q waveform.

For more information, see [Chapter 8, "Building Pulse Sequences"](#), on page 111.

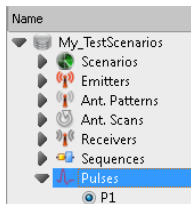
Antenna



The antenna characterizes the radiation pattern of the emitter. Typical radar antennas like the phase array antennas can adapt their radiation pattern dynamically, depending on the operating mode the radar is working in. For example, an antenna can use a cosecant beam in scan mode and pencil beam in track mode.

For more information, see [Chapter 10, "Defining Antenna Patterns and Antenna Scans"](#), on page 161.

Pulse



In the R&S Pulse Sequencer, pulses are the fundamental building block of any signal. Pulses are described mathematically with their amplitude (envelope), the applied modulation on pulse (MOP) and the enabled marker signals. One single pulse description is however not sufficient to calculate (and process) a signal; pulses have to be organized in a sequence, and assigned to an emitter.

For more information, see [Chapter 7, "Creating a Pulse Library"](#), on page 70.

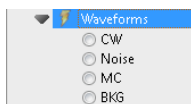
IPM (Inter-Pulse Modulation)



Inter-Pulse Modulation (IPM) profiles modify a pulse parameter (like level, pulse repetition interval, or frequency offset) from one pulse to another one. The IPM profiles are applied within a sequence.

For more information, see [Chapter 9, "Defining and Enabling Inter-Pulse Modulation Effects"](#), on page 134.

Waveforms and Interferers

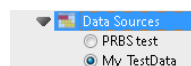


Additionally to creating pulse signals you can also load waveform files and create waveform scenarios with them or even use them as interferers.

An interferer contains complete waveforms, that can be user-defined or created with the signal generation software R&S WinIQSIM2. Interferers can be mixed with the signal.

For more information, see [Chapter 13, "Working with Waveforms and Generating Interfering Signals"](#), on page 219.

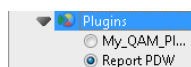
Data Source



Data source defines a numerical source for the symbols used in the modulations applied on the pulse (MOP).

For more information, see [Chapter 18, "Defining the Data Source"](#), on page 291.

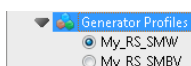
Plugins



A plugin is an extension to the standard functions of the R&S Pulse Sequencer. For example, a plugin can be a user-defined MOP or IPM profile. For even more flexibility, plugins may contain variables. If different values are assigned the same variable in the different pulses, the applied MOP varies.

For more information, see ["How to import and assign user defined plugins"](#) on page 297.

Instruments and Generator Profiles



The generator profile is a logical placeholder for a physical instrument. It describes the minimum requirements that must be fulfilled so that the signal can be generated. Generator profiles are assigned to physical instruments.

For more information, see:

- [Chapter 14, "Creating and Mapping Generator Profiles"](#), on page 237.

- [Chapter 17, "Playing the Generated Waveform Files"](#), on page 272.

5 Organizing the Project Data in Repositories

This section explains the basic repository concept and informs you briefly how the software data base is built.

An introduction to the concept of elements handling is provided in "Creating elements" on page 23.

5.1 Repository Settings

Access:

- ▶ In the project tree, double click on the repository name.

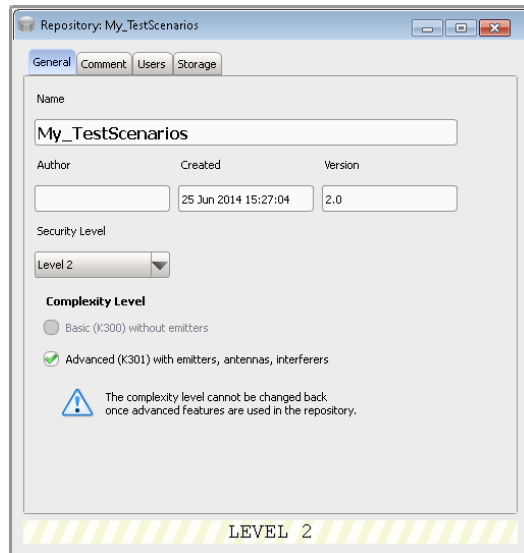
The repository tree shows the content of the currently loaded repositories.

Available settings:

General Repository Settings	43
Comment	44
Users Management	44
Storage	45

General Repository Settings

The "General Settings" tab provides general information, like the name of the author and the date of creation.



- "Name" When a new repository is created, it is named automatically but you can rename it afterwards.
The automatically assigned names follow the syntax:
New_<yyyy-mm-dd><T><hhmmss>, where <yyyy-mm-dd> and T<hhmmss> indicate the current date and time respectively.
- "Security Level" A security level information is assigned to each repository. "Level 0" is the lowest level that means no restrictions and "Level 4" is the highest one. Elements belonging to a repository with higher security level cannot be copied to a repository with lower one.
If more than one repository are currently loaded, the workspace displays the highest required security level, see [Chapter 3.4, "Understanding the Displayed Information"](#), on page 18.
See also ["To change the general program settings"](#) on page 37.
- "Complexity Level" The complexity level defines whether basic or advanced features are used in the repository. The complexity level also reflects the option concept of the signal generators.
For example, if "Complexity Level > Advanced (K301)" is used, the signal generator has to be equipped with the option Advanced Features for Pulse Sequencer, e.g. R&S SMW-K301.
See also:
- [Chapter 2, "Welcome to the R&S Pulse Sequencer"](#), on page 12
 - [Chapter 3.1, "Required Options"](#), on page 14

Remote command:

[REPOSITORY:CATALOG?](#) on page 329

[REPOSITORY:CREATE](#) on page 329

[REPOSITORY:SELECT](#) on page 330

[REPOSITORY:REMOVE](#) on page 332

[REPOSITORY:AUTHOR](#) on page 420

[REPOSITORY:DATE](#) on page 420

[REPOSITORY:VERSION](#) on page 421

[REPOSITORY:COMPLEXITY](#) on page 420

[REPOSITORY:SECURITY](#) on page 421

Comment

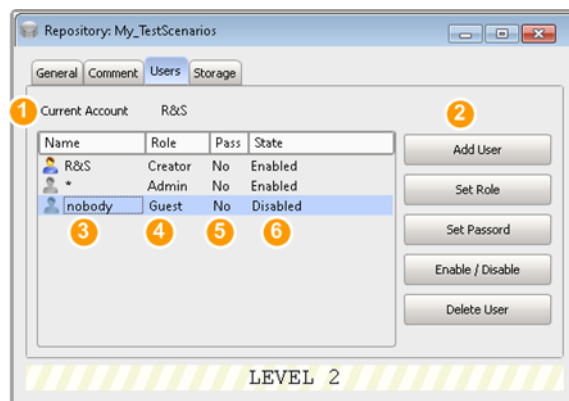
Enter a short description of the repository.

Remote command:

[REPOSITORY:COMMENT](#) on page 331

Users Management

The R&S Pulse Sequencer uses its *own user rights management* that is independent from the user administration of the operating system but follows the same principles: the current user cannot delete its own account; users with lower access rights cannot delete accounts with higher rights.



- 1 = Current user
 2 = Standard functions for user management
 3 = Login name or custom user name
 4 = Assigned user role; the creator role is assigned automatically and cannot be changed (see also [Table 5-1](#))
 5 = Identifies if the repository is password protected or not
 6 = Current user state

Repositories support simultaneous access from several users. Per default, all users are granted with the same (administrator) rights. The user rights can be restricted by assigning of a different user role, other than the creator role. At least one user with administrator rights must always exist.

To protect a repository from accidental changes, the repository data can be password protected. Passwords are then required to access a repository and any loss of the password would render the repository inaccessible.

Table 5-1: Overview of the available user roles and their access rights

User role	Read access	Write access	Create a scenario form the existing elements	Delete elements	Copy repository elements
Creator	x	x	x	x	x
Admin	x	x	x	x	x
User	x	-	x	-	x
Guest	x	-	-	-	x

Per default, the R&S Pulse Sequencer uses your Windows login name to access the repositories but you can also use a custom name.

See also:

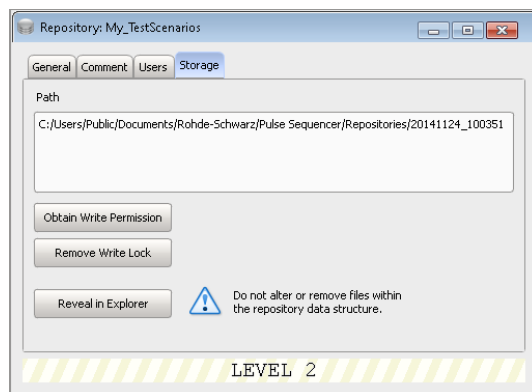
- ["To protect a repository from accidental changes"](#) on page 52
- ["To change the default user used to access repositories"](#) on page 53

Remote command:

[REPOSITORY:ACCESS?](#) on page 419

Storage

This tab provides settings necessary to configure the default storage folder for the repositories.



"Path" Displays the location (network or local folder) the repository is stored in.

Remote command:

[REpository:PATH?](#) on page 421

"Obtain Write Permission"


Repositories allow the simultaneous access from several users. If a user with write permission has opened a repository, this write permission is exclusive. Another user will be granted with the write permission only after the first user had closed the repository or removed it from its workspace.

You can request an exclusive write permission only on a repository that is not opened by another user with write permission.

See also:

- ["Discovered Repositories on the Mass Storage"](#) on page 48
- ["To obtain write permission on a repository"](#) on page 52

"Remove Write Lock"

The R&S Pulse Sequencer locks an improperly or unexpectedly closed repository and indicates this situation with a red lock symbol  in the *"Repository Tree"*. A locked repository can be unlocked only from the same PC and by the same user with write permission that had opened the repository before.

See ["To remove write lock that results from a previously crashed session"](#) on page 479

"Reveal in Explorer"

You can open the repository file structure in the Windows Explorer.

Note: Do not change the repository content by changing the file structure in the windows Explorer!

To move or share repositories with other users, always use the "Export/Import Repository Archive" function.

See:

- ["To create and export a repository archive"](#) on page 51
- ["To import a repository archive"](#) on page 51

Remote command:

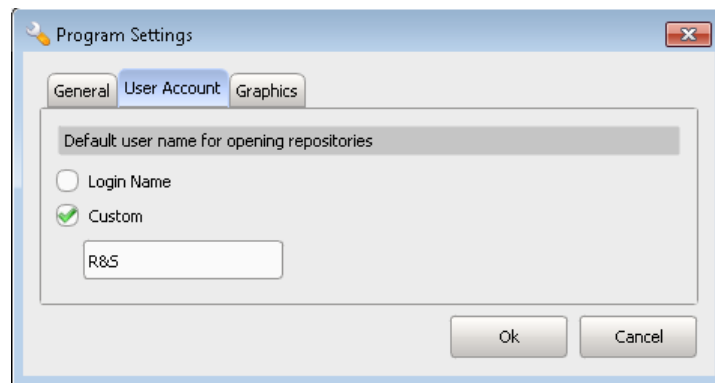
[REpository:FILENAME?](#) on page 420

[REpository:SAVE](#) on page 421

5.2 User Account Settings

Access:

1. In the menu bar, select "Configure > Settings".
2. Select "User Account".



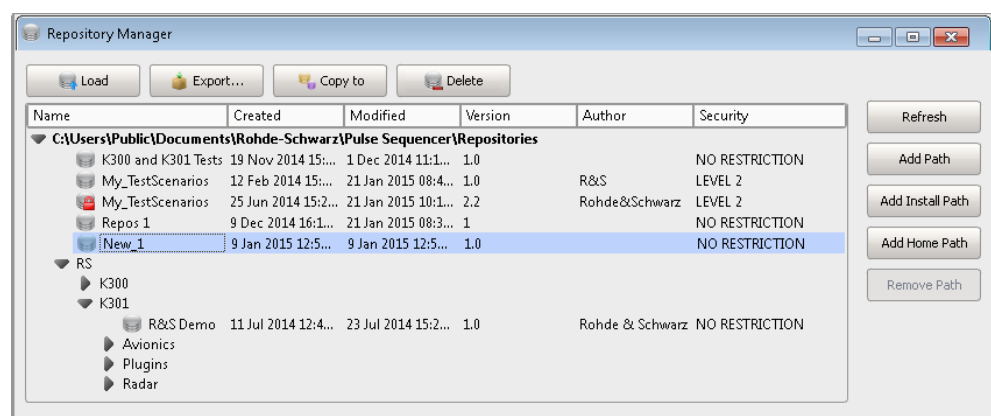
User Account

Set the user name for opening the repositories. Per default, the user's login name is used, but you can also define and use a custom user name.

5.3 Repository Manager Settings

Access:

- ▶ In the menu bar, select "File > Load and Manage Repository".



The "Repository Manager" comprises the setting for managing the repository files and archives on the mass storage.


Available settings:

Discovered Repositories on the Mass Storage.....	48
Load.....	48
Export.....	48
Copy to.....	48
Delete.....	49
Refresh.....	49
Add Path, Add Install Path, Add Home Path.....	49
Remove Path.....	49

Discovered Repositories on the Mass Storage

Lists all found directories and the repository files they contain.

You can change the column width and order.

A red lock symbol  on the data base icon in the "Repository Manager" indicates one of the following situations:

- Repository is opened by another user with explicit write access
- Insufficient user right for the particular repository
- Insufficient user right on the file system (directory) the repository is stored in.

Try out to:

- Wait until the repository has been closed and try to obtain exclusive write permission, see ["To obtain write permission on a repository"](#) on page 52.
- Change your current user or use a user with different role, see ["To change the default user used to access repositories"](#) on page 53
- Log in as user with sufficient (write) access rights on the file system.

Remote command:

`REPManager:CATalog?` on page 422

Load

Loads and displays the selected repository in the workspace.

See ["To load a repository"](#) on page 51.

Remote command:

`REPManager:LOAD` on page 422

Export

Exports the selected repository to an archive file.

See ["To create and export a repository archive"](#) on page 51.

Remote command:

`REPManager:EXPort` on page 423

Copy to

Copies the selected repository to a new location.

The R&S Pulse Sequencer automatically copies all relevant files and data base elements.

Delete

Deletes the entire repository from the permanent mass storage, if:

- The current user has write permission, see [Table 5-1](#)
- The repository is not opened by another user with write permission

Remote command:

[REPManager:DELeTe](#) on page 423

Refresh

Refreshes the list of found repository files.

Add Path, Add Install Path, Add Home Path

Opens the "Find Path" dialog that is similar to the Windows Explorer.

Navigate to the required directory and load it. Found repository files are displayed in the list.

Remote command:

[REPManager:PATH:ADD](#) on page 424

[REPManager:CATalog?](#) on page 422

Remove Path

Removes the selected path. The repository files within this directory are also removed from the list.

Remote command:

[REPManager:PATH:DELeTe](#) on page 424

5.4 How to Manage the Project Data

See:

- ["To create a repository"](#) on page 49
- ["To rename the repository"](#) on page 34
- ["To save the repository"](#) on page 34
- ["To delete a repository"](#) on page 50
- ["To remove a repository from the workspace"](#) on page 50
- ["To load a repository"](#) on page 51
- ["To create and export a repository archive"](#) on page 51
- ["To import a repository archive"](#) on page 51
- ["To share a repository with other users"](#) on page 51
- ["To obtain write permission on a repository"](#) on page 52
- ["To protect a repository from accidental changes"](#) on page 52
- ["To remove the password protection on a repository"](#) on page 53

To create a repository

1. In the menu bar, select "File > New Repository".

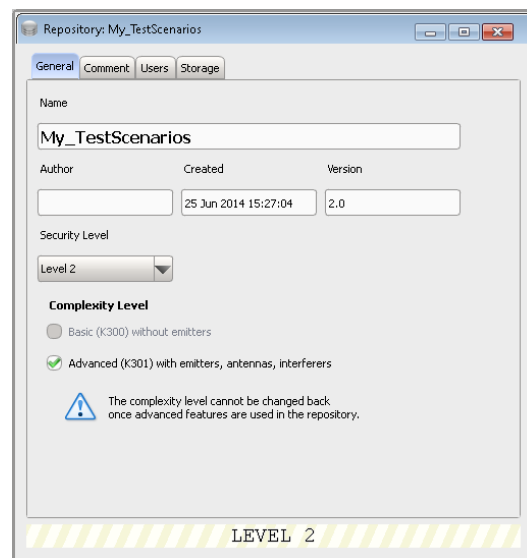
A new empty repository is created and added to the project tree.

Repositories are named automatically. The repository names follow the syntax:

New_<yyyy-mm-dd><T><hhmmss>, where <yyyy-mm-dd> and T<hhmmss> indicate the current date and time respectively.

- In the project tree, double click on the repository name.

The "Repository" dialog opens.



- Change the settings, e.g. change the automatically assigned "Name" or add a "Comment".
- Set the "Complexity Level > Advanced (K301)".
- Close the dialog.

To delete a repository

You can delete repositories that you no longer need, if this repository is not opened by another user with write permission.

- Use a user with write access rights, e.g. Admin or Creator (see [Table 5-1](#)).
- In the menu bar, select "File > Load and Manage Repository".
- From the listed files, select a repository file, e.g. "My_TestScenarios".
- Select "Discard".

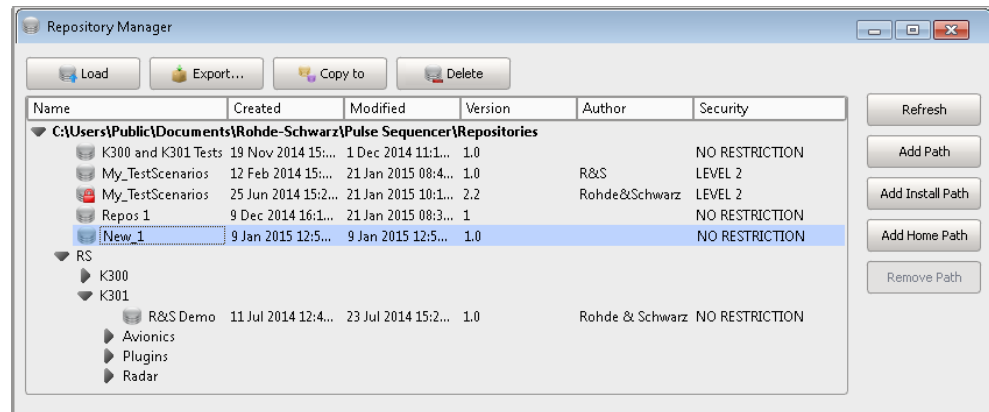
To remove a repository from the workspace

- ▶ In the project tree, select the repository e.g. "My_TestScenarios", open the context menu, and select "Remove from Workspace".

The repository is removed from the workspace but not deleted. You can open it again, see ["To load a repository"](#) on page 51.

To load a repository

1. In the menu bar, select "File > Load and Manage Repository".
2. If necessary, for example by loading and exchanging repositories with another user, change the directory the repository files are searched in.
3. From the listed files, select a repository file, e.g. "New_1".



4. Confirm with "Load".

The repository is opened and displayed in the project tree.

To create and export a repository archive

We assume that you have created a repository and configured at least one scenario.

1. In the menu bar, select "File > Load and Manage Repository".
2. If necessary, change the directory repository archive files are stored in.
3. From the listed files, select a repository file, e.g. "My_TestScenarios".
4. Select "Export".

The repository archive is stored in a file with extension *.psarch.

To import a repository archive

1. In the menu bar, select "File > Import Repository".
2. Navigate to the network directory and select a *.psarch file.

The "Load and Manage Repository" dialog opens.

3. Select a repository file and confirm with "Load".

The repository is opened and displayed in the project tree.

To share a repository with other users

To move or share repositories, use the "Export/Import Repository Archive" function. Do not access the repositories from your Windows Explorer.

1. See ["To create and export a repository archive"](#) on page 51

2. See ["To import a repository archive"](#) on page 51

To obtain write permission on a repository

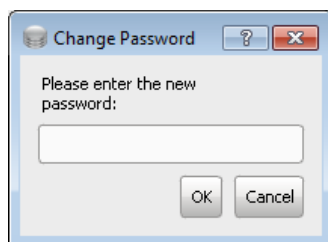
Repositories can be opened with executive write permission by one user at a time.

If you have sufficient user rights, you can request an exclusive write permission on a repository that is not opened by another user with write permission.

1. In the project tree, double click on the repository name.
2. In the "Repository" dialog, select "Storage".
3. Select "Obtain Write Permission".

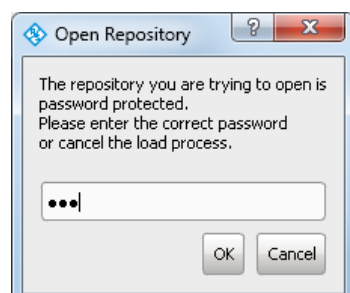
To protect a repository from accidental changes

1. In the project tree, double click on the repository name.
2. In the "Repository" dialog, select "Users".
3. To protect the repository with a password:
 - a) Select "Users > <current Account>" and select "Set Password".
The "Change Password" dialog opens.



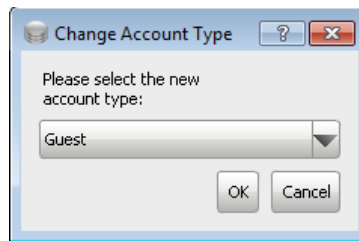
- b) Enter a password and confirm with "Ok".

On a new startup or when this repository is loaded, the R&S Pulse Sequencer requires the correct password to load the repository.



4. To allow read-only access:

- a) Select "Users > *" and select "Set Role".
The "Change Account Type" dialog opens.



- b) Select a new role with read-only access, e.g. "Guest" and confirm with "Ok".

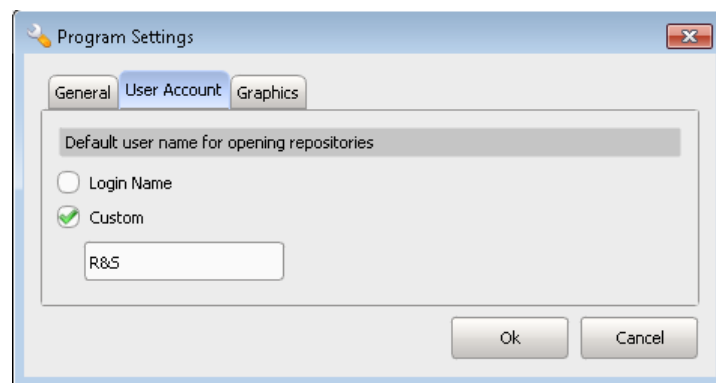
To remove the password protection on a repository

1. In the project tree, double click on the repository name.
2. In the "Repository" dialog, select "Users".
3. Select "Users > <current Account>" and select "Set Password".
4. In the "Change Password" dialog, leave the password field blank.
5. Confirm with "Ok".

The "Users" dialog confirms that the repository is not protected with password.

To change the default user used to access repositories

1. In the toolbar, select "Configure > Settings > User Account".



2. Select "Custom > On" and enter an existing user name.

6 Selecting a Suitable Scenario and Creating Scenarios

This section explains the provided scenario types and helps you select the scenario most fitting to your current task.

6.1 Overview of the Available Scenarios and Their Complexity

The R&S Pulse Sequencer supports six scenarios with different complexity. Select the scenario most fitting your test case:

- **Single Sequence**
The Single Sequence scenario is used to create sequences from individual pulses. Pulse parameters include the pulse envelope, modulation on pulse (MOP), carrier frequency, and level. The sequence editor defines the order in which the pulses are generated and also adds control elements such as loops and fillers. Inter-pulse modulation profiles can be applied to vary pulse parameters.
- **Sequences (Collection)**
The Sequences (Collection) scenario is used to create multiple sequences which can be selected in arbitrary order. Pulse parameters include the pulse envelope, modulation on pulse (MOP), carrier frequency, and level. The sequence editor defines the order in which the pulses are generated and also adds control elements such as loops and fillers. Inter-pulse modulation profiles can be applied to vary pulse parameters.
- **Waveform Sequence**
The Waveform Sequence combines individual waveforms to one joint waveform or to one multi-segment waveform. The sequence editor defines the order of the individual waveforms and adds control structures such as loops and fillers.
- **Single Emitter**
The Emitter scenario combines a pulse train with an antenna pattern and a particular antenna scan type. An emitter comprises several modes. Each mode is defined by an antenna pattern and an antenna scan, and multiple beams that contain the pulse train.
- **Emitters (Collection)**
The Emitter Collection scenario provides a choice of multiple emitters that can be selected manually. An emitter comprises several modes. Each mode is defined by an antenna pattern and an antenna scan, and multiple beams that contain the pulse train.
- **Localized Emitters**
The Localized Emitters scenario combines multiple emitters and one receiver on a 2D map. An emitter comprises several modes. Each mode is defined by an antenna pattern and an antenna scan, and multiple beams that contain the pulse train. The receiver is defined by a single antenna pattern and scan.

- **Direction Finding**
The Direction Finding scenario simulates multiple emitters and a single receiver with up to 10 antennas on a 2D map.
The individual signals for each receive antenna are simulated to enable the receiver to locate the position of a signal source.

6.2 Scenario Settings

Access:

- ▶ Select "Repository Tree > Scenario".

The "Scenario" dialog provides the settings, necessary to configure the signal processing, that is to select the signal source, to create the ARB waveform, to select the generator profile, and to transfer the waveform. The dialog provides also basic instrument control functions that are required for waveform playback and signal routing.

The dialog also shows an interactive block diagram of the main processing blocks, together with their current settings and state.

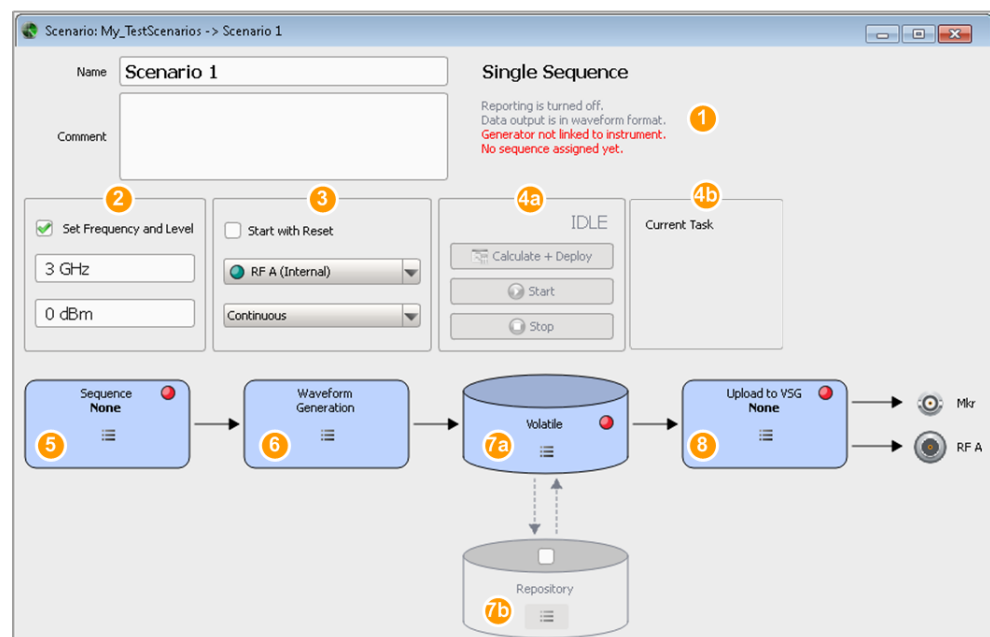


Figure 6-1: Scenario dialog: Understanding the displayed information

- | | |
|---------|---|
| 1 | = Status information : report generation and kind of the generated waveform file |
| 2 | = Major RF parameters |
| 3 | = Functions for basic instrument control |
| 4a | = Calculate + Deploy builds the ARB waveform and transfers it to the instrument |
| 4b | = Waveform generation status, displayed during waveform calculation |
| 5+6+7+8 | = Block diagram of the processing chain |
| 5 | = Represents the signal source the ARB waveform file is built up from, i.e. single or multiple pulse sequences, a waveform sequence or single or multiple emitters |
| 6 | = Accesses settings that influence the Waveform Generation |

- 7a, 7b = [Volatile, Repository](#) represent the temporary memory and the network storage the generated ARB waveform is stored in
- 8 = Displays the currently selected generator's profile ([Upload to VSG \(Vector Signal Generator\)](#)) or indicates that a local ARB waveform is generated ([Load ARB File](#))
- Green LED = Indicates suitable source and instrument profile or stored local waveform
- Yellow LED = Indicates insufficient instruments capabilities
- Red LED = Indicates that there is no source or instrument's profile or no waveform assigned

The provided settings depend on the selected scenario type.

6.2.1 Common Scenario Settings

The following settings are common to all scenarios:

Scenario Name	56
Comment	56
Scenario Type	56
Status information	57
Set RF frequency and level on target instrument	57
Frequency	57
Ref. Level	57
Start with Reset	58
Path (RF output connector)	58
Playback mode	58
Calculate + Deploy, Start/Stop, Busy/Idle	59
Waveform Generation	59
Volatile, Repository	59
Upload to VSG (Vector Signal Generator)	60
Load ARB File	62
New, Edit, Clear, Select, Config, View	63

Scenario Name

Enter a unique name to identify for your current scenario in the repository.

Remote command:

- [SCENario:CREate](#) on page 329
- [SCENario:CATalog?](#) on page 329
- [SCENario:NAME](#) on page 331
- [SCENario:SElect](#) on page 330
- [SCENario:REMove](#) on page 331

Comment

Enter a short description of the scenario. The first line is displayed in the repository tree.

Remote command:

- [SCENario:COMMent](#) on page 331

Scenario Type

Displays the scenario type.

Remote command:

[SCENario:TYPE](#) on page 430

Status information

Displays information on:

- Kind of generated waveform file, single segment or [MSW](#)
- Report generation status, type and file location.
To enable report generation, select "Waveform Generation > Reporting > Reporting > Enable"
See also [Chapter 21, "Creating Reports and Documenting Measurement Results"](#), on page 308.

Set RF frequency and level on target instrument

If enabled, the RF frequency and output level on the signal generator is automatically configured.

Remote command:

[SCENario:OUTPut:RF:ENABle](#) on page 444

Frequency

Sets the carrier frequency of the instrument on which the generated waveform is modulated on.

Emitters use their current operating frequency, see "[EIRP](#)" on page 199.

Remote command:

[SCENario:OUTPut:FREQuency](#) on page 443

Ref. Level

Sets the reference level.

This value is used by the calculation of the pulse envelope. The reference level corresponds to the "Ref." line, displayed on the "Pulse Envelope Graph" as a relative ratio. For example, "Ref. = 1" corresponds to $100\% * \text{"Ref. Level"}$.

For unattenuated pulses, the "Ref. Level" sets the pulse top power.

Example: Pulse top power of an unattenuated pulse

If:

- "Ref. Level = -30 dBm"
- "Pulse > Level > Attenuation > Top Power = 0 dB"

Then the pulse top power would be -30 dB (at the RF output).

Example: Pulse top power of an attenuated pulse

If:

- "Ref. Level = -30 dBm"
- "Pulse > Level > Attenuation Top Power = 10 dB"

Then the pulse top power would be -40 dB (at the RF output).

If:

- "Ref. Level = -30 dBm"
- "Pulse > Level > Attenuation > Top Power = 0 dB"
- Δ Level = 10 dB

Then the pulse top power would be -20 dB (at the RF output).

Remote command:

[SCENario:OUTPut:LEVel](#) on page 443**Start with Reset**

Sends an initial reset command (*RST) to the connected instrument to set this instrument to a definite state.

Remote command:

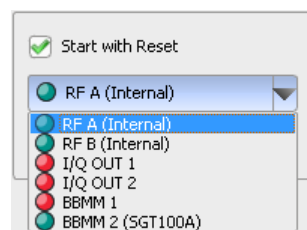
[SCENario:OUTPut:RESet:ENABLe](#) on page 444**Path (RF output connector)**

Selects the signal path that plays the generated signal.

The available values depend on the connected instrument. Listed are all available output connectors, where the green LEDs indicate the active outputs.

Example:Let the test setup use a two-path R&S SMW with the connected R&S SGT, see [Figure 14-2](#) and [Figure 14-3](#).

The generated signal can be routed to both RF outputs and to the BBMM1 connector to that the R&S SGT is connected.



Remote command:

[SCENario:GENerator:PATH](#) on page 433**Playback mode**

Defines the way the generated sequence is processed: played back once ("Single Shot") or looped ("Continuous").

Remote command:

[SCENario:OUTPut:RUNMode](#) on page 444

Calculate + Deploy, Start/Stop, Busy/Idle

Standard control function.

"Calculate + Deploy"	<p>Starts signal calculation, if the prerequisites are fulfilled.</p> <p>The software informs you about the estimated file size. A red "Busy" indication and a detailed progress information indicate that the calculation is in progress.</p> <p>If a signal generator is selected, the waveform is transferred and stored in its volatile memory. Signal generation is not started.</p>
"Start"	<p>If a signal generator is selected, loads the waveform in the ARB and starts signal generation.</p> <p>Signal routing and RF settings are configured as defined with:</p> <ul style="list-style-type: none"> • Set RF frequency and level on target instrument, Frequency, and Ref. Level • Start with Reset, Path (RF output connector), and Playback mode

Remote command:

[SCENario:CALCulate](#) on page 431

[SCENario:START](#) on page 431

[SCENario:STOP](#) on page 431

[SYSTEM:PROGress?](#) on page 456

[SCENario:STATe?](#) on page 431

Waveform Generation

In the general case, the software calculates a waveform, most suitable for the current signal generator; e.g. an ARB file or a multi-segment waveform.

"Config..."	<p>Accesses a dialog, where you can change the settings influencing the waveform generation, see Chapter 17.1, "Waveform Generation Settings", on page 272.</p>
"Open Report Folder"	<p>Opens the report folder in the Windows Explorer, see Chapter 21, "Creating Reports and Documenting Measurement Results", on page 308.</p>

Volatile, Repository

Represents the volatile (temporary) memory and the repository.

Depending on your user rights, you have read-only or full access to the repository. Users granted with full access rights are allowed to perform:

"Volatile > View"

Visualizes the generated waveform file from the volatile memory. See [Chapter 16, "Visualizing and Analyzing Waveforms"](#), on page 264.

Remote command:

[SCENario:VOLatile:VIEW](#) on page 445

"Repository > On"

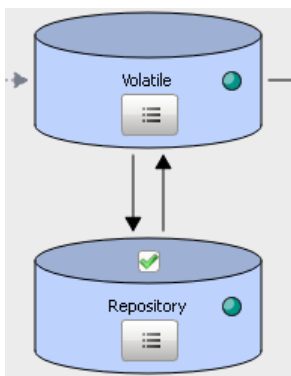
Enables file storage in the repository.

Remote command:

[SCENario:CACHe:REPository:ENABle](#) on page 431

"Volatile > Release to Repository"

Exports and stores the waveform files in the repository, i.e. the waveform is archived in the repository. You can access archived waveforms any time latter, e.g. after software restart.



Remote command:

[SCENario:CACHe:VOLatile:RELease](#) on page 432

"Volatile > Load from Repository"

Loads and play files from this storage; stored waveform files are played back without being recalculated.

Remote command:

[SCENario:CACHe:VOLatile:REStore](#) on page 432

"Volatile/Repository > Clear"

Deletes files from the volatile memory and the repository.

Remote command:

[SCENario:CACHe:VOLatile:CLEar](#) on page 432

[SCENario:CACHe:REPository:CLEar](#) on page 432

Remote command:

[SCENario:CACHe:VOLatile:VALid?](#) on page 432

[SCENario:CACHe:REPository:VALid?](#) on page 432

Upload to VSG (Vector Signal Generator)

Select "Upload to VSG/Local ARB File" > "**Target > Generator**" to transfer and upload the waveform to a signal generator. The second line indicates the selected generator profile.



See also:

- ["To retrieve the profile of the connected generator"](#) on page 246
- ["To find connected instruments and assign the generator profiles to them"](#) on page 245
- ["To assign a connected generator to a scenario"](#) on page 247
- ["To map the generated signals to the signal generator"](#) on page 263
- ["To delete waveforms from the hard disk of the signal generator"](#) on page 286

Following functions are available in the context menu:

- "Select" Assigns an existing generator profile to the scenario.
- Profiles must match with the capabilities of a physical instrument.
- **Green LED:** indicates that the selected profile is suitable for the current scenario;
If all other prerequisites are fulfilled, the "Start" function is active.
 - **Yellow LED:** indicates insufficient capabilities
 - **Red LED:** indicates that no profile is assigned
The status message "Check generator mapping" is displayed.

Remote command:

[SCENario:OUTPut:TARGet](#) on page 444

[SCENario:GENerator](#) on page 433

[SCENario:GENerator:PATH](#) on page 433

"Clear, New, Edit"

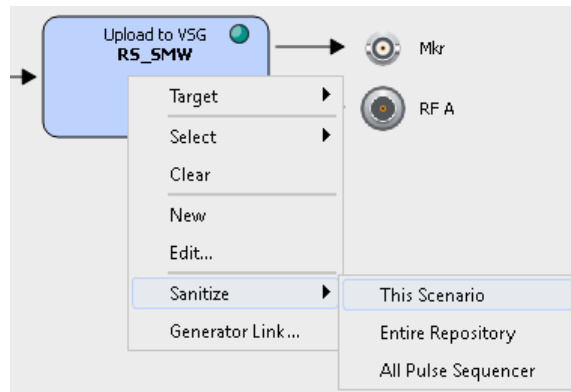
Standard editing functions.

Remote command:

[SCENario:GENerator:CLEar](#) on page 335

"Sanitize"

Removes uploaded waveforms from the hard disk of the signal generator.

**Remote command:**

[SCENario:SANitize](#) on page 445

"Generator Link"

Accesses the "Profile to Instrument Links" dialog, see [Chapter 14.1, "Profile to Instrument Links Settings"](#), on page 237.

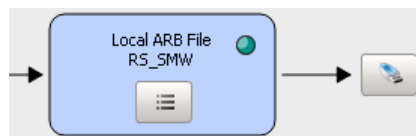
"Assign Emitters"

In a "Scenario Type > Localized Emitters", accesses the dialog for mapping of signals to generator profiles.

See [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275.

Load ARB File

Select "Upload to VSG/Local ARB File" > **"Target > File"** to generate an ARB file instead of transferring the file directly to the signal generator.



Following functions are available in the context menu:

"Set Path" Defines the storage place for the generated ARB file.

"Open Path in Explorer"

Accesses the storage folder in the Windows Explorer.

"Generator Link"

Accesses the "Profile to Instrument Links" dialog, see [Chapter 14.1, "Profile to Instrument Links Settings"](#), on page 237.

"Assign Emitters"

In a "Scenario Type > Localized Emitters", accesses the dialog for mapping of signals to generator profiles.

See [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275.

Remote command:

SCENario:OUTPut:PATH on page 443

New, Edit, Clear, Select, Config, View

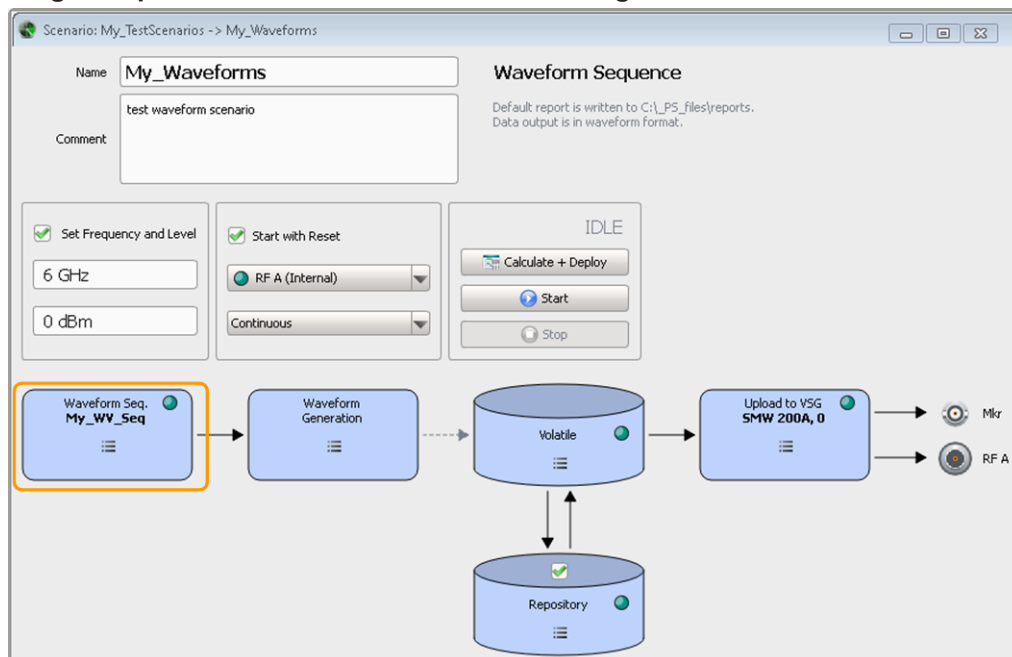
Standard functions in the context-sensitive menu.

See [Table 3-4](#).

6.2.2 Scenario Specific Settings

[Single Sequence and Waveform Scenario Settings](#)..... 63
[Sequence \(Collection\) Scenario Settings](#)..... 64
[Single Emitter Scenario Settings](#)..... 65
[Emitters \(Collection\) Scenario Settings](#)..... 65
[Localized Emitters Scenario Settings](#)..... 67
[Direction Finding Scenario Settings](#)..... 67

Single Sequence and Waveform Scenario Settings



(for an example of a single sequence scenario, see [Figure 6-1](#))

These scenarios generate an ARB signal form a selected sequence or a waveform.

Use the context menu in the "Sequence/Waveform" block to select and configure the transmitted sequence or waveform.

See:

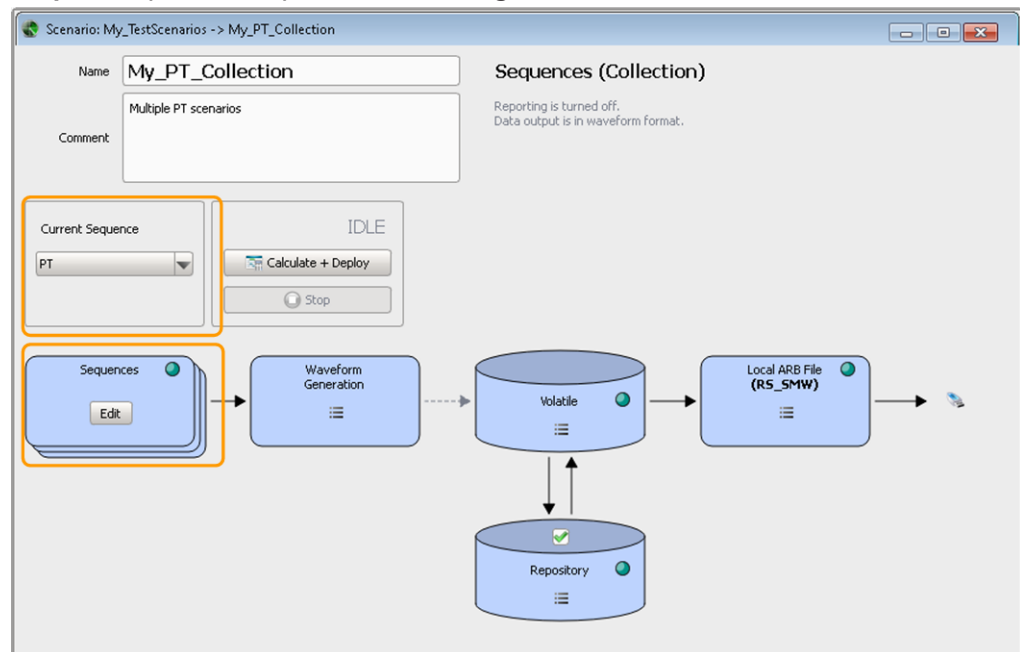
- [Chapter 8, "Building Pulse Sequences"](#), on page 111
- [Chapter 13, "Working with Waveforms and Generating Interfering Signals"](#), on page 219
- [Chapter 17.5, "How to Create, Transfer and Play the Waveforms of Simple Scenarios"](#), on page 284
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56

Remote command:

[SCENario:SEquence](#) on page 433

[SCENario:SEquence:CLEar](#) on page 433

Sequence (Collection) Scenario Settings



A scenario with multiple sequences requires a list of sequences. You can switch between these sequences and select the one to be transmitted.

The "Current Sequence" indicates the alias name of the currently selected one.

See:

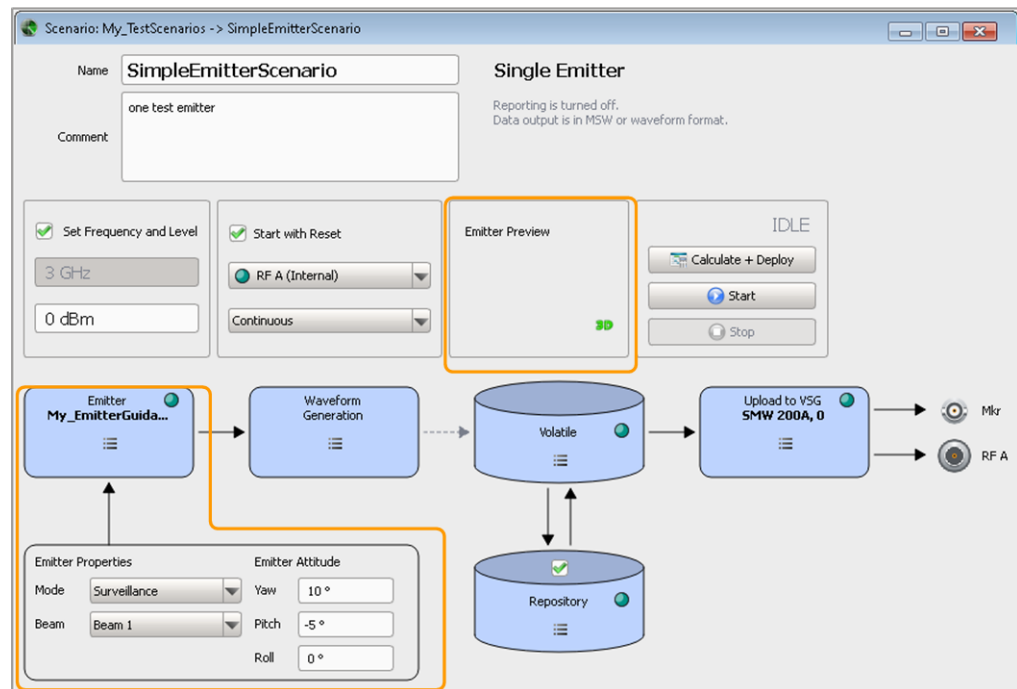
- [Chapter 8.2.6, "Lists with Multiple Sequences"](#), on page 123
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56

Remote command:

[SCENario:CSEquence](#) on page 441

[SCENario:CSEquence:CURRent](#) on page 434

Single Emitter Scenario Settings



Use the context menu in the "Emitter" block to select and configure the emitter.

You can switch between the emitter modes, define the current beam, set the attitude, and visualize the signal received by a static receiver on a 3D preview.

In this scenario, the "Frequency" is a read-only parameter, that displays the [Frequency](#) value of the "Current Emitter" (as selected in the "Emitter" dialog).

See:

- ["Emitter attitude"](#) on page 196
- [Chapter 11.4, "How to Create and Configure Emitters"](#), on page 205
- [Chapter 11.1, "Emitter Settings"](#), on page 198
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56

Remote command:

[SCENario:EMITter](#) on page 435

[SCENario:EMITter:MODE](#) on page 435

[SCENario:EMITter:MODE:BEAM](#) on page 436

[SCENario:EMITter:CLEar](#) on page 335

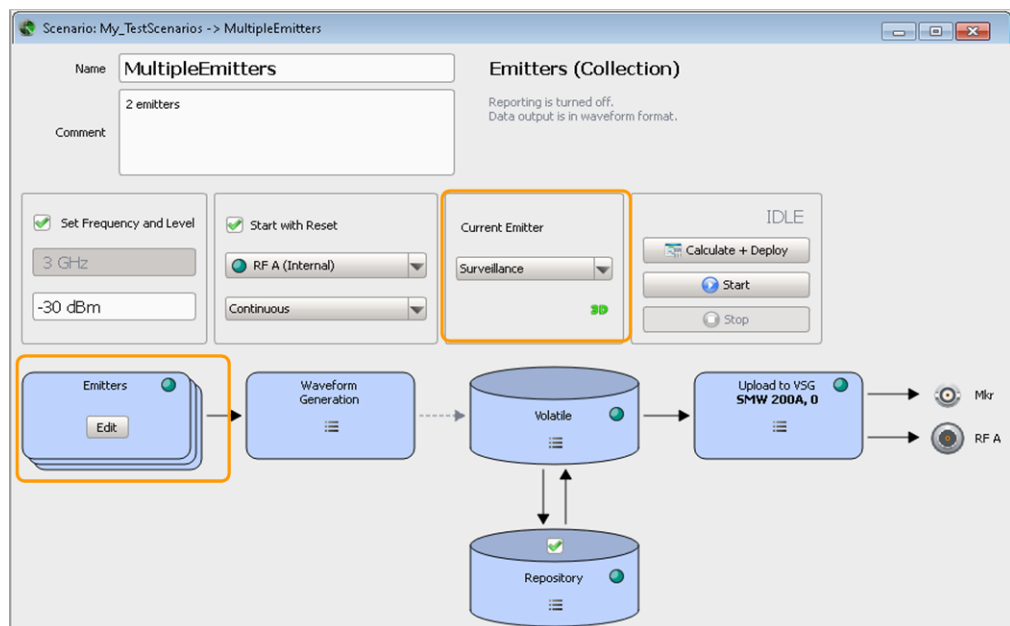
[SCENario:EMITter:DIRection:PITCh](#) on page 434

[SCENario:EMITter:DIRection:YAW](#) on page 434

[SCENario:EMITter:DIRection:ROLL](#) on page 435

Emitters (Collection) Scenario Settings

You can use "Emitter (Collection)" scenario test the receiver's ability to detect the signal from different static emitters. In this dedicated scenario, you can configure several emitters and switch between them sequentially. You can also configure the receiver and change its position in the scanning beam of the emitters.



A scenario with several emitters requires a list of emitters. You can switch between these emitters and select which one is transmitting.

The "Current Emitter" parameter indicates the alias name of the currently selected one. In this scenario, the "Frequency" is a read-only parameter, that displays the [Frequency](#) value of the "Current Emitter" (as selected in the "Emitter" dialog).

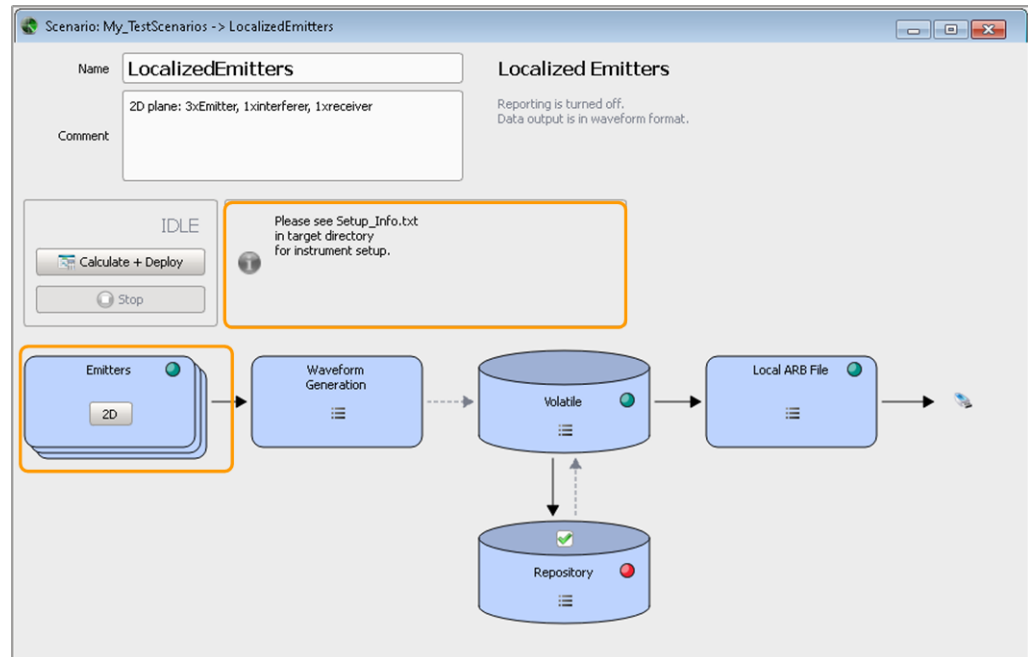
See:

- ["To configure complex scenarios with several emitters"](#) on page 208
- [Chapter 11.3, "Lists with Multiple Emitters"](#), on page 203
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56

Remote command:

[SCENario:CEMit:CURRent](#) on page 434

Localized Emitters Scenario Settings



You can use "Localized Emitter" scenario to test the receiver's ability to detect the sum signal from different static emitters and interferers. In this dedicated scenario, you configure the signal of one or more emitters and optional one or more interferers that would be received by a receiver with defined characteristics. You can also configure the receiver and change its position in the scanning beam of the emitters.

Select "Emitters > 2D" to display the 2D view of the receiver and currently configured emitters and interferers, together with their main characteristics.

See:

- [Chapter 15.5, "How to Create Scenarios with Emitters, Interferers and a Receiver"](#), on page 258
- [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56

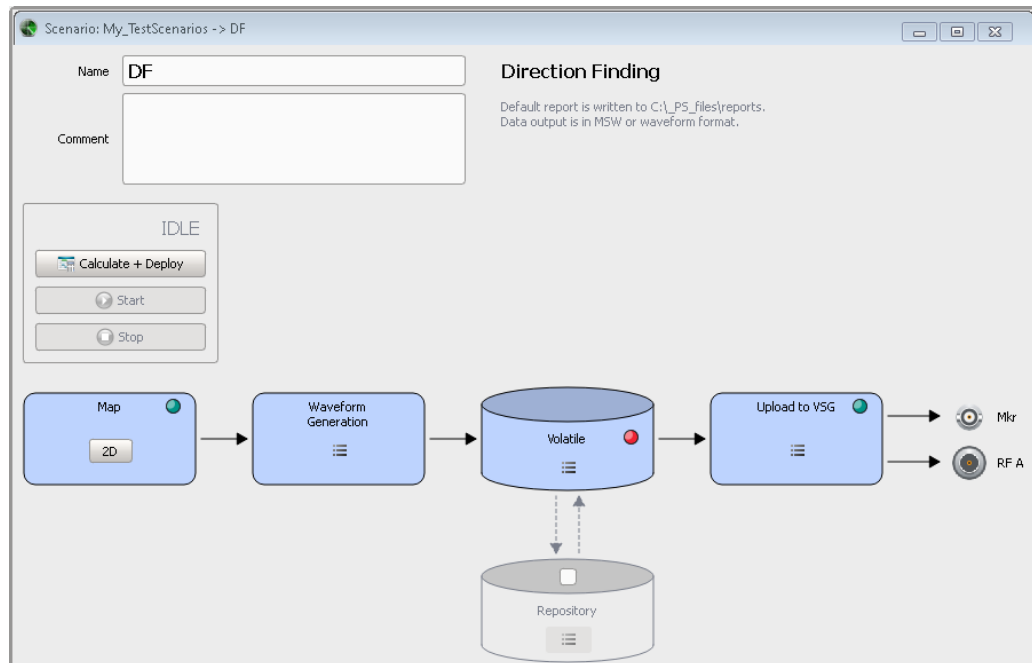
When you create ARB files ("Local ARB File > Target > File"), select "Local ARB File > Open Path in Explorer" to access the generated waveforms and the `Setup_Info.txt` file. Follow the instructions in this `Setup_Info.txt` file.

See also ["To create, transfer and play the waveforms of 2D scenario manually"](#) on page 289.

Direction Finding Scenario Settings

The "Direction Finding" scenario simulates multiple emitters and a single receiver with up to 10 antennas on a 2D map. This scenario is suitable to test direction finding systems and to detect interfering signals or transmissions in not authorized bands.

The generated waveforms are the individual signals of each receive antenna.



Select "Map > 2D" to display the 2D view of the receiver and the configured emitters and interferes, together with their main characteristics.

See:

- [Chapter 12.2, "Receiver Settings"](#), on page 214
- [Chapter 15.5, "How to Create Scenarios with Emitters, Interferers and a Receiver"](#), on page 258
- [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275
- [Chapter 6.2.1, "Common Scenario Settings"](#), on page 56


6.3 How to Select and Create a Test Scenario

The overview in [Table 6-1](#) helps you decide which scenario type is suitable for a certain test case.

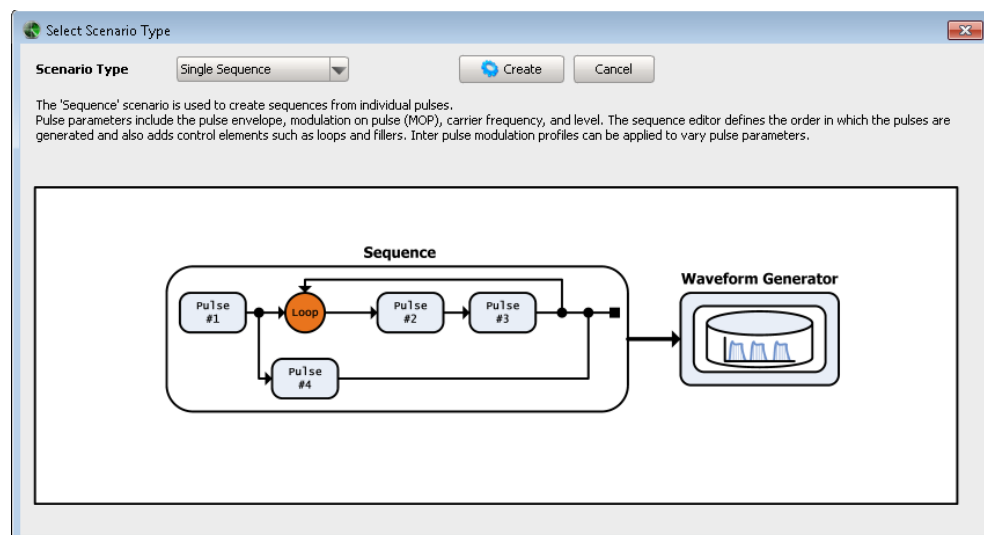
Table 6-1: Typical test cases with their suitable scenario types

Test case	Scenario type
Low-level receiver tests Receiver tests with IPM profiles and modulation on pulse	Single Sequence Sequences (Collection) Waveform Sequence
Advanced receiver tests Receiver tests with antenna patterns and scans, and reception of multiple signals at the same time	Single Emitter Emitters (Collection) Localized Emitters
Receiver tests for direction finding applications Detection of interferer in dense electro-magnetic environments	Direction Finding

To create a single pulse train scenario

1. To access the scenario settings dialog, perform one of the following:
 - a) Select "Repository Tree > Scenario > New".
 - b) In the "Toolbar", select the scenario icon .

A scenario wizard opens that helps you to select a predefined scenario configuration.



2. Select "Scenario Type > Single Sequence" and confirm with "Create"

Created is a new scenario with default name `Scenario <n>`, where `n` is a number starting at one.

The provided settings depend on the selected scenario type.

They are described in [Chapter 6.2, "Scenario Settings"](#), on page 55.
3. Add information to describe and identify the scenario, like a name or a comment. Use the standard functions in the context menus to select or create sequences, select the signal generator, etc. See [Table 3-4](#).

Perform further configurations, see:

- [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124
- [Chapter 14, "Creating and Mapping Generator Profiles"](#), on page 237
- [Chapter 16.3, "How to Analyze the Content of Waveform Files and Files with I/Q Data"](#), on page 269

7 Creating a Pulse Library

This section provides background information on pulse generation and pulse characteristics, explains the provided settings and how to use them to create a pulse library.

- [Basics on Pulse Signals and Pulse Generation](#)..... 70
- [Pulse Settings](#)..... 72
- [How to Create a New Pulse and Adjust Its Settings](#)..... 104

7.1 Basics on Pulse Signals and Pulse Generation

Some background knowledge on basic terms and principles used in pulse generation is provided here for a better understanding of the required configuration settings.

The pulse parameters to be configured are based primarily on the IEEE 181 Standard 181–2003. For detailed descriptions refer to the standard documentation ("IEEE Standard on Transitions, Pulses, and Related Waveforms", from the IEEE Instrumentation and Measurement (I&M) Society, 7 July 2003).

Basic terms

The definition of the pulse parameters depends on the used timing profile (see [Standard Timing Profile](#)) and distinguishes between:

- **Power** 10/50/90
- **Voltage** 10/50/90
- (Power/Voltage) 10/100

Where the name of the timing profile follows the naming convention:

- Power/Voltage: indicates whether power or voltage-based reference levels are used
- Indication of the used reference levels, i.e.:
 - 10/50/90 means that three reference levels are used: the lower (10% nominally), the medial (50% nominally) and the upper (90% nominally)
 - 10/100 indicates that two levels are used: the base (0%) and the top (100%) level

A pulsed signal is a signal whose carrier power is modulated by two states: ON and OFF.

If the top power level of a pulse is not constant, this effect is called an amplitude **droop**.

The graphic on [Figure 7-1](#) illustrates the main pulse parameters and characteristic values when a **Power** 10/50/90 profile is used. If voltage profile is used, the definition of the pulse parameters is identical, but the parameters are related to **voltage**.

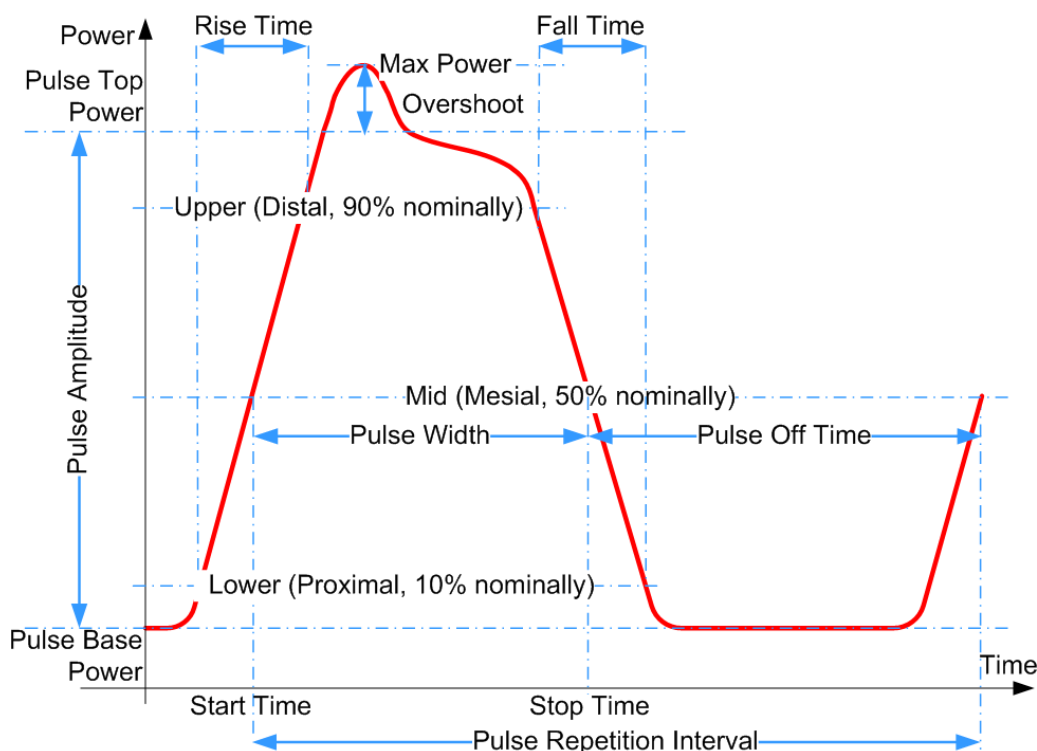


Figure 7-1: Definition of the main pulse parameters and characteristic values ("Timing Profile > Power 10/50/90")

Pulse parameters ("Timing Profile > 10/50/90")

The following definitions apply:

- **Rise time, Fall time** (also known as transition duration of the first and the second transitions)
That is the difference between two reference levels (10% and 90% nominally) of the same transition.
- **Pulse width** (also known as pulse ON time or pulse duration)
That is the difference between the first and the second transition of one pulse, i.e. the duration the signal remains above the 50% reference level.
- **Pulse Off time** (also known as pulse separation)
The duration between the 50% reference level of the second transition of one pulse and the 50% reference level of the first transition of the subsequent pulse in the same pulse train.
- **Pulse repetition interval** (also known as pulse period)
Is defined as the duration of one complete cycle consisting of:
 - The pulse width (i.e. pulse ON time or pulse duration)
 - The pulse Off time (pulse separation)

Pulse parameters ("Timing Profile > 0/100")

Note that the pulse parameters differ, if a "Timing Profile > 0/100" is used.

The following definitions apply:

- **Rise time, Fall time**
That is the difference between the 0% and 100% reference levels of the same transition, i.e. between the base and the top level.
- **Pulse width** (pulse ON time)
The duration the signal power remains at the top level (100% reference level).
- **Pulse Off time**
The duration the signal power remains at the base level (0% reference level).
- **Pulse repetition interval**
Is defined as the duration of one complete cycle consisting of:
 - The rise time
 - The pulse width
 - The fall time
 - The pulse off time

For description of the related settings, see:

- [Chapter 7.2.3, "Pulse Level Settings"](#), on page 77
- [Chapter 7.2.4, "Pulse Envelope Settings"](#), on page 80

7.2 Pulse Settings

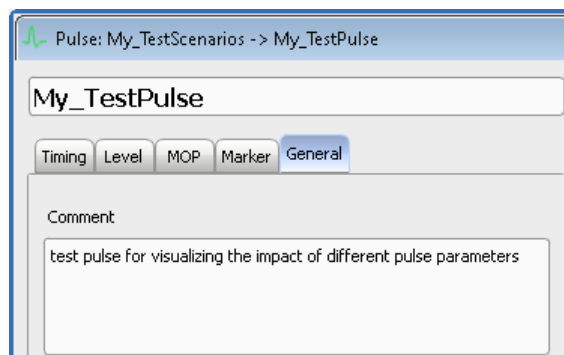
Access:

- ▶ Select "Repository Tree > Pulse > New".

Pulses are characterized by the following settings:

- [Pulse Name and Comment](#)..... 73
- [Pulse Timing Settings](#)..... 74
- [Pulse Level Settings](#)..... 77
- [Pulse Envelope Settings](#)..... 80
- [Modulation on Pulse \(MOP\) Settings](#)..... 83
- [Built-In Modulation Types and their Settings](#)..... 86
- [Envelope and Modulation Graphs](#)..... 101

7.2.1 Pulse Name and Comment



Settings:

Pulse Name	73
Comment	73

Pulse Name

Displays the name of the current pulse.

Remote command:

[PULSe:CREate](#) on page 329

[PULSe:NAME](#) on page 330

[PULSe:CATalog?](#) on page 329

[PULSe:SElect](#) on page 330

[PULSe:REMove](#) on page 331

Comment

General pulse information like comment, etc.

Remote command:

[PULSe:COMMeNt](#) on page 331

7.2.2 Pulse Timing Settings

Access:

- ▶ Select "Pulse > Timing".

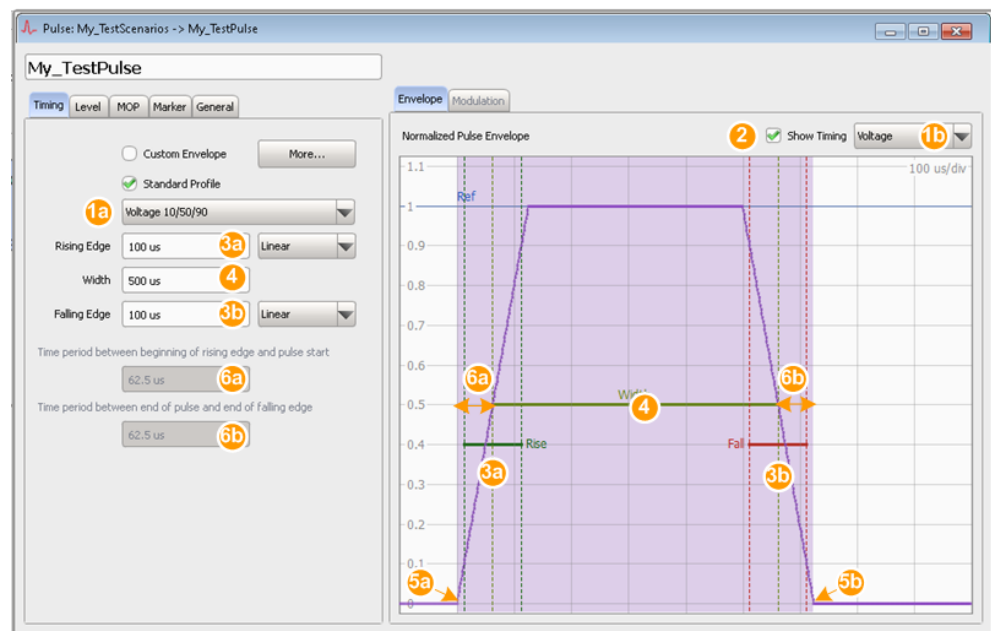


Figure 7-2: Pulse Timing: understanding the displayed information

- 1a = "Standard Profile > Voltage 10/50/90"
- 1b = "Pulse Envelope Units > Voltage"
- 2 = "Show Timing > On" indicates the pulse timing parameters on the envelope graph
- 3a = "Rising Edge = 100 us", i.e. the time it takes the voltage to rise from 10% to 90% of the top level
- 3b = "Falling Edge = 100 us", i.e. the time it takes the voltage to fall from 90% to 10% of the top level
- 4 = "Width = 500 us", i.e. the duration the voltage is above 50% of the top level
- 5a, 5b = Beginning of the rising and falling edge
- 6a = "Time period between beginning of the rising edge and pulse start = 62.5 us", i.e. the time it takes the voltage to rise from 0% to 50% of the top level
- 6b = "Time period between end of pulse and the end of falling edge = 62.5 us", i.e. the time it takes the voltage to fall from 50% to 0% of the top level

Settings:

Use the following parameters to define the pulse in the time domain:

Custom Envelope.....75
 Standard Timing Profile.....75
 Rising/Falling Edge, Width, Rising/Falling Slope..... 76
 Time period between beginning of rising edge and pulse start/Time period between end of pulse and end of falling edge..... 76

Custom Envelope

Applies an envelope shaping according to a custom envelope function.

See Chapter 7.2.4, "Pulse Envelope Settings", on page 80.

Remote command:

PULSe: CUSTom on page 399

PULSe: ENVELOpe:MODE on page 401

Standard Timing Profile

Enables and selects a predefined standard envelope profile.

Select from a voltage-based "Voltage 10/50/90", a power-based "Power 10/50/90" or a 0-100% "0/100" timing profile.

For brief introduction to the timing profiles and their impact on the pulse timing, see Chapter 7.1, "Basics on Pulse Signals and Pulse Generation", on page 70.

See Figure 7-3 for illustration of the pulse width, pulse rise and fall time, and period between the beginning of the rising edge and the pulse start for each of the envelope profiles.

Enable "Show Timing" to visualize the timing parameters. Note that the "Envelope Graphs" use different "Normalized Pulse Envelope" settings.

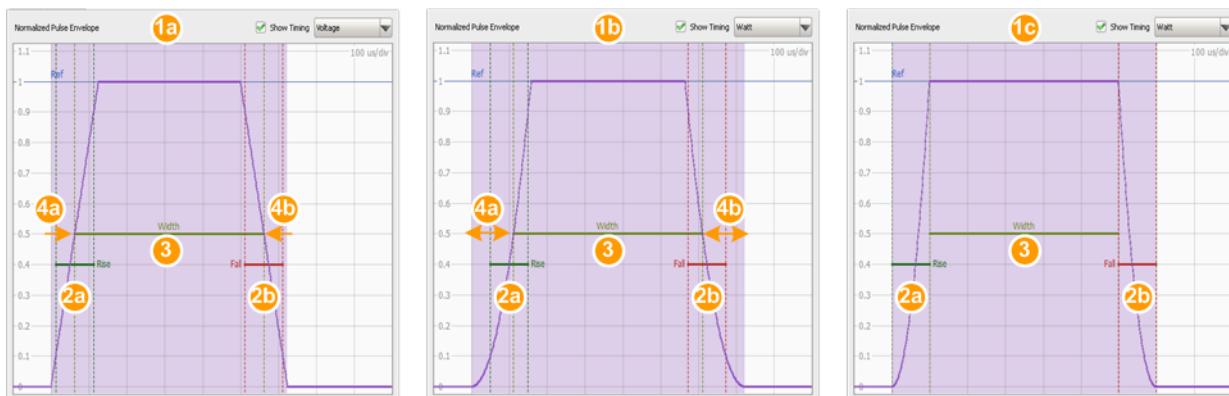


Figure 7-3: Pulse timing parameters and the envelope profiles (linear rising and falling slopes)

- 1a = "Voltage 10/50/90" profile
- 1b = "Power 10/50/90" profile
- 1c = "0/100" profile
- 2a = "Rising Edge = 100 us"
- 2b = "Falling Edge = 100 us"
- 3 = "Width = 500 us"
- 4a = "Time period between beginning of the rising edge and pulse start" (with "0/100" profile this time is 0 us)
- 4b = "Time period between end of pulse and the end of falling edge" (with "0/100" profile this time is 0 us)

Remote command:

[PULSe:TIME:REFerence](#) on page 413

Rising/Falling Edge, Width, Rising/Falling Slope

These parameters define the pulse shape.

The current pulse shape is displayed on the [Envelope Graph](#). To visualize the timing parameters on the graph, select "Envelope > Show Timing > On".

The parameter definition depends on the selected [Standard Timing Profile](#), see:

- ["Pulse parameters \("Timing Profile > 10/50/90"\)"](#) on page 71
- ["Pulse parameters \("Timing Profile > 0/100"\)"](#) on page 71

The total pulse duration and the pulse off time are calculated automatically from the selected pulse width, rise and fall time, and the [PRI](#) (see [PRI](#)).

"Rising/Falling Edge"

Transition time of the rising/falling edge.

Remote command:

[PULSe:TIME:RISE](#) on page 413

[PULSe:TIME:FALL](#) on page 413

"Rising Slope Type"

Form of the rising/falling slope.

Available are linear, cosine, raised and root cosine. You can also use arbitrary envelope data or import your custom shapes, see [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.

Remote command:

[PULSe:TYPE:RISE](#) on page 414

[PULSe:TYPE:FALL](#) on page 414

"Width" Pulse duration.

Remote command:

[PULSe:TIME:WIDTh](#) on page 413

Time period between beginning of rising edge and pulse start/Time period between end of pulse and end of falling edge

Display information on the time periods during that the RF power is attenuated or suppressed but per default there is modulation or data content present.

Note: Avoiding data truncation.

To avoid truncation of data during the rising time, add guard bits to the data source.

Per default, modulation is already present during the rising and falling phases. You can restrict the pulse area on that the modulation is applied, see ["Restrict modulation to a certain area of the pulse"](#) on page 84.

7.2.3 Pulse Level Settings

Access:

- ▶ Select "Pulse > Level".

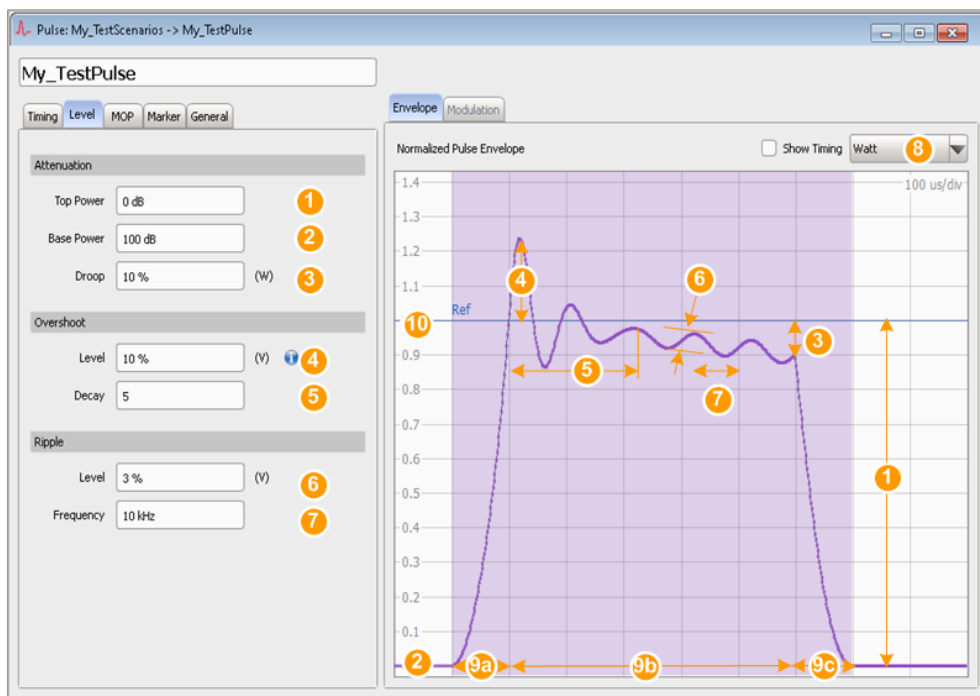


Figure 7-4: Pulse Level: understanding the displayed information

- 1 = Pulse top power corresponds to 1000 mW ("Pulse Envelope = Watt"), i.e. little or no attenuation ("Attenuation Top Power = 0 dB")
- 2 = "Base Power", i.e. high attenuation; typically 100 dB
- 3 = Amplitude change from the beginning to the end of the pulse ON time
- 4 = Overshoot level, in Volt (diagram uses Watts)
- 5 = Damping constant that influences the time it takes the overshoot to decay
- 6 = Amplitude of the oscillating signal, in Volt (diagram uses Watts)
- 7 = Signal oscillating with the selected frequency ("Frequency = 10 kHz", i.e. the oscillation period is 100 us)
- 8 = Pulse amplitude units (Y-axis units)
- 9a, 9b, 9c = Pulse timing: "Profile > Standard 0/100", "Rising Edge = Falling Edge = 100 us", "Width = 500 us" (see [Standard Timing Profile](#))
- 10 = Reference value used by the calculation of the pulse level parameters: "Ref. = 0 dB (1 Watt)"; the value is set with the parameter [Ref. Level](#)

Settings:

Use the following parameters to define the power/amplitude pulse parameters:

[Attenuation](#)..... 77

[Overshoot](#)..... 78

[Ripple](#)..... 79

Attenuation

Simulates changes in the pulse amplitude.

- "Top Power" The pulse top power is used as a reference (100%) to determine other parameter values such as the rising/falling thresholds. Typically, the "Top Power" uses a little or no attenuation, such as 0 dB. See also [Figure 7-1](#).
- "Base Power" The pulse base power is used as a reference (0%) to determine other parameter values such as the rising/falling thresholds. Typically, the "Base Power" uses a high attenuation, such as 100 dB. You can change this value if you use a "0/100" profile. The difference between the "Top Power" and the "Base Power" is the pulse amplitude.
- "Droop" Droop is a measure for the amplitude change from the beginning to the end of the pulse. The percentage ratio values are calculated in %W. A positive value decreases the amplitude.

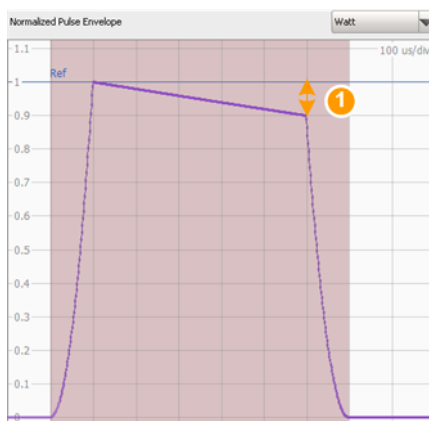


Figure 7-5: Illustration of the effect of the amplitude droop (Profile = 0/100)

1 = "Droop = 10% (W)"

Remote command:

[PULSe:LEVel:ON](#) on page 401

[PULSe:LEVel:OFF](#) on page 401

[PULSe:LEVel:DROOp](#) on page 401

Overshoot

Simulates that the signal exceeds the pulse "Top Power".

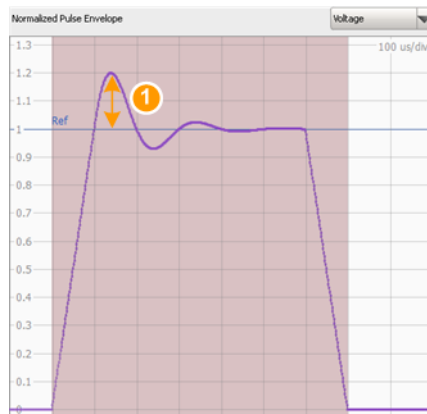


Figure 7-6: Illustration of the effect of the overshoot parameters (Profile = 0/100)

1 = "Overshoot = 20%" (value in percentage related to Volts)

"Level" The height of the local maximum (Max. Power) after a rising edge, divided by the pulse amplitude, see [Figure 7-1](#).

"Decay" Sets the normalized decay constant. It influences the time it takes the overshoot to decay.

"Decay = 1" is the aperiodic border case. The bigger the value the longer the decay time.

Remote command:

[PULSe:OVERshoot](#) on page 412

[PULSe:OVERshoot:DECay](#) on page 412

Ripple

Simulates the unwanted ringing effect, i.e. that the signal oscillates around the power level. This oscillation is defined with a frequency and level.

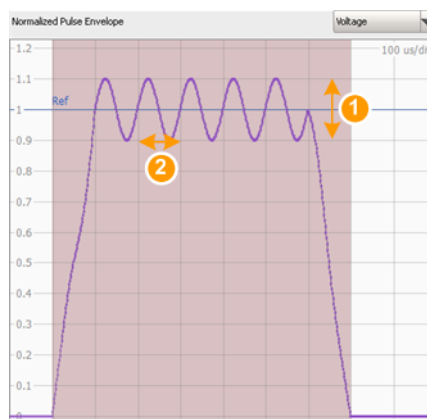


Figure 7-7: Illustration of the effect of the ripple parameters (Profile = 0/100)

1 = "Level = 20% " (value in percentage related to Volts)

2 = "Frequency = 10 kHz", i.e. the oscillation period is 100 us

- "Level" The ripple is calculated as the difference between the maximum and minimum deviation from the pulse top reference, within a user specified interval.
The percentage ratio values are calculated in %W, if the "Pulse Envelope" is defined in W, otherwise in %V.
 - "Frequency" Sets with which frequency the ripple oscillates.
- Remote command:
[PULSe:RIPPlE](#) on page 412
[PULSe:RIPPlE:FREQuency](#) on page 413

7.2.4 Pulse Envelope Settings

Instead of defining a pulse by its rise, on and fall time you can use your custom envelope shape.



Custom envelopes are described in one of the following ways:

- As an equation, see "[Envelope Definition as an Equation](#)" on page 80
- As a function of imported I/Q data, see "[Envelope Definition as a Function form Imported Data](#)" on page 81

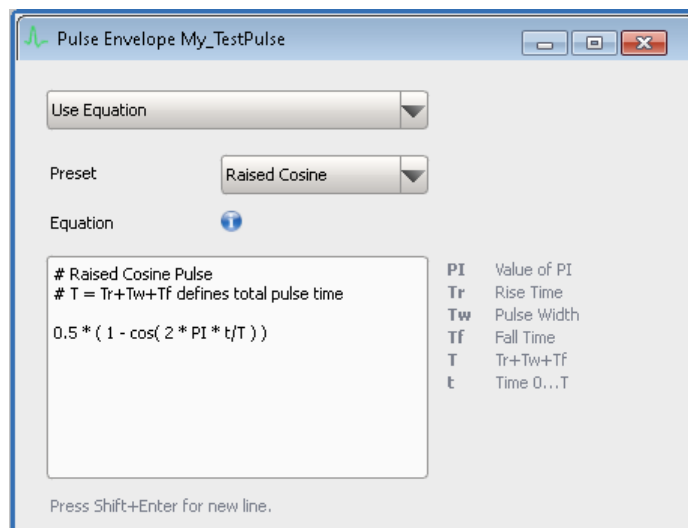
Both ways affect the level versus time and can be used with any kind of modulation (MOP). Custom envelopes are multiplied with the existing pulse shape as defined with the timing parameters. The software maps the custom envelope to the pulse shape.

Settings:

[Envelope Definition as an Equation](#)..... 80
[Envelope Definition as a Function form Imported Data](#).....81

Envelope Definition as an Equation

Envelope is described by an equation, as selected with the parameter "Preset".



The software provides a selection of envelope shapes. Upon selection, the dialog displays the used equation and information on the used parameters and variables.

For description of the provided envelope functions, see ["Equations and parameters used to define custom pulse envelopes"](#) on page 493.

See also:

- ["To generate a pulse with raised cosine envelope shape"](#) on page 108
- ["To define your custom envelope shape with an equation"](#) on page 109

Remote command:

`PULSe:ENvelope:EQUation` on page 400

Envelope Definition as a Function form Imported Data

You can define an envelope function, store it in a file, and import and load files describing envelope functions.

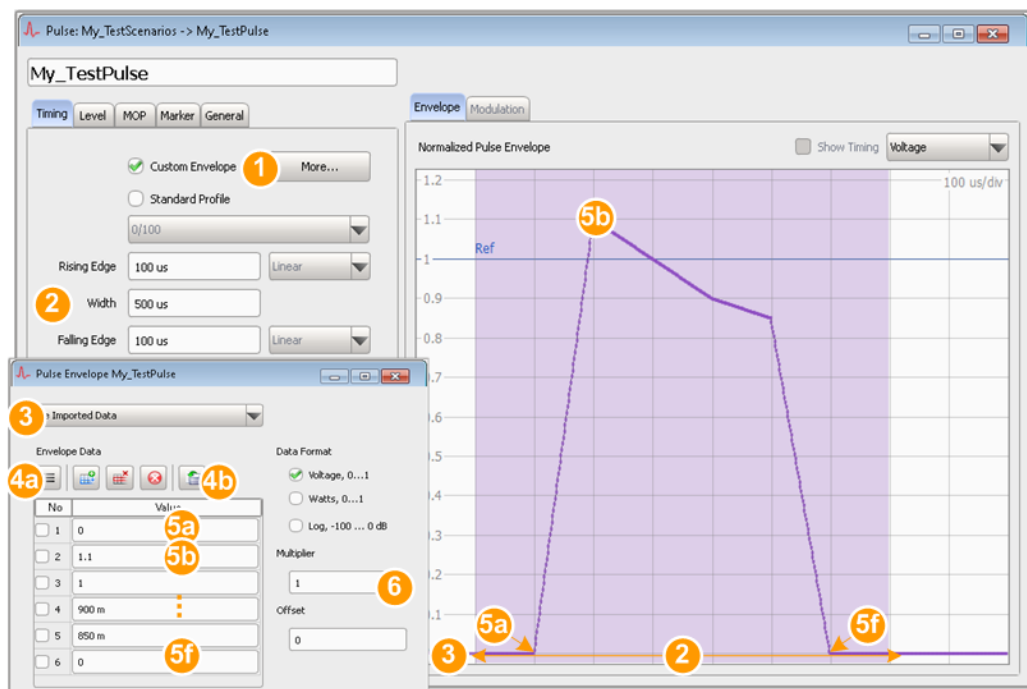


Figure 7-8: "Custom Envelope > Use Imported Data": understanding the displayed information

- 1 = "Custom Envelope"
- 2 = The duration between the beginning of the rising edge and the end of the falling edge is $100\ \mu\text{s} + 500\ \mu\text{s} + 100\ \mu\text{s} = 700\ \mu\text{s}$; i.e. the x-axis values of the 6 items are distributed equidistant, at each $100\ \mu\text{s}$
- 3 = "Use Imported Data"
- 4a = Icons that access standard functions
- 4b = "Import Data from File" access a dialog for loading of custom envelopes
- 5a, 5b, ... = "Value", i.e. value on the y-axis
- 5e = "Offset = 0"
- 6 = "Offset = 0"

"Envelope Data Table"

An envelope curve consists of two or more items. Each item is a line, defined by the y-value of end point; the start point is always the end point of the previous item.

The x-axis values are calculated automatically so that the items are distributed equidistant between the beginning of the rising edge and the end of the falling edge.

"Data Format" Selects the scaling used by the [Envelope Graph](#).

"Multiplier" Applies a multiplier to the Y-values.

"Offset" Offsets the envelope.



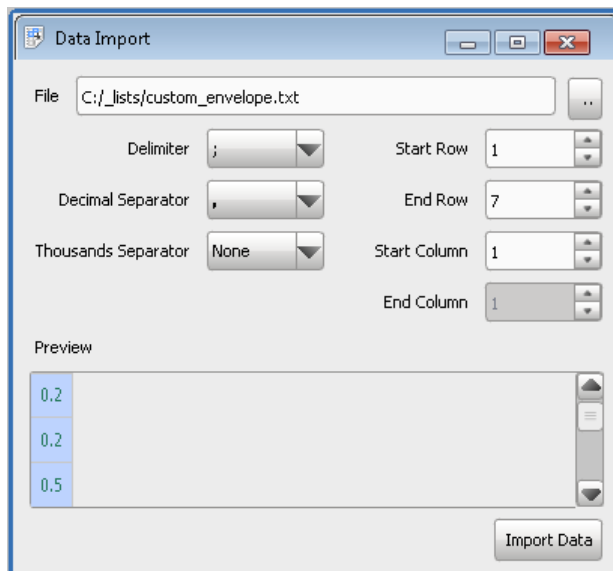
Icons that access standard functions, like add or delete row (see also [Table 3-4](#))

Use the functions in the context-sensitive menu to append new item or reorder the existing item.

"Import Data from File"

Standard file management function to load custom envelopes from and as an ASCII file with *.txt or *.csv file extension.

When file is loaded, the "Data Import" dialog assists you to select the used delimiter and separator to interpret the file content.



The provided settings are self-explanatory.

Remote command:

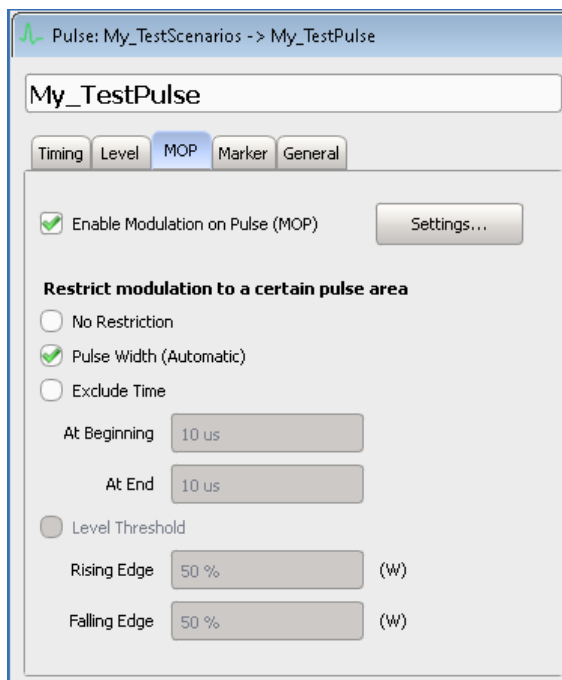
- [PULSe:ENVELOpe:DATA:ITEM:ADD](#) on page 332
- [PULSe:ENVELOpe:DATA:ITEM:SELEct](#) on page 333
- [PULSe:ENVELOpe:DATA:ITEM:DELEte](#) on page 334
- [PULSe:ENVELOpe:DATA:ITEM:VALue](#) on page 399
- [PULSe:ENVELOpe:DATA:UNIT](#) on page 399
- [PULSe:ENVELOpe:DATA:MULTIplier](#) on page 400
- [PULSe:ENVELOpe:DATA:OFFSet](#) on page 400
- [PULSe:ENVELOpe:DATA:LOAD](#) on page 399
- [PULSe:ENVELOpe:DATA:CLEAr](#) on page 335

7.2.5 Modulation on Pulse (MOP) Settings

Access:

1. Select "Pulse > MOP".

2. Select "Enable Modulation on Pulse (MOP)".



Settings:

Use the following common parameters to define and enable modulation on the pulses:

[Enable Modulation on Pulse \(MOP\)](#)..... 84
[Settings](#)..... 84
[Restrict modulation to a certain area of the pulse](#)..... 84

Enable Modulation on Pulse (MOP)

Enables/disables that the pulse signal is modulated. If the MOP is disabled, the software generates an envelope signal based on the pulse level and timing parameters.

Remote command:

[PULSe:MOP:ENABLe](#) on page 406

Settings...

Access the dialog with modulation settings, see [Built-In Modulation Types and their Settings](#).

Restrict modulation to a certain area of the pulse

Customizes (restricts) the pulse area on that the MOP is applied.

With its default settings ("Configure > Colors > Preset > Classic"), the software indicates (highlights) the pulse area on that the MOP is applied. See [Figure 7-9](#) and compare the colors used to indicate the pulse (lilac) and the pulse area on that MOP is applied (pastel orange).

See also [Chapter 3.7.2, "Changing Colors and Default Configuration"](#), on page 36.

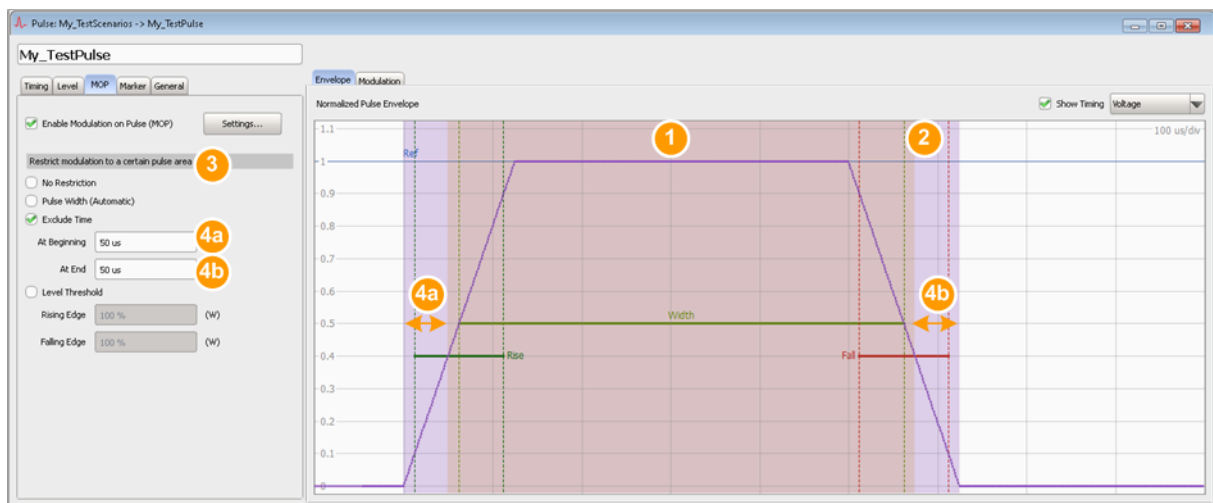


Figure 7-9: Restrict modulation to a certain area of the pulse: understanding the displayed information ("Timing Profile > Voltage 10/50/90")

- 1 = Orange color: indicates the pulse area on that the MOP is applied
- 2 = Lilac color: indicates the pulse
- 3 = "Restrict modulation to a certain area of the pulse > Exclude Time"
- 4a, 4b = Excluded are 50 us from the beginning of the rising edge and at the end of the falling edge; note that the pulse area the MOP is applied on is wider than the pulse width

"No restrictions"

The MOP is applied from the beginning of the rising edge until the end of the falling edge.

"Pulse Width (Automatic)"

The MOP is applied during the time specified as pulse width time ("Pulse > Timing > Width").

"Exclude time" Defines a time span to be excluded at the beginning of the rising edge and at the end of the falling edge.

"Level Threshold"

The MOP is applied during the time the signal level is greater than the specified level limits.

Remote command:

[PULSe:MOP:EXCLude:ENABle](#) on page 407

[PULSe:MOP:EXCLude:MODE](#) on page 407

[PULSe:MOP:EXCLude:TIME:STARt](#) on page 407

[PULSe:MOP:EXCLude:TIME:STOP](#) on page 407

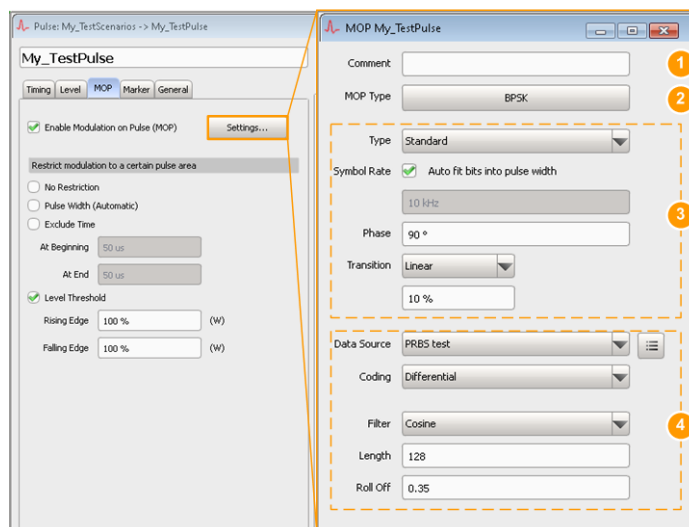
[PULSe:MOP:EXCLude:LEVel:STARt](#) on page 407

[PULSe:MOP:EXCLude:LEVel:STOP](#) on page 407

7.2.6 Built-In Modulation Types and their Settings

Access:

- ▶ Select "Pulse > MOP > Settings".



1 = "Comment"

2 = "MOP Type" selects the modulation scheme

3 = Modulation parameters, depending on the selected "MOP Type"

4 = "Data source, Coding and Filter" settings (if required by the selected "MOP Type")

Use the parameter "MOP Type" to select a modulation scheme and define the respective modulation parameters. You can also add a description in the "Comment" field.

Remote commands:

[PULSe:MOP:COMMeNt](#) on page 331

[PULSe:MOP:TYPe](#) on page 412

Further available settings depend on the selected modulation.

- [AM and ASK Modulation](#)
- [FM and FSK Modulation](#)
- [Chirp Modulation](#)
- [Phase Modulation](#)
- [QPSK Modulation](#)
- [Noise](#)
- [Plugins](#)

The [Modulation Graph](#) displays the characteristics of the currently used MOP.

Some MOP types require a data source. For description of the related settings, see [Chapter 7.2.6.8, "Data Source, Coding, Filter"](#), on page 100.

7.2.6.1 AM and ASK Modulation

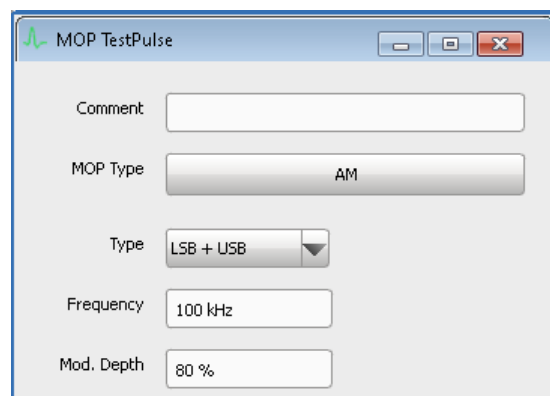
The amplitude modulation (AM) and amplitude shift key modulation (ASK) are modulation schemes that vary the amplitude of the signal.

Settings:

AM.....	87
ASK.....	87
AM Step.....	88

AM

Enables an AM with a single tone.



"AM Type" Modulation method, like a double-sideband ("Standard"), a Low Side Band ("LSB"), an Upper Side Band ("USB"), or a double sideband with suppressed carrier ("LSB+USB") amplitude modulation.

"Frequency" Modulation frequency.

"Modulation Depth" Depth of the modulation signal in percent, i.e. the peak change in the RF amplitude from its unmodulated value to the amplitude of the unmodulated carrier.

Remote command:

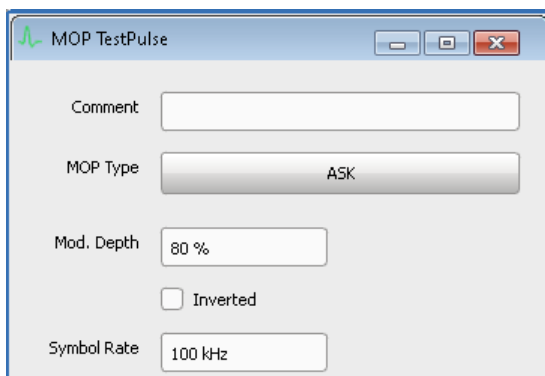
[PULSe:MOP:AM:TYPE](#) on page 402

[PULSe:MOP:AM:FREQuency](#) on page 401

[PULSe:MOP:AM:MDEPth](#) on page 402

ASK

Enables an ASK where the amplitude of the RF carrier is attenuated for a symbol value of zero and remains at full level for symbol values of one. The level of attenuation is specified as depth in percent.



"Modulation Depth"

Depth of the modulation signal in percent.
 $m = (Amplitude_{max} - Amplitude_{min}) / (Amplitude_{max} + Amplitude_{min})$

"Inverted"

Inverts the amplitude-shift keying (ASK) modulation. That is, the RF carrier is attenuated for a symbol value of one.

"Symbol Rate"

Symbol rate of the modulating signal.

"Data Source, Coding, Filter"

See:

- [Chapter 18.1, "Data Sources Settings"](#), on page 293
- ["Coding"](#) on page 100
- ["Filter"](#) on page 101

Remote command:

- [PULSe:MOP:ASK:MDEPth](#) on page 403
- [PULSe:MOP:ASK:INVert](#) on page 403
- [PULSe:MOP:ASK:SRATe](#) on page 403

AM Step

Defines a modulation as a sequence of discrete states (table rows), each described with its "Duration" and "Level".

Use the standard "Append", "Remove Last", or "Delete All" functions to add or remove a row. The provided settings are self-explanatory.

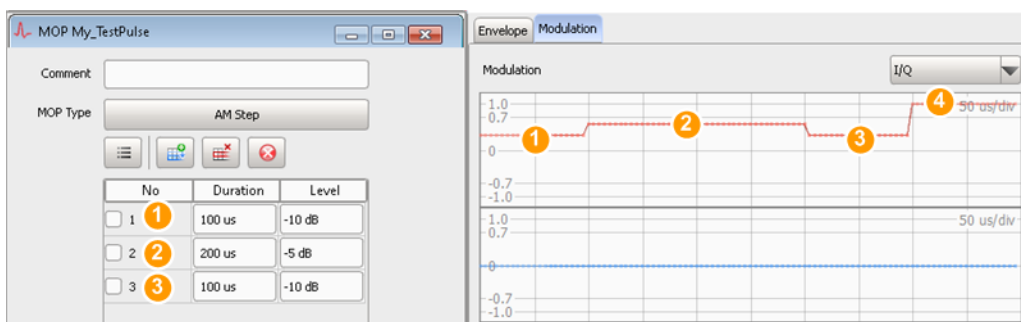


Figure 7-10: "MOP Type > AM Step": understanding the displayed information ("Pulse Width = 500 us", "Restrict MOP to > Pulse Width")

- 1, 2, 3 = Three discrete states with different attenuation values and a total duration of 400 us
- 4 = The last 100 us are not attenuated

If the total time of the defined states is shorter than the duration of the pulse area on that the MOP should be applied, an attenuation of 0 dB is assumed for the remaining time.

Remote command:

- [PULSe:MOP:AMSTep:ADD](#) on page 332
- [PULSe:MOP:AMSTep:COUNT?](#) on page 333
- [PULSe:MOP:AMSTep:SElect](#) on page 333
- [PULSe:MOP:AMSTep:INSert](#) on page 334
- [PULSe:MOP:AMSTep:LEVel](#) on page 402
- [PULSe:MOP:AMSTep:DURation](#) on page 402
- [PULSe:MOP:AMSTep:CLEar](#) on page 335
- [PULSe:MOP:AMSTep:DELeTe](#) on page 334

7.2.6.2 FM and FSK Modulation

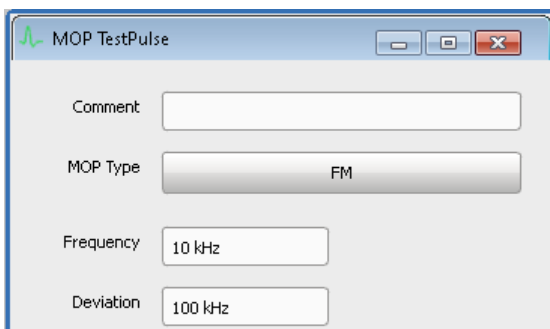
The frequency modulation (FM) and frequency shift keying (FSK) modulation are modulation schemes that vary the frequency of the signal.

Settings:

FM	89
FSK	89
FM Step	90

FM

Enables an FM with a single tone.



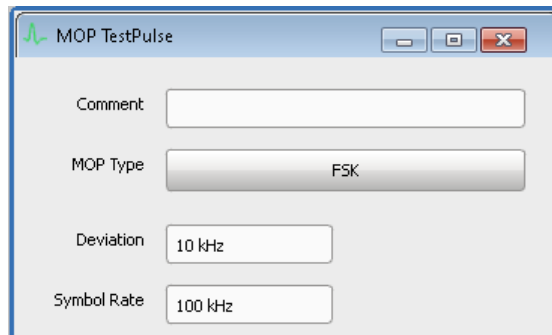
- "Frequency" Modulation frequency
- "Deviation" FM frequency deviation

Remote command:

- [PULSe:MOP:FM:FREQuency](#) on page 409
- [PULSe:MOP:FM:DEVIation](#) on page 409

FSK

Enables an FSK. High symbols set the frequency to "+Deviation" and low symbols to "-Deviation".



"Deviation" FSK frequency deviation

"Symbol Rate" Symbol rate of the modulating signal.

"Data Source, Coding, Filter"

See:

- [Chapter 18.1, "Data Sources Settings"](#), on page 293
- ["Coding"](#) on page 100
- ["Filter"](#) on page 101

Remote command:

[PULSe:MOP:FSK:DEVIation](#) on page 409

[PULSe:MOP:FSK:SRATE](#) on page 409

FM Step

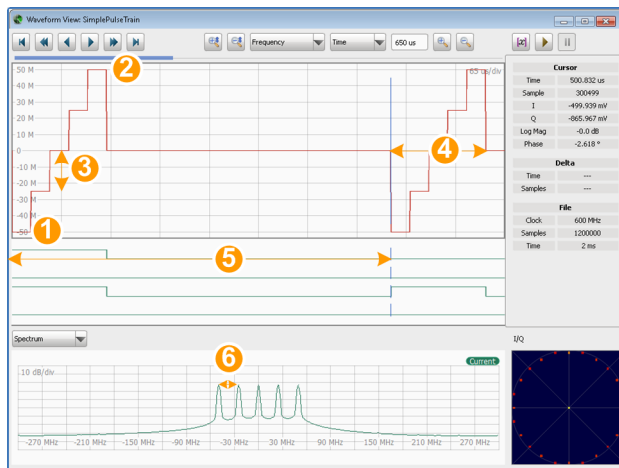
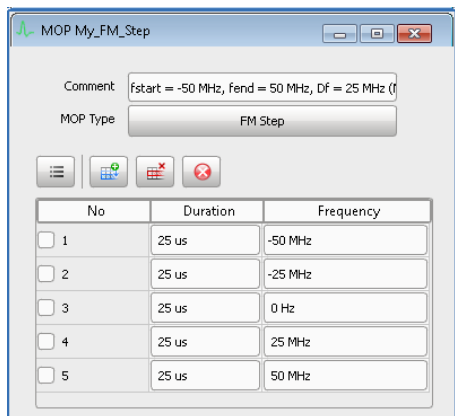
The FM step modulation is a common pulse compression technique. It defines a modulation as a sequence of discrete states (table rows), each described with its "Duration" and "Frequency".

Use the standard "Append", "Remove Last", or "Delete All" functions to add or remove a row. The provided settings are self-explanatory.

Example: Sequence composed of four pulses with MOP=FM Step

Configure an FM Step with $f_{start} = -50$ MHz, $f_{end} = 50$ MHz, Step = 25 MHz, #Steps = 5.

Build a sequence and use the "Waveform View" dialog to visualize the signal, see Chapter 16.3, "How to Analyze the Content of Waveform Files and Files with I/Q Data", on page 269.



Example: Sequence composed of four pulses with MOP=FM Step

- 1, 2: Start and stop frequency
- 3: Step = 25 MHz
- 4: Pulse width = 125 us
- 5: PRI = 500 us
- 6: Frequency spacing = 25 MHz

Remote command:

- [PULSe:MOP:FMSTep:ADD](#) on page 332
- [PULSe:MOP:FMSTep:COUNT?](#) on page 333
- [PULSe:MOP:FMSTep:SElect](#) on page 333
- [PULSe:MOP:FMSTep:INSert](#) on page 334
- [PULSe:MOP:FMSTep:FREquency](#) on page 403
- [PULSe:MOP:FMSTep:DURation](#) on page 402
- [PULSe:MOP:FMSTep:CLEar](#) on page 335
- [PULSe:MOP:FMSTep:DELeTe](#) on page 335

7.2.6.3 Chirp Modulation

A chirp is a signal in which the frequency varies over the time. The chirp sweeps the RF signal across a set frequency range.

The chirp modulation scheme is used for pulse compression.

Settings:

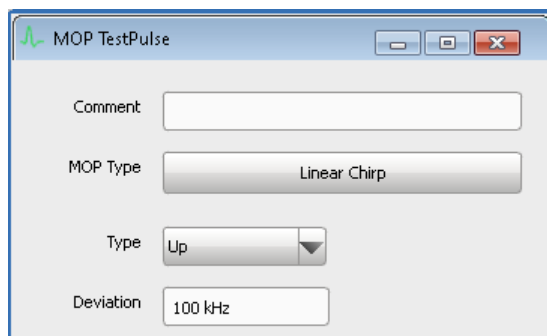
[Linear Chirp](#).....92

[Equation](#).....92

[Polynomial Chirp](#).....92

Linear Chirp

The linear chirp varies the frequency linearly over time.



- "Type" Defines the frequency variation:
- Linearly increasing (UP)
 - Linearly decreasing (Down)
 - Following a full sine wave (Sine)
 - Ascending and then descending (triangle)

"Deviation" Chirp deviation.

Remote command:

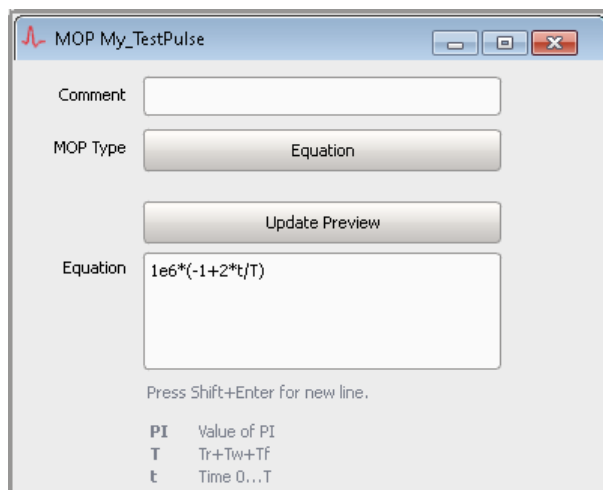
[PULSe:MOP:CHIRp:TYPE](#) on page 406

[PULSe:MOP:CHIRp:DEVIation](#) on page 406

Equation

Define your custom chirp mathematically as a function of time, e.g. $1e6*(-1+2*t/T)$.

Select "Update Preview" to visualize the signal.



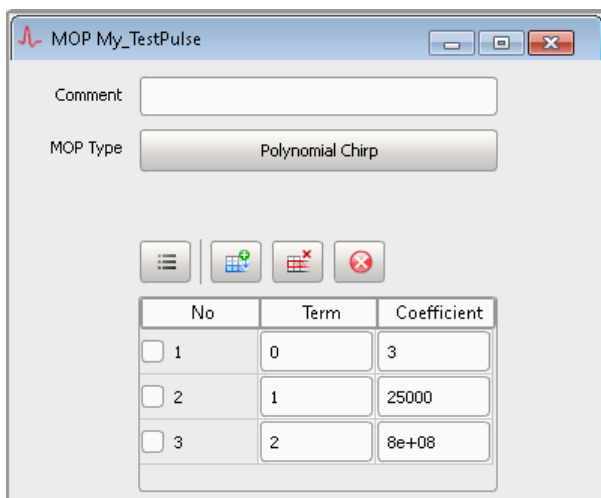
See also [Chapter B, "Formula Syntax"](#), on page 492.

Remote command:

[PULSe:MOP:NLCHirp:EQUation](#) on page 410

Polynomial Chirp

Define the chirp as a polynomial with one or more "Term"/"Coefficient" value pairs.



The instantaneous frequency versus time is calculated according to the equation:

$$f(t) = s * \sum (a_n * t^n), \text{ where:}$$

- n = 1 .. 5
- The multiplier s and the coefficients a_n are values within the range – 10⁶ to 10⁶.

Remote command:

- [PULSe:MOP:PCHirp:ADD](#) on page 332
- [PULSe:MOP:PCHirp:COUNT?](#) on page 333
- [PULSe:MOP:PCHirp:SElect](#) on page 333
- [PULSe:MOP:PCHirp:INSert](#) on page 334
- [PULSe:MOP:PCHirp:TERM](#) on page 410
- [PULSe:MOP:PCHirp:COEFFicient](#) on page 410
- [PULSe:MOP:PCHirp:CLEar](#) on page 335
- [PULSe:MOP:PCHirp:DELeTe](#) on page 335

7.2.6.4 Phase Modulation

Provided are the following phase modulation technics:

Barker	93
Poly Phase	95
Custom Phase	96
BPSK	96

Barker

Phase modulation with Barker codes results in signals with low autocorrelation properties.

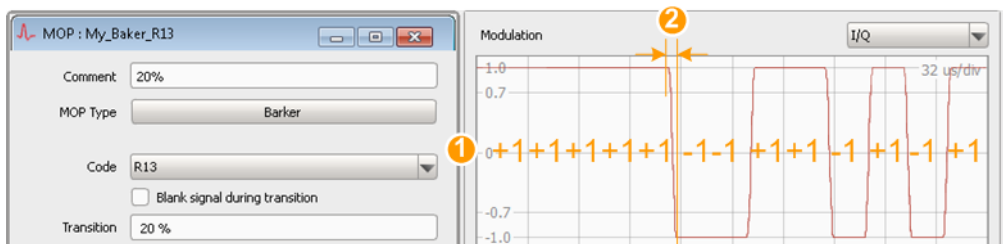


Figure 7-11: "MOP Type > Barker": understanding the displayed information

1 = Barker sequence "Code = R13"

2 = Transition time for "Transition = 20%" ("Blank signal during transition > Off")

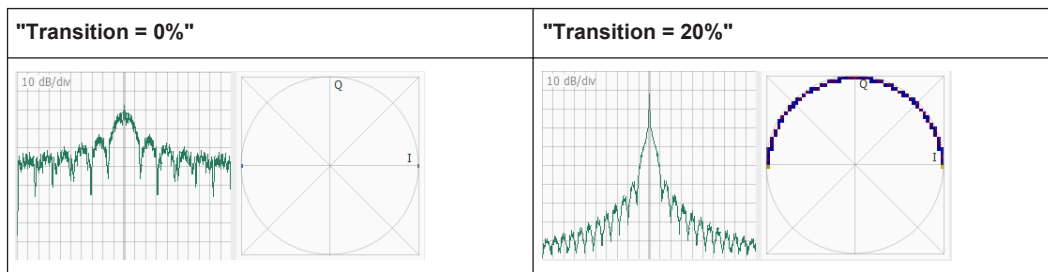
Barker codes are used for pulse compression and are defined by:

"Barker Code" A Barker code is a finite sequence of N values of +1 and -1; R13 for example is the '+1+1+1+1-1-1+1+1-1+1-1+1' sequence.

"Transition" Sets the transition duration for constant envelope operation. A "Transition = 0%" is the border case, where the change for +1 to -1 or conversely is performed immediately. The transition skips (switches back and forth on a direct line) between the constellation points.

With a "Transition > 0%", the two constellation points are still exactly opposite, but the transition follows a circular arc. The values on the transition curve are uniformly distributed.

Observe the effect of a "Transition > 0%" on the modulation envelope, the displayed spectrum, and on the I/Q constellation diagram. The greater the value, the slower the transition, the longer the transition time.



"Blank signal during transition"

If "Transition > 0%", blanks out the signal during the transition time. Observe the effect on the modulation envelope, the displayed spectrum, and on the I/Q constellation diagram.

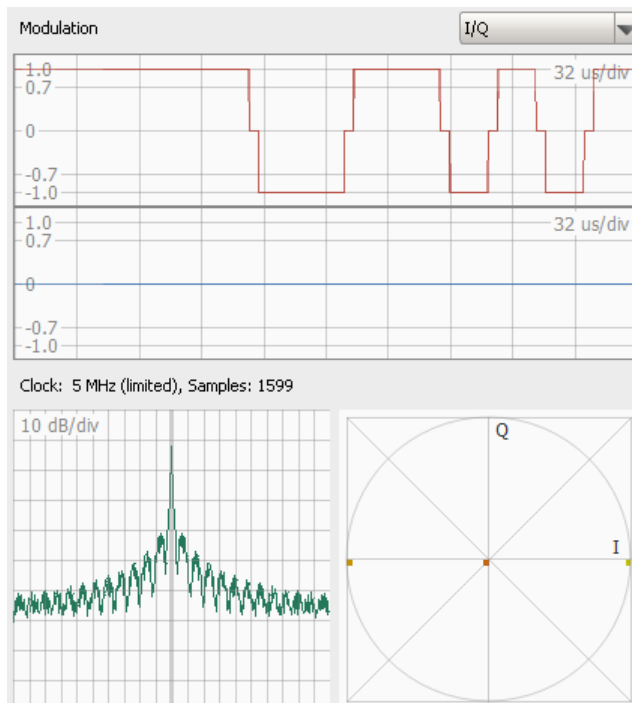


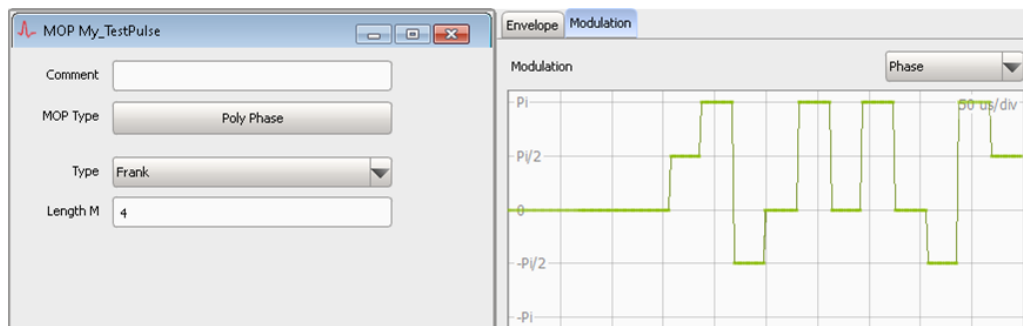
Figure 7-12: Effect of the "Blank signal during transition = On" ("MOP Type = Barker", "Code = R13" modulation with "Transition = 20%")

Remote command:

- [PULSe:MOP:BARKer:CODE](#) on page 404
- [PULSe:MOP:BARKer:TTIME](#) on page 404
- [PULSe:MOP:BARKer:BLANK](#) on page 404

Poly Phase

A polyphase code modulation is suitable for pulse compression and is used in Low Probability of Intercept (LPI) radars.



"Type"

Selection of polyphase compression codes: the default Frank code, and four modified versions of the Frank code, the P1, P2, P3 and P4 codes.

"Length M" Sets the code order, i.e. the number of elements.

Remote command:

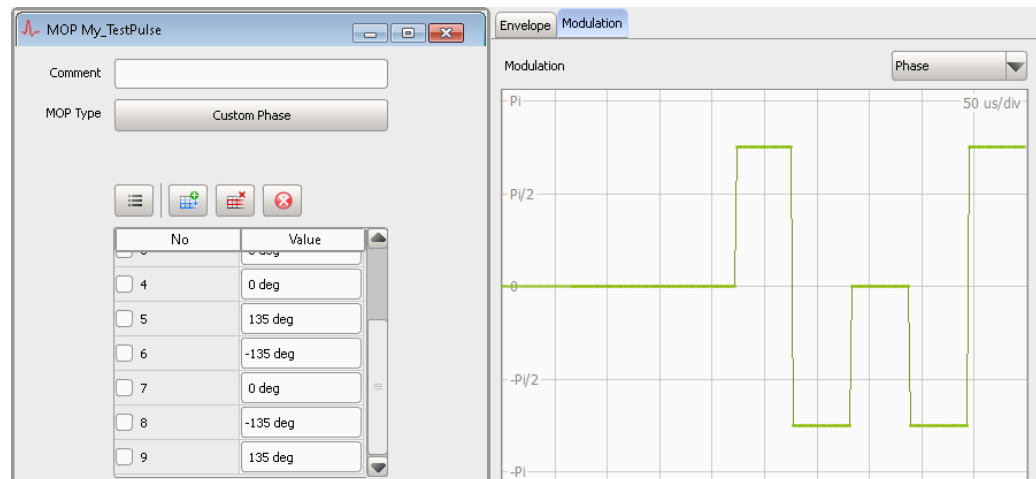
[PULSe:MOP:POLY:TYPE](#) on page 411

[PULSe:MOP:POLY:LENGTh](#) on page 411

Custom Phase

Defines a sequence of phase values. The phase changes are distributed evenly.

Use the standard "Append", "Remove Last", or "Delete All" functions to add or remove a row. The provided settings are self-explanatory.



Remote command:

[PULSe:MOP:PLISt:ADD](#) on page 332

[PULSe:MOP:PLISt:COUNT?](#) on page 333

[PULSe:MOP:PLISt:SELEct](#) on page 333

[PULSe:MOP:PLISt:INSert](#) on page 334

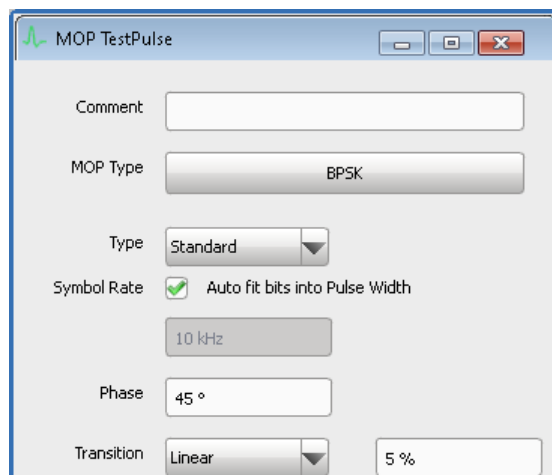
[PULSe:MOP:PLISt:VALue](#) on page 410

[PULSe:MOP:PLISt:CLEar](#) on page 335

[PULSe:MOP:PLISt:DELEte](#) on page 335

BPSK

A Binary Phase Shift Keying (BPSK) modulation applies a phase shift with a selected offset ("Phase") at each constellation point.



"Type" Selects the BPSK modulation method, like a standard BPSK modulation or a BPSK modulation with a constant envelope (**C-BPSK**).

"Symbol Rate" Symbol rate of the modulating signal, user defined or automatically estimated.

"Phase" Phase shift between the constellation point.

"Transition/Transition Function"

These parameters are important, when a BPSK modulation with a constant envelope is used.

A "Transition = 0%" is a border case, where the change between two constellation points is performed immediately. The transition skips (switches back and forth on a direct line) between the constellation points.

The greater the value, the slower the transition. With a "Transition > 0%", the two constellation points are still exactly opposite, but the transition follows a circular arc.

The "Transition Function" defines how the values are distributed on the transition curve.

- "Linear": all values are uniformly distributed in the transition
- "Cosine": there are fewer values in the middle of the transition and accumulated values near to the two end constellation points.

"Data Source, Coding, Filter"

See:

- [Chapter 18.1, "Data Sources Settings"](#), on page 293
- ["Coding"](#) on page 100
- ["Filter"](#) on page 101

Remote command:

[PULSe:MOP:BPSK:TYPE](#) on page 405

[PULSe:MOP:BPSK:SRATe](#) on page 405

[PULSe:MOP:BPSK:SRATe:AUTO](#) on page 405

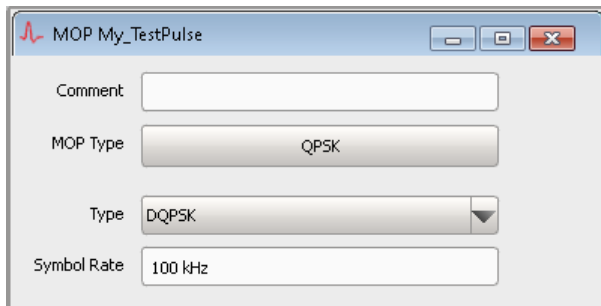
[PULSe:MOP:BPSK:PHASe](#) on page 404

[PULSe:MOP:BPSK:TTYPe](#) on page 405

[PULSe:MOP:BPSK:TTIMe](#) on page 405

7.2.6.5 QPSK Modulation

A quadrature phase shift keying (QPSK) modulation applies a phase shift at each constellation point.



QPSK Type

Selects the QPSK modulation method.

"QPSK, OQPSK, DQPSK"

A regular QPSK, an offset QPSK ("OQPSK"), or a differential QPSK "DQPSK" modulation.

"SOQPSK-A, SOQPSK-B, SOQPSK-TG"

3 variants of the shaped-offset QPSK ("SOQPSK") modulation.

For the aeronautical telemetry "SOQPSK-TG" modulation, you can enable differential encoding according to the telemetry standard IRIG 106-04.

Remote command:

[PULSe:MOP:QPSK:TYPE](#) on page 411

[PULSe:MOP:QPSK:SOQPsk:IRIG](#) on page 411

Symbol Rate

Symbol rate of the modulating signal.

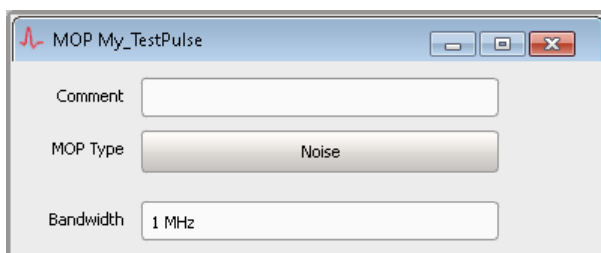
Remote command:

[PULSe:MOP:QPSK:SRATe](#) on page 411

7.2.6.6 Noise

Generates an **AWGN** noise signal with selected "Bandwidth".

The noise is superimposed during all pulse phases.



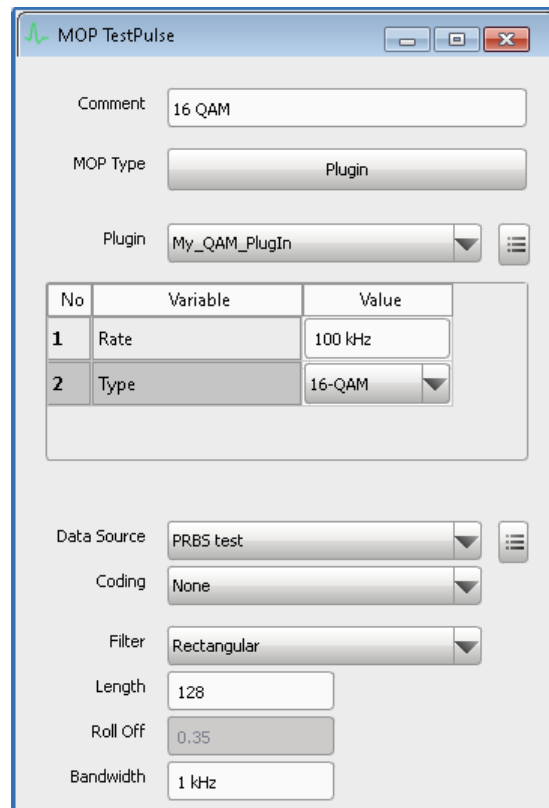
Remote command:

[PULSe:MOP:NOISe:BWIDth](#) on page 410

7.2.6.7 Plugins

Custom modulation schemes are used in the same way as the internal modulation schemes.

See [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.



No	Variable	Value
1	Rate	100 kHz
2	Type	16-QAM

Settings:

Plugin , 	99
Plugin Variables	100
Data Source, Coding, Filter	100

Plugin ,

Selects and loads a user-defined modulation from a file.

The modulation must exist in the "Plugin" library.

Use the standard context menu functions to:

- Reset the plugin variables
- Access the load "Plugin" settings, see [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297

Remote command:

[PULSe:MOP:PLUGIn:NAME](#) on page 330

Plugin Variables

Resets all variables to the preset values defined in the plugin.

Remote command:

[PULSe:MOP:PLUGIn:VARiable:CATalog?](#) on page 393

[PULSe:MOP:PLUGIn:VARiable:SElect](#) on page 393

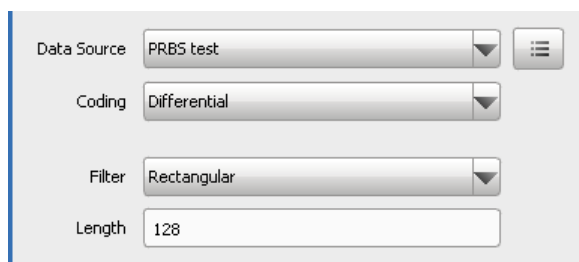
[PULSe:MOP:PLUGIn:VARiable:VALue](#) on page 393

Data Source, Coding, Filter

See:

- [Chapter 18.1, "Data Sources Settings"](#), on page 293
- ["Coding"](#) on page 100
- ["Filter"](#) on page 101

7.2.6.8 Data Source, Coding, Filter



Settings:

The following settings are available for a subset of the provided MOP types:

Data Source	100
Coding	100
Filter	101

Data Source

Selects the data source for the custom modulation.

The data source has to be created first. See [Chapter 18, "Defining the Data Source"](#), on page 291 for description of the provided settings.

Remote command:

[PULSe:MOP:DATA:DSRC](#) on page 406

Coding

Coding is a technique used to improve the signal properties and signal reception. In general, the coding schemes are applied before the modulation. The modulation symbols are coded directly before I and Q values or frequency shifts are assigned.

This software supports the Differential and Grey coding and the combination of both.

For more information, see [Chapter E, "Common Coding Algorithms"](#), on page 521.

Remote command:

[PULSe:MOP:DATA:CODing](#) on page 406

Filter

Filters are applied to shape the signal before it is modulated on the RF. The selected filter type and shape affect the output signal, especially while generating broadband signals.

This software supports the following filter types: "Rectangular", "Gauss", "Gauss FSK", "Cosine", "Root Cosine", "Low Pass", "SOQPSK" and "Rectangular (R&S SMW)".

Depending on the selected filter type, one or more filter parameters are provided. You can adjust the filter characteristics to select more steeper edges, to customize the transition bandwidth, or to adjust the filter length.

For more information, see [Chapter F, "Supported Filter and Impact of the Filter Parameters"](#), on page 522.

Remote command:

[PULSe:MOP:FILTer:TYPE](#) on page 409

[PULSe:MOP:FILTer:BT](#) on page 408

[PULSe:MOP:FILTer:BWIDth](#) on page 408

[PULSe:MOP:FILTer:LENGth](#) on page 408

[PULSe:MOP:FILTer:ROLLoff](#) on page 408

7.2.7 Envelope and Modulation Graphs

[Envelope Graph](#)..... 101

[Modulation Graph](#)..... 102

 L [Time domain display](#)..... 103

 L [I/Q constellation diagram](#)..... 103

 L [Spectrum](#)..... 103

Envelope Graph

Displays the current pulse shape depending on the selected [Standard Timing Profile](#) and [Rising/Falling Edge, Width, Rising/Falling Slope](#) settings or [Pulse Envelope Settings](#).

For example, see the envelope graph on [Figure 7-4](#).

"Normalized Pulse Envelope"

Changes the used units.

Available are "Watt", "dBm (W)" and "Voltage"

"Show Timing"

Set "Show Timing > On" to visualize the pulse width, rise and fall time on the graph.

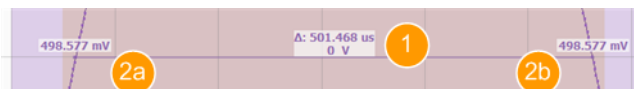
This visualization is useful, if you are not familiar with the definition of the pulse parameters when different profiles are used.

See also [Chapter 7.1, "Basics on Pulse Signals and Pulse Generation"](#), on page 70.

"Temporary delta information"

Drag the mouse cursor on the graph.
The delta information is displayed temporarily above the line.

Example: Measuring the pulse width with the help of delta marker



- 1 = Delta information: upper value is the Δx [s], lower value is the Δy [units depending on the selected "Pulse Envelope Units"]
- 2a, 2b = Start and end y values [units depending on the selected "Pulse Envelope Units"]

Remote command:

`PULSe:PREView:MODE` on page 414

Modulation Graph

Displays the current modulation characteristics depending on the selected [Modulation on Pulse \(MOP\) Settings](#).

The MOP is visualized on three diagrams: the time domain view, the spectrum, and the I/Q constellation diagrams.

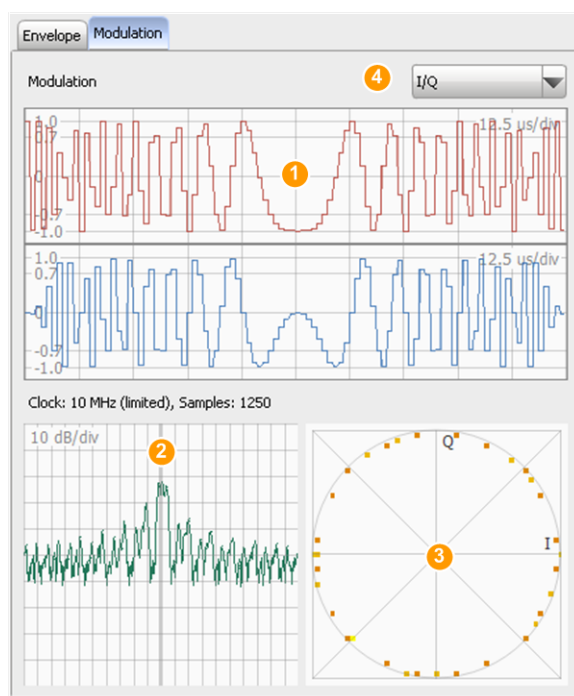


Figure 7-13: Modulation Graph: Understanding the displayed information ("MOP Type > Polyphase code", "Type = P4", "Length M = 10")

- 1 = Time domain view
- 2 = Spectrum
- 3 = I/Q constellation diagram
- 4 = "Modulation > I/Q"

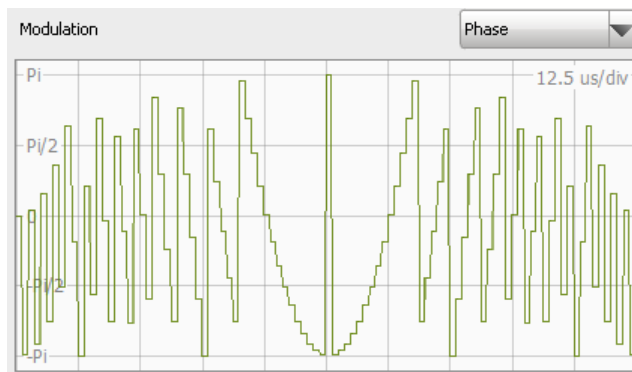
Time domain display ← Modulation Graph

Use the "Modulation" parameter to change the displayed modulation characteristics.

Available are:

"I/Q diagram" Displays the I and Q signal versus time. Both signals use a linear scale in the range between -1.0 and +1.0.

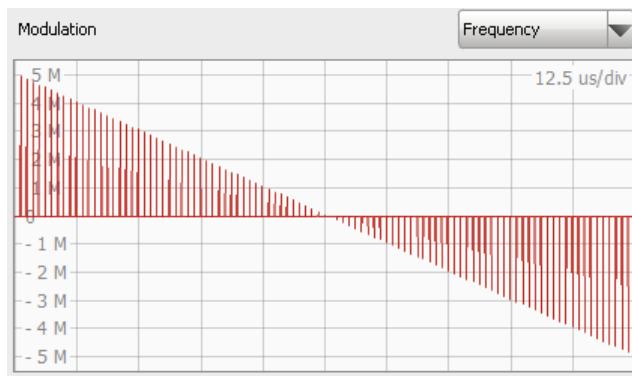
"Phase diagram" Displays the phase angle of the signal versus time. The diagram shows the phase changes in the range $-\pi$ and $+\pi$.



A phase change is equivalent to a rotation at a constant radius in the constellation diagram.

"Frequency diagram"

Displays the instantaneous signal frequency versus time.



Remote command:

[PULSe:PREView:MOP](#) on page 414

I/Q constellation diagram ← Modulation Graph

Displays the probability at which I/Q points occur.

The color scale is logarithmic; the most probable points are indicated with bright yellow color.

Spectrum ← Modulation Graph

Displays the FFT spectrum. The FFT uses a logarithmic scale between 0 dB and -100 dB.

7.3 How to Create a New Pulse and Adjust Its Settings

Pulses are the fundamental building elements of any sequence.

See:

- ["To create a pulse"](#) on page 104
- ["To define the timing parameters of a pulse"](#) on page 104
- ["To define the level parameters of a pulse"](#) on page 105
- ["To define and apply modulation on the pulses \(MOP\)"](#) on page 107
- ["To generate a pulse with raised cosine envelope shape"](#) on page 108
- ["To define your custom envelope shape with an equation"](#) on page 109
- ["To visualize the pulse characteristics"](#) on page 110

To create a pulse

1. Perform one of the following:

- a) Select "Repository Tree > Pulse > New".
- b) In the "Sequence" dialog, press the pulse icon.

A new pulse with default settings is created and added to the repository tree.

New pulses are named `Pulse <n>` where n is a number starting at one.

You can add information to describe and identify the pulse, like a name or a comment.

2. Change the pulse name, e.g. "P1".

A pulse is described by its timing and level parameters, applied modulation on pulse (MOP) and enabled marker signals.

For background information on the pulse parameters, see [Chapter 7.1, "Basics on Pulse Signals and Pulse Generation"](#), on page 70.

For description of the required settings, see:

- [Chapter 7.2.2, "Pulse Timing Settings"](#), on page 74
- [Chapter 7.2.3, "Pulse Level Settings"](#), on page 77
- [Chapter 7.2.5, "Modulation on Pulse \(MOP\) Settings"](#), on page 83
- [Chapter 7.2.4, "Pulse Envelope Settings"](#), on page 80
- [Chapter 7.2.7, "Envelope and Modulation Graphs"](#), on page 101

To define the timing parameters of a pulse

1. In the repository tree, select "Pulse > PulseName e.g. P1 > Timing".
2. Select "Standard Profile > On".
3. Select "Profile > Voltage 10/50/90".
4. Set the timing-related parameters, such as delay, rise, fall, on, and off time, and the shape of the rising and falling slopes.
5. To configure the pulse repetition interval (PRI), use the parameter [PRI](#).

6. Use the "Envelope" graph to visualize the current envelope profile.

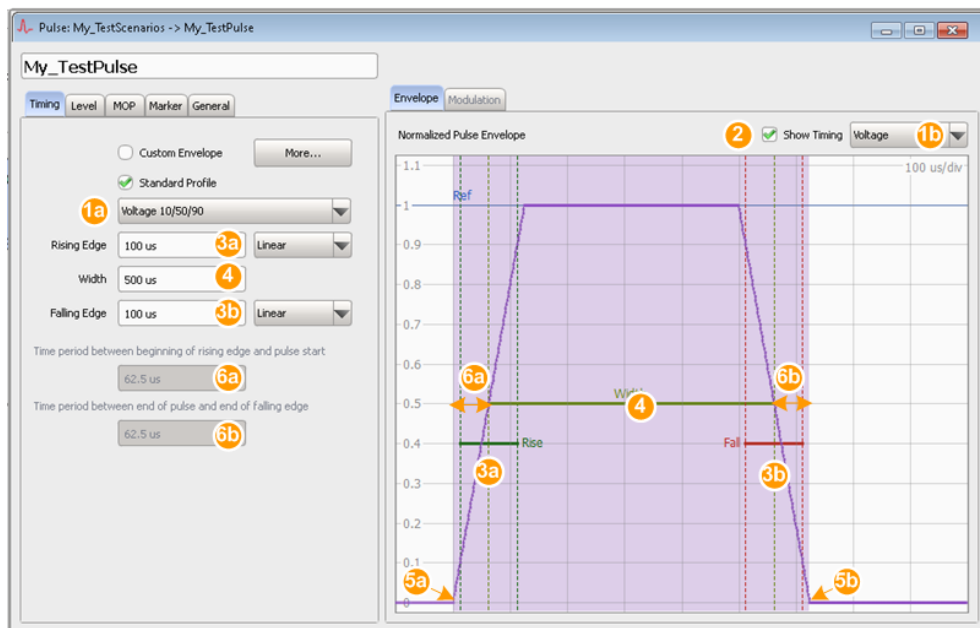


Figure 7-14: Pulse Timing: understanding the displayed information

- 1a = "Standard Profile > Voltage 10/50/90"
- 1b = "Pulse Envelope Units > Voltage"
- 2 = "Show Timing > On" indicates the pulse timing parameters on the envelope graph
- 3a = "Rising Edge = 100 us", i.e. the time it takes the voltage to rise from 10% to 90% of the top level
- 3b = "Falling Edge = 100 us", i.e. the time it takes the voltage to fall from 90% to 10% of the top level
- 4 = "Width = 500 us", i.e. the duration the voltage is above 50% of the top level
- 5a, 5b = Beginning of the rising and falling edge
- 6a = "Time period between beginning of the rising edge and pulse start = 62.5 us"; this is the time it takes the voltage to rise from 0% to 50% of the top level
- 6b = "Time period between end of pulse and the end of falling edge = 62.5 us"; this is the time it takes the voltage to fall from 50% to 0% of the top level

You can also import a custom shape or I/Q data, see:

- ["To generate a pulse with raised cosine envelope shape"](#) on page 108
- ["To define your custom envelope shape with an equation"](#) on page 109



Per default, the pulses within a pulse sequence, that is built by repeating the same pulse, have constant on time and PRI.

You can define IPM profiles to vary the pulse parameters on a pulse-to-pulse basis. See [Chapter 9.2, "How to Create IPM Profiles and Use Them to Vary Pulse Parameters"](#), on page 150.

To define the level parameters of a pulse

Pulse level settings control the RF output power level during all pulse phases. The level is determined by two main settings, the "Top Power" and the "Base Power".

To access the level settings:

1. In the repository tree, select "Pulse > PulseName > Level".

The tab comprises the pulse level settings, required to simulate a non-ideal pulse. The major reasons for non-ideal pulse shapes are the overshoot and ripple.

2. Change the "Level" settings and use the [Envelope Graph](#) to visualize the pulse characteristics.

The "Envelope Graph" confirms that the envelope overshoots the "Top Power", experiences damped oscillations and amplitude "Droop" and settles to the final value.

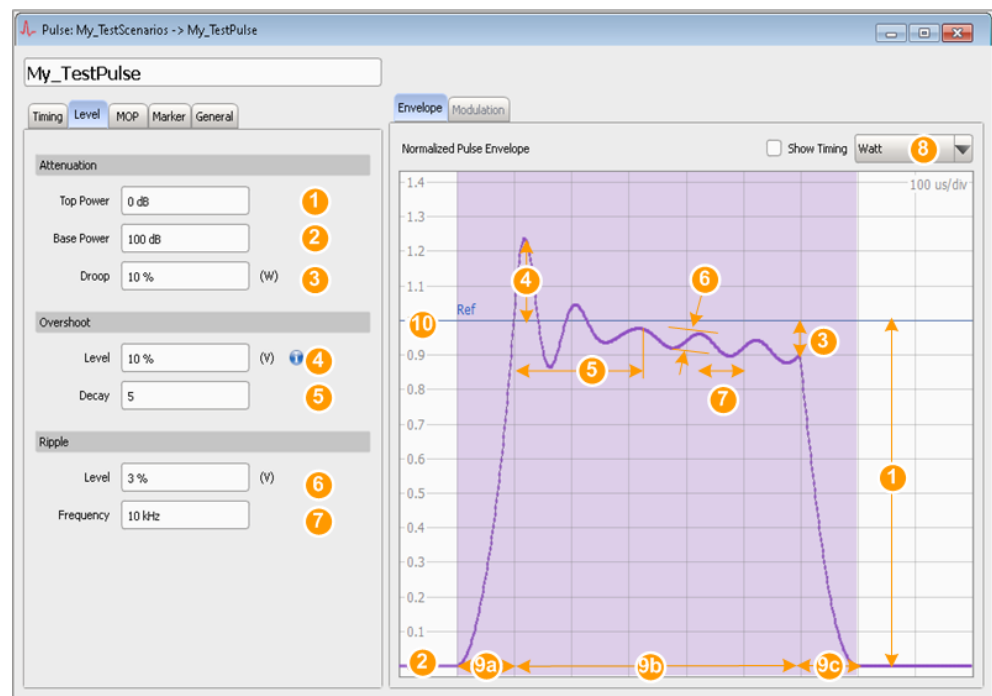


Figure 7-15: Pulse Level: understanding the displayed information

- | | |
|---------|---|
| 1 | = Pulse top power corresponds to 1000 mW ("Pulse Envelope = Watt"), i.e. little or no attenuation ("Attenuation Top Power = 0 dB") |
| 2 | = "Base Power", i.e. high attenuation; typically 100 dB |
| 3 | = Amplitude change from the beginning to the end of the pulse ON time |
| 4 | = Overshoot level, in Volt (diagram uses Watts) |
| 5 | = Damping constant that influences the time it takes the overshoot to decay |
| 6 | = Amplitude of the oscillating signal, in Volt (diagram uses Watts) |
| 7 | = Signal oscillating with the selected frequency ("Frequency = 10 kHz", i.e. the oscillation period is 100 us) |
| 8 | = Pulse amplitude units (Y-axis units) |
| 9a, 9b, | = Pulse timing: "Profile > Standard 0/100", "Rising Edge = Falling Edge = 100 us", "Width = |
| 9c | 500 us" (see Standard Timing Profile) |
| 10 | = Reference value used by the calculation of the pulse level parameters: "Ref. = 0 dB (1 Watt)"; the value is set with the parameter Ref. Level |

To define and apply modulation on the pulses (MOP)

The "MOP" settings define the intra-pulse modulation, i.e. the modulation applied on the pulse rise, on, and off time. The R&S Pulse Sequencer provides a wide range of commonly used modulation schemes, such as AM, FM, or Chirp. You can also add your custom pulse content, [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.

Some "MOP" require data sources that are used with a modulation scheme. A data source for the symbols of the modulation has to be created first.

See [Chapter 18, "Defining the Data Source"](#), on page 291 for description of the provided settings.

To access the "MOP" settings:

1. In the repository tree, select "Pulse > PulseName > MOP".
2. Select "MOP > Settings".

The available modulation settings depend on the selected "MOP Type".

See [Chapter 7.2.6, "Built-In Modulation Types and their Settings"](#), on page 86.

3. Select for example "MOP > MOP Type > Chirp > Linear Chirp".

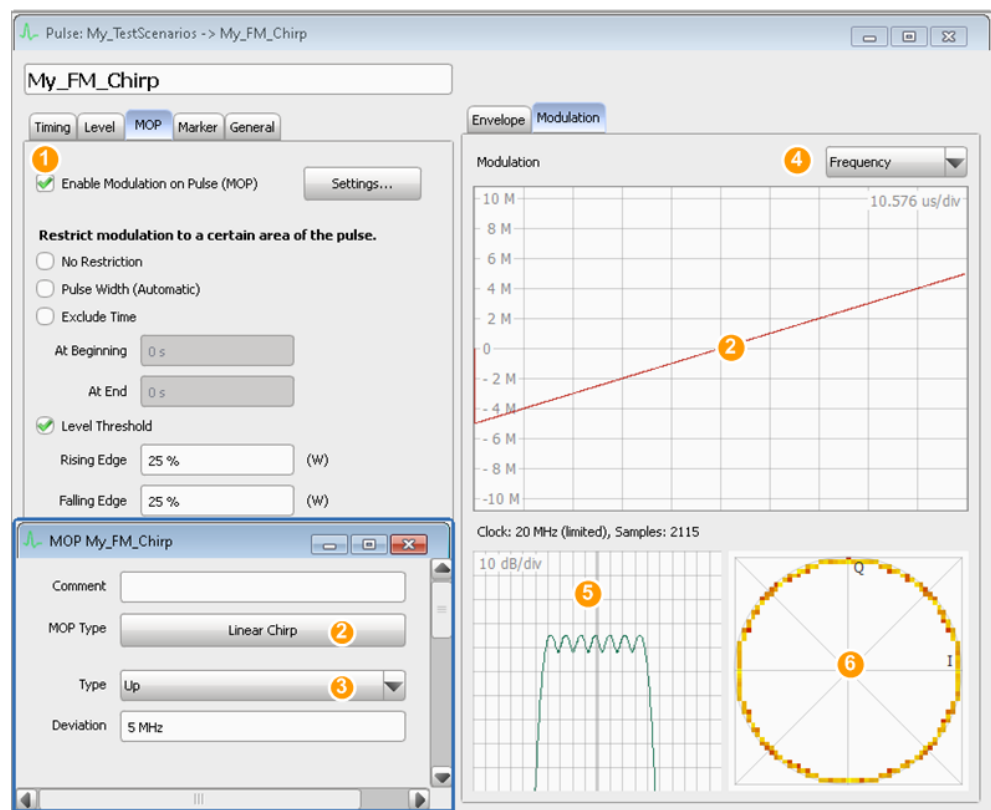


Figure 7-16: Modulation on pulse (MOP): understanding the displayed information

- 1 = "Enable Modulation on Pulse (MOP) > On", i.e. MOP is applied
- 2 = Enabled is a linear chirp modulation; the diagram confirms the frequency variation
- 3 = "Type > Up", i.e. the frequency is increasing linearly

- 4 = Frequency diagram
- 5 = Spectrum
- 6 = Constellation diagram

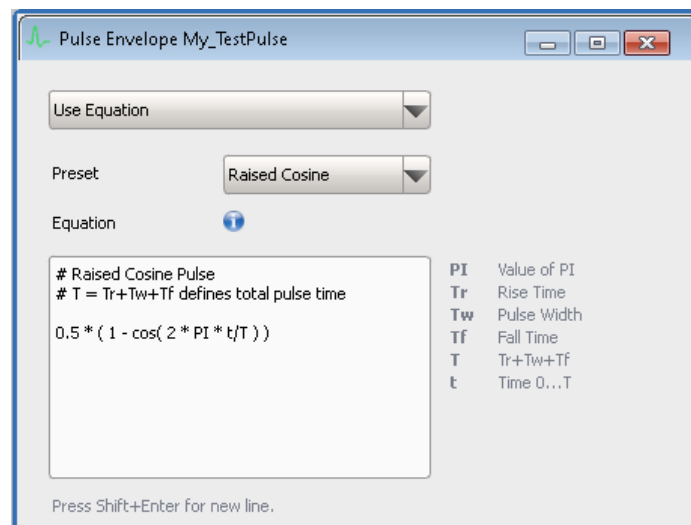
For a description of the provided settings, see [Chapter 7.2.5, "Modulation on Pulse \(MOP\) Settings"](#), on page 83.

4. Select "Enable Modulation on Pulse (MOP) > On".
5. Use the [Restrict modulation to a certain area of the pulse](#) settings to determine the exact pulse part the modulation is applied to, e.g.:
 - a) Select "Level Threshold > On".
 - b) Select "Rising Edge = Falling Edge = 25%".

Use the "Waveform View" dialog to observe the effect of the enabled modulation, see [Chapter 16, "Visualizing and Analyzing Waveforms"](#), on page 264.

To generate a pulse with raised cosine envelope shape

1. Select "Pulse > Timing"
2. Select "More" to customize the envelope.
3. Select "Use Equation".
Envelope is described by an equation, as selected with the parameter "Preset".
4. Select "Preset > Raised Cosine"



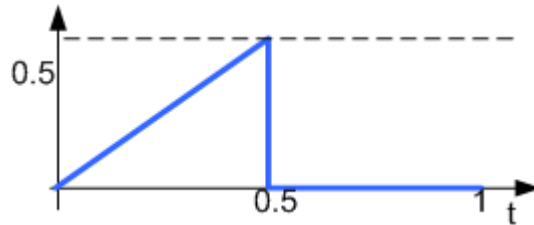
The dialog displays the used equation.

For overview of the available envelope shapes, see ["Equations and parameters used to define custom pulse envelopes"](#) on page 493.

5. Select "Pulse > Timing > Custom Envelope > On" to apply the envelope.
The [Envelope Graph](#) displays the resulting envelope.

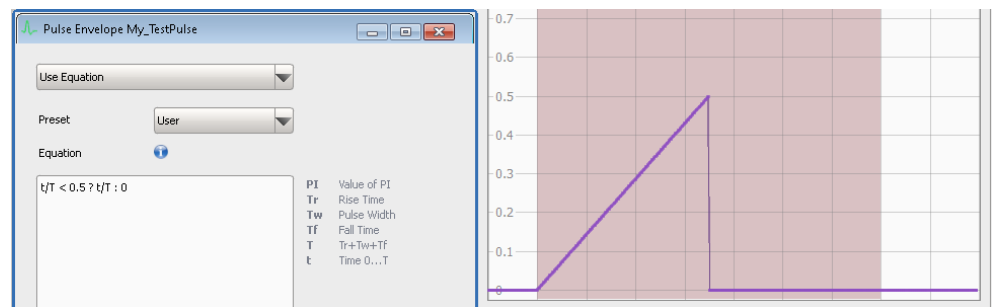
To define your custom envelope shape with an equation

This example explains how to enable an envelope shape that follows the shape on the following figure.



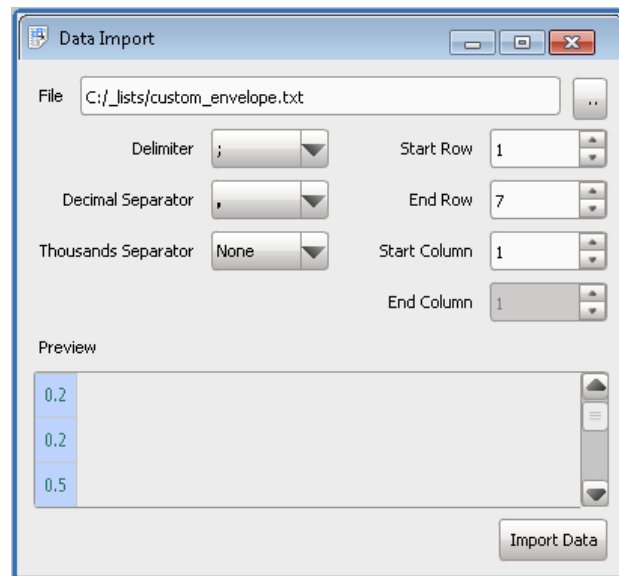
Equations are related to Voltage. Allowed are amplitude values from 0 through 1.

1. Select "Pulse > Timing > More" and select "Use Equation".
2. Select "Preset > User" and enter the equation:
 $t/T < 0.5 ? t/T : 0$
 See also [Chapter B, "Formula Syntax"](#), on page 492.
3. Select "Envelope > Normalized Pulse Envelope > Voltage" and compare the resulting shape.



To generate an envelope shape in a table form

1. Select "Pulse > Timing > More" and select "Use Imported Data".
2. Select "Import Data from File", navigate to the file describing the envelope and load it.



To visualize the pulse characteristics

1. Use the "Modulation Graph" to visualize the configured modulation schemes. See ["Modulation Graph"](#) on page 102.
2. Use the "Envelope Graph" to display the envelope shape. See [Figure 7-14](#).
3. Set "Show Timing > On" to visualize the pulse width, rise and fall time on the graph.

8 Building Pulse Sequences

This section explains the sequencing principles and signal processing, like creating loops and defining overlaying signals.

For information on sequences containing waveforms, see [Chapter 13, "Working with Waveforms and Generating Interfering Signals"](#), on page 219.

8.1 About the Sequencing Principles

Sequences are described in table form where the rows are processed sequentially and appended to the final waveform, see [Figure 8-1](#).

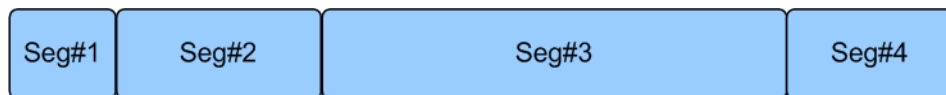


Figure 8-1: Default segments processing: One segment after the other

Each row is repeated once or as defined with the repetition count parameter (see ["Rep. Cont."](#) on page 117).

See also:

- ["To define and enable pulse repetition"](#) on page 125
- [Chapter 8.2.2, "Pulse Repetition Settings"](#), on page 118

Complex sequences can also contain loops, overlays, and fillers:

- **Loop**

A loop spans one or more rows (pulses) and processes these rows in the same order several times. Nested loops are complex loops that span rows and loops. The loop duration is defined by the loop repetition number. The loop repetition number is a fixed value or a randomly selected value within a value range, defined with its minimum, maximum and step values.

See also:

- ["To define and apply a loop"](#) on page 126
- [Chapter 8.2.3, "Loop Settings"](#), on page 119

- **Overlay**

An overlay processes rows in parallel. The row processing start at the same time and lasts up to the overlay duration is. Longer segments are truncated.

Overlay is a method that changes the default behavior ([Figure 8-1](#)) and adds several pulses (segments) to a common time line; that is segments are stacked on top of each other. An overlay group spans several segments of any kind, like regular pulses and fillers.

Segments are added to the existing data starting at the beginning of the first segment. If the new segment is shorter than the overlay length, the remainder is left blank. Longer segments are truncated, see [Figure 8-2](#).

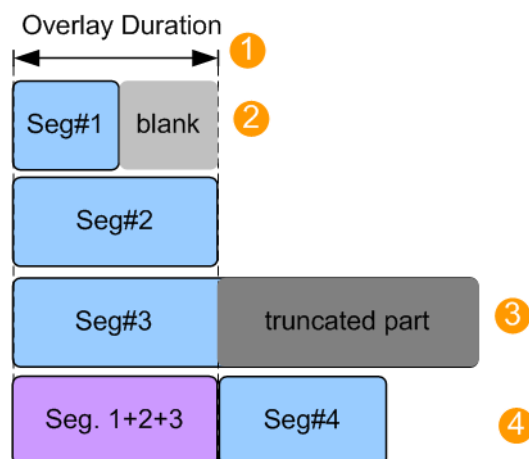


Figure 8-2: Segments processing in overlay mode: Segments are added

1 = Overlay Duration: sets the length of the resulting sum segment

2 = Short segment: remainder is left blank

3 = Long segment: segment is truncated

4 = Resulting sum segment (Seg#1+Seg#2+Seg#3): Seg#4 is processed right after the overlay duration elapses

See also:

- ["To define and enable overlaying segments"](#) on page 129
- ["To include a subsequence in an existing overlay"](#) on page 131
- [Chapter 8.2.4, "Overlay Settings"](#), on page 121

- **Subsequence**

A subsequence defines a sequential sequence of segments within an overlay.

See also:

- ["To include a subsequence in an existing overlay"](#) on page 131

- **Filler**

Fillers are unmodulated continuous wave (CW) or blank signals used to fill in certain period in a sequence.

There are two ways to set the duration of a filler: by defining the time duration to be filled in (total duration) or by defining of a certain moment of time (time synchronization).

The filler duration can be a fixed value or defined as an equation. If the filler is configured within a loop, the equation uses the loop variables.

See also:

- ["To create and add a CW segment"](#) on page 128
- [Chapter 8.2.5, "Fillers Settings"](#), on page 121

8.2 Sequence Settings

Access:

- In the repository tree, select "Sequence > New".

The screenshot shows the 'Sequence Settings' window for 'My_TestSequence'. It includes a configuration section with 'Sequence Type' set to 'Pulse-Based' and 'Phase Mode' set to 'Absolute'. Below is a 'Sequence Description' table with 6 rows. The 'Block Diagram' section shows a graphical representation of the sequence flow: 1: Overlay (0, 5.0 ms) leads to 2: Pulse (P1, 2x, 1.0 ms PRI), which then branches to 3: Sub and 4: Filler (CW, 5.0 ms). From 3: Sub, the flow goes to 5: Pulse (P1, 5x, 1.5 ms PRI), which then leads to 6: Pulse (P2, 2x, 2.5 ms PRI).

No	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
1		Overlay		...		1 2 3 4 ...					
2		Pulse	P1	2	Static	1 2 3 4 ...	0 Hz	0 dB	0°	1 ms	0 s
3		Sub Seq.		1		1 2 3 4 ...					
4		Filler		...		1 2 3 4 ...					
5		Pulse	P1	5	Static	1 2 3 4 ...	0 Hz	0 dB	0°	1.5 ms	0 s
6		Pulse	P2	2	Static	1 2 3 4 ...	0 Hz	0 dB	0°	2.5 ms	0 s

Figure 8-3: Sequence Settings

- 1 = Sequence name, comment, type and phase mode
- 2 = Sequence description table, where each row represents one item
- 3 = Interactive graphical representation of the current sequence, displayed if "Block Diagram > On"

For step-by-step descriptions, see [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124.

Settings:

- [Sequence Description Settings](#)..... 114
- [Pulse Repetition Settings](#)..... 118
- [Loop Settings](#)..... 119
- [Overlay Settings](#)..... 121
- [Fillers Settings](#)..... 121
- [Lists with Multiple Sequences](#)..... 123

8.2.1 Sequence Description Settings

Sequence Name, Comment, Type, Sequence Description Table, Block Diagram.....	114
Phase Mode.....	114
New, Insert, Append, Remove, Clear Items, Undo/Redo.....	116
Select, No.....	116
Nesting.....	116
Type.....	116
Pulse/Waveform.....	116
Rep. Cont.....	117
... (more).....	117
IPM.....	117
Marker.....	117
Δ Freq.....	117
Δ Level.....	117
Phase.....	117
PRI.....	118
Delay.....	118


Sequence Name, Comment, Type, Sequence Description Table, Block Diagram

A sequence is defined with its unique name, an optional description and with information on the sequence content, waveforms, or pulses.

A sequence is defined in table form where each row represents one item.

If enabled, an interactive block diagram displays the current sequence configuration.

- Left-click to select the item (see "Select, No." on page 116).
- Right-click to access the context menu (see "New, Insert, Append, Remove, Clear Items, Undo/Redo" on page 116).
- A double click on an item works like the "More (...)" icon in the "Sequence Description Table" and accesses the settings of the item directly.

Tip: Use the split window icon  to detach the diagram and open it in a separate window.

Remote command:

- SEquence:CREate on page 329
- SEquence:CATalog? on page 329
- SEquence:SElect on page 330
- SEquence:NAME on page 331
- SEquence:COMMeNt on page 331
- SEquence:TYPE on page 448
- SEquence:REMove on page 331

Phase Mode

Defines how the phase is set at each pulse start.

"Absolute" The pulse start phase is set to value defined in the sequence description table (Phase) or by the IPM.

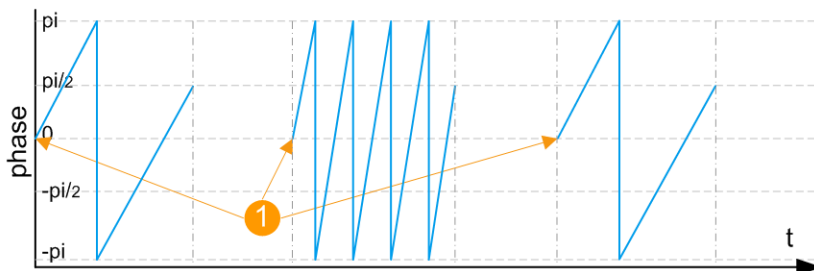


Figure 8-4: "Phase Mode = Absolute" and "Sequence Description (table) > Phase = 0 deg" (IPM not defined)

1 = Start phase of all pulses is 0 deg

"Continuous (Relative)" The phase defined in the sequence description table (Phase) or by IPM is used as a relative phase change. The phase of the last valid sample of a pulse is held; the specified phase value is applied as a phase offset on it.

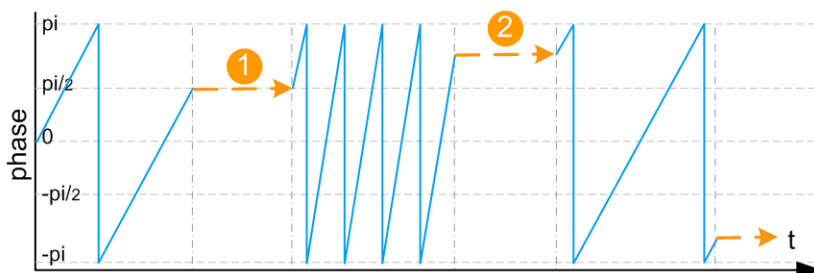


Figure 8-5: "Phase Mode = Continuous (Relative)" and "Sequence Description (table) > Phase = 0 deg" (IPM not defined)

1, 2 = With "Phase = 0 deg", the phase offset is also 0 deg. The phase of last sample of a pulse is also the start phase of the next one

"Memory" If there are frequency steps, the phase is set to a state as if the frequency has never changed.

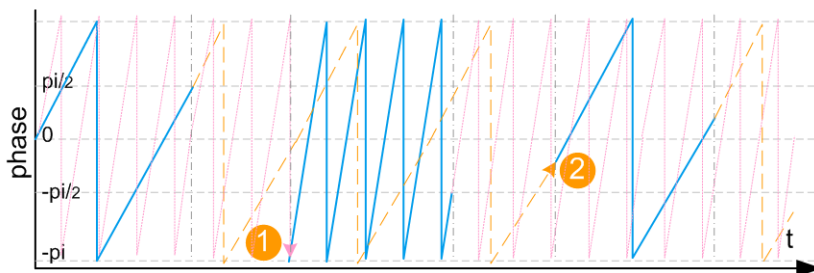


Figure 8-6: "Phase Mode = Memory" and "Sequence Description (table) > Phase = 0 deg" (IPM not defined)

1, 2 = Alternating switching between the outputs of frequency synthesizers (2 signals in the example). At the beginning of each pulse, the phase is set to a state the frequency would have had if it had been continuously running

Remote command:

[SEquence: PHASe: MODE](#) on page 448

New, Insert, Append, Remove, Clear Items, Undo/Redo

Standard functions for items handling, see "[Standard function in the context menus](#)" on page 24.

Select a row for example to delete it or to insert a row before it.

Remote command:

[SEquence: ITEM: ADD](#) on page 332

[SEquence: ITEM: CLear](#) on page 335

[SEquence: ITEM: DElete](#) on page 335

Select, No.

Select an item to perform any row-based actions.

Items are indicated by consecutive number; also used for indication in remote control.

Remote command:

[SEquence: ITEM: SElect](#) on page 333

[SEquence: ITEM: COUNT?](#) on page 333

Nesting

Indents table rows to include them in a loop, to assign them to an overlay or build parallel branches.

See "[To define and enable overlaying segments](#)" on page 129.

Remote command:

[SEquence: ITEM: INDent](#) on page 449

Type

Defines the type of content of the current item. Available are:

- | | |
|--------------|--|
| "Pulse/Wave" | To select a pulse or a waveform, use the Pulse/Waveform . |
| "Filler" | Creates dummy data, like a blank segment, or a CW signal.
See Chapter 8.2.5, "Fillers Settings" , on page 121. |
| "Loop" | Creates a loop
See Chapter 8.2.3, "Loop Settings" , on page 119. |
| "Overlay" | An overlay processes sequence items in parallel
See Chapter 8.2.4, "Overlay Settings" , on page 121. |
| "Sub. Seq" | In overlay, includes a sequence in a sequence, i.e. creates a parallel branch containing a sequence of items.
See To include a subsequence in an existing overlay . |

Remote command:

[SEquence: ITEM: TYPE](#) on page 454

Pulse/Waveform

Lists all available pulses and waveforms.

- To access the settings of an existing pulse, select "... (more)".
- To create a new pulse, select the pulse icon.

See also [Chapter 7, "Creating a Pulse Library"](#), on page 70.

Remote command:

[SEquence:ITEM:PULSe](#) on page 452

[SEquence:ITEM:WAVeform](#) on page 452

Rep. Cont.

Sets the number of times the item is repeated. For more complex repetition patterns, select "... (more)".

See [Chapter 8.2.2, "Pulse Repetition Settings"](#), on page 118.

Remote command:

[SEquence:ITEM:REP:COUNT:FIXed](#) on page 453

[SEquence:ITEM:LOOP:COUNT:FIXed](#) on page 450

... (more)

Accesses a dialog with further settings.

IPM

Enables and defines an inter-pulse modulation.

See [Chapter 9, "Defining and Enabling Inter-Pulse Modulation Effects"](#), on page 134.

Marker

Defines the active markers per pulse phase, see [Chapter 20, "Defining and Enabling Marker Signals"](#), on page 301.

Δ Freq.

Enables a frequency offset relative to the reference frequency.

Frequency-dependent parameter values are defined relative to this value.

See also ["To define and apply a loop"](#) on page 126.

Remote command:

[SEquence:ITEM:FREQuency:OFFSet](#) on page 449

Δ Level

Defines a level offset per row.

A negative level offset simulates a pulse attenuation. For example, a "Level = -3 dB" means that the pulses of the current row are processed with power level that is the half of the power.

Level-dependent parameter values are defined relative to this value.

See also ["To define and apply a loop"](#) on page 126.

Remote command:

[SEquence:ITEM:LEVel:OFFSet](#) on page 450

Phase

Sets a phase offset at the beginning of the row. Phase dependent parameter values are defined relative to this value.

For example, a "Phase = 30 deg" shifts the phase in the first pulse of the current row; enabled modulation would further change the phase. The phase offset is not compensated at the row end.

See also ["Phase Mode"](#) on page 114.

Remote command:

[SEquence:ITEM:PHASe:OFFSet](#) on page 452

PRI

Sets the Pulse Repetition Interval (PRI), i.e. the pulse period (see [Figure 7-1](#)).

Use the "PRI" parameter to define pulses with constant total pulse cycle duration and enabled timing variations like jitter for example.

Remote command:

[SEquence:ITEM:PRI](#) on page 452

Delay

Enables a start delay.

A delay within an overlay shifts the start time of the overlay elements relative to each other. Time-dependent parameter values are defined relative to this value.

See also ["To define and apply a loop"](#) on page 126.

Remote command:

[SEquence:ITEM:PDELay](#) on page 451

8.2.2 Pulse Repetition Settings

Per default, a table row in the sequence description table, i.e. a pulse or a waveform, is processed once. To repeat an item several times, enable a repetition.

Rep: Line Item 1

Repetitions:

Fixed Value
 Randomly Selected

Min: 1
Max: 10
Step: 1

Auto Set by Duration

Duration: 1 ms

Repetition Variables:

Prefix:

Index 1...N <prefix> i
Rep. Count N <prefix> n
Start Time <prefix> t

For a step-by-step instruction, see ["To define and enable pulse repetition"](#) on page 125.

Repetition Number

Defines how many times a pulse, a waveform, or a loop is repeated.

The repetition number can be:

- A fixed value
- A randomly selected value within a value range, defined with its minimum, maximum and step values ("Randomly Selected")
- Defined to fit in a selected duration ("Auto Set by Duration"), where the repetition number can be rounded up or down

Remote command:

[SEquence:ITEM:REP:TYPE](#) on page 454

[SEquence:ITEM:REP:COUNt:MINimum](#) on page 453

[SEquence:ITEM:REP:COUNt:MAXimum](#) on page 453

[SEquence:ITEM:REP:COUNt:STEP](#) on page 453

[SEquence:ITEM:REP:COUNt:DURation](#) on page 452

[SEquence:ITEM:REP:COUNt:ROUNding](#) on page 453

Repetition Variables

You can define repetition variables that are used by the pulse calculation within one row.

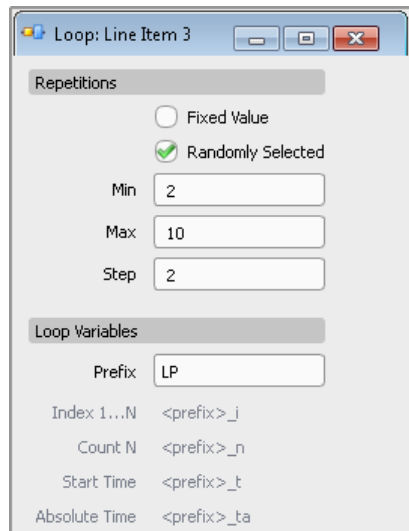
"Prefix" Indicates the repetition the variable belongs to. If empty, no variables are used.
 The variable prefix is a case-sensitive entry.

Remote command:

[SEquence:ITEM:REP:VARiable](#) on page 454

8.2.3 Loop Settings

Per default, table rows in the sequence description table are processed sequentially. To process a subset of pulses or waveforms several times, create a loop.



For a step-by-step instruction, see ["To define and apply a loop"](#) on page 126.

Loop Repetition

The loop repetition number is a fixed value or a randomly selected value within a value range, defined with its minimum, maximum and step values.

Remote command:

[SEquence: ITEM: LOOP: TYPE](#) on page 451

[SEquence: ITEM: LOOP: COUNT: MINimum](#) on page 450

[SEquence: ITEM: LOOP: COUNT: MAXimum](#) on page 450

[SEquence: ITEM: LOOP: COUNT: STEP](#) on page 450

Loop Variables

You can define loop variables that are used by the pulse calculation within the loop.

"Prefix" Indicates the loop to that the variable belongs. If empty, no loop variables are used.
The variable prefix is a case-sensitive entry.

"Loop Index 1 .. N" "<prefix>_j" indicates the current repetition number and is a value in the range 1 .. N, where N is the repetition number.

"Loop Count N" "<prefix>_n" indicates the repetition number, fixed or randomly selected as defined with [Loop Repetition](#).

"Start Time" "<prefix>_t" indicates the start time of each subsequent loop

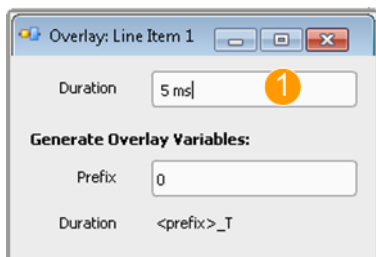
"Absolute Time" "<prefix>_ta" indicates the absolute time

Remote command:

[SEquence:ITEM:LOOP:VARiable](#) on page 451

8.2.4 Overlay Settings

An overlay processes rows in parallel. The row processing starts at the same time and lasts up to the overlay duration is. Longer items are truncated.



1 = Overlay [Duration](#): sets the length of the resulting sum item

For a step-by-step instruction, see "[To define and enable overlaying segments](#)" on page 129.

Duration

Sets the duration of the sum item.

Remote command:

[SEquence:ITEM:OVL:WTIME](#) on page 451

Overlay Variables

You can define overlay variables that are used by the pulse calculation.

"Prefix" Indicates the overlay to that the variable belongs. If empty, no overlay variables are used.
The variable prefix is a case-sensitive entry.

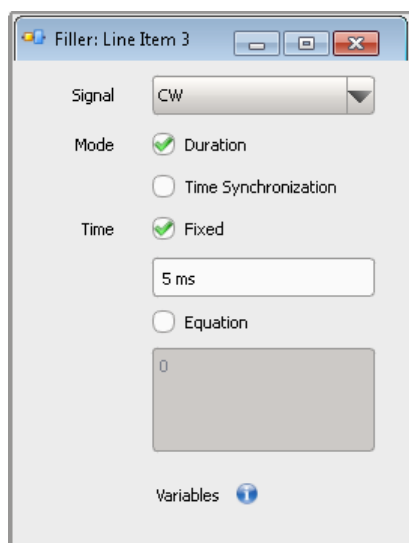
"Duration" "<prefix>_t" indicates the overlay duration.

Remote command:

[SEquence:ITEM:OVL:VARiable](#) on page 451

8.2.5 Fillers Settings

You can enable and add fillers.



For a step-by-step instruction, see ["To create and add a CW segment"](#) on page 128.

Signal

Selects the type of signal to be generated for filling in the dummy segment.

- "Blank" A blank segment with the selected duration is appended to the sequence.
- "CW" A segment containing a continuous wave (CW), i.e. unmodulated signal, is used.
- "Hold" Holds the last available pulse sample.

Remote command:

[SEquence:ITEM:FILLer:SIGNal](#) on page 448

Mode

Filling segments are commonly used for synchronization purposes. This parameter defines the way the filler duration is determined:

- "Duration" Enables a dummy signal for a given period of time.
- "Time Synchronization"
 In this mode, signal data is generated up to the defined point in time. If the selected time has already elapsed, an error is generated and the calculation is stopped.
 Use this mode if a signal should start at a defined point in time and the length of the previous signals vary or is not known.

Remote command:

[SEquence:ITEM:FILLer:MODE](#) on page 448

Time

Defines the timing of the filler.

- "Fixed" Uses a fixed time
- "Equation" Defines the time point as an equation.

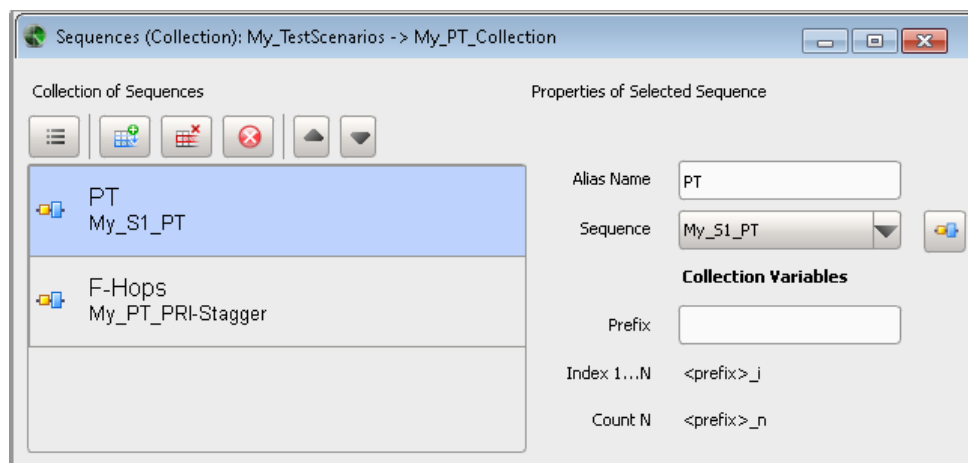
Remote command:

[SEquence:ITEM:FILLer:TIME](#) on page 448

[SEquence:ITEM:FILLer:TIME:FIXed](#) on page 449

[SEquence:ITEM:FILLer:TIME:EQUation](#) on page 449

8.2.6 Lists with Multiple Sequences



Switching between multiple sequences is possible in the pulse train sequential scenario.

List of sequences

Displays a list of sequences.

To add a sequence, select the "Data Source" icon and select "Insert After".

To reorder sequences (items in the list), use the "Up/Down" functions.

Use the standard "Append", "Remove Last", or "Delete All" functions to add or remove a sequence. The provided settings are self-explanatory.

Remote command:

[SCENario:CSEquence:ADD](#) on page 332

[SCENario:CSEquence:SElect](#) on page 333

[SCENario:CSEquence:CURRent](#) on page 434

[SCENario:CSEquence:CLEar](#) on page 335

[SCENario:CSEquence:DElete](#) on page 335

Name

Enters a name.

Remote command:

[SCENario:CSEquence:ALias](#) on page 433

Sequence

Select a sequence from the list of available sequences.

Remote command:

[SCENario:CSEquence](#) on page 441

Collection Variables

You can define collection variables and use them e.g. in the formula editors, in the reports, etc.

"Prefix" Indicates the sequence to that the variable belongs. If empty, no variables are used.
The variable prefix is a case-sensitive entry.

"Index 1 .. N" "<prefix>_i" indicates the current number and is a value in the range 1 .. N.

"Count N" "<prefix>_n" is the number of variables.

Remote command:

[SCENario:CSEquence:VARiable](#) on page 441

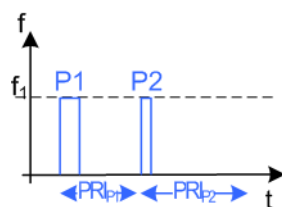
8.3 How to Create Sequences and Use the Control Elements

See:

- ["To create a simple sequence"](#) on page 124
- ["To create and add a CW segment"](#) on page 128
- ["To define and enable pulse repetition"](#) on page 125
- ["To define and enable overlaying segments"](#) on page 129
- ["To include a subsequence in an existing overlay"](#) on page 131
- ["To define and apply a loop"](#) on page 126
- ["To create a scenario with several pulse train sequences"](#) on page 132

To create a simple sequence

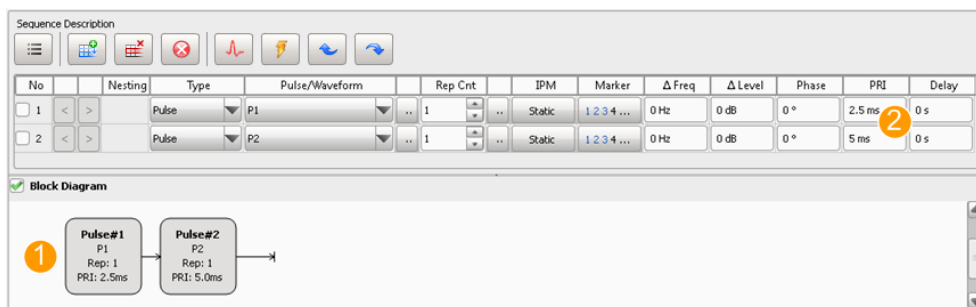
This example explains how to create a simple sequence with two pulses, P1 and P2.



We assume that the required pulses have been created, see [Chapter 7.3, "How to Create a New Pulse and Adjust Its Settings"](#), on page 104.

1. In the repository tree, select "Sequence > New".
2. In the "Sequence" dialog, enter a "Name" and a "Comment", e.g. `MY_S1_PT`.

3. For each sequence item, perform the following steps:
 - a) Select the "Append New Item" icon.
 - b) In the "Sequence Description" table, select "Type > Pulse" and select an existing pulse, e.g. "Pulse > P1".
 - c) If the list is empty, select the pulse icon and create a pulse first.
 - d) See [Chapter 7.3, "How to Create a New Pulse and Adjust Its Settings"](#), on page 104
 - e) Set the PRI_{P1} , e.g. "PRI = 2.5 ms".
4. Enable "Show Block Diagram".



1 = Sequence with two items
 2 = Pulse with different PRIs, i.e items with different duration

The items are processed once and one after the other (see also [Figure 8-1](#)). You can change the default processing, see:

- ["To define and enable pulse repetition"](#) on page 125
- ["To define and enable overlaying segments"](#) on page 129
- ["To define and apply a loop"](#) on page 126

To define and enable pulse repetition

This example explains how to repeat pulses several times, for example to configure a pulse sequence as illustrated on [Figure 8-7](#).

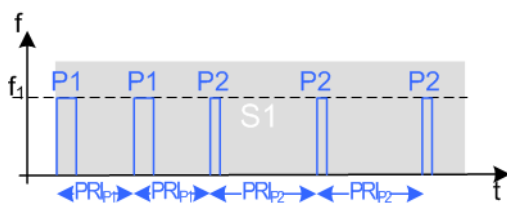


Figure 8-7: Example of pulse sequence composed of 2xP1 and 3xP2, where the pulses use different PRIs

$PRI_{P1} = 2.5 \text{ ms}$
 $PRI_{P2} = 5 \text{ ms}$

We assume that a sequence with two items has been created, see ["To create a simple sequence"](#) on page 124.

1. Open this sequence.
2. In the "Sequence Description" table, for the first pulse, select "Rep. Count = 2"

3. For the second item, set "Rep. Count = 3".

No	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
1	< >	Pulse	P1	2	Static	1 2 3 4 ...	0 Hz	0 dB	0°	2.5 ms	0 s
2	< >	Pulse	P2	3	Static	1 2 3 4 ...	0 Hz	0 dB	0°	5 ms	0 s

The pulse sequence S1 consists of 5 pulses and has a $PRI_{S1} = 20$ ms.



Use the "Waveform View" display to visualize the resulting waveform, see [Chapter 16, "Visualizing and Analyzing Waveforms"](#), on page 264.

To define and apply a loop

This example explains how to create a loop with pulses and to repeat it several times, for example to configure a pulse sequence as illustrated on [Figure 8-8](#).

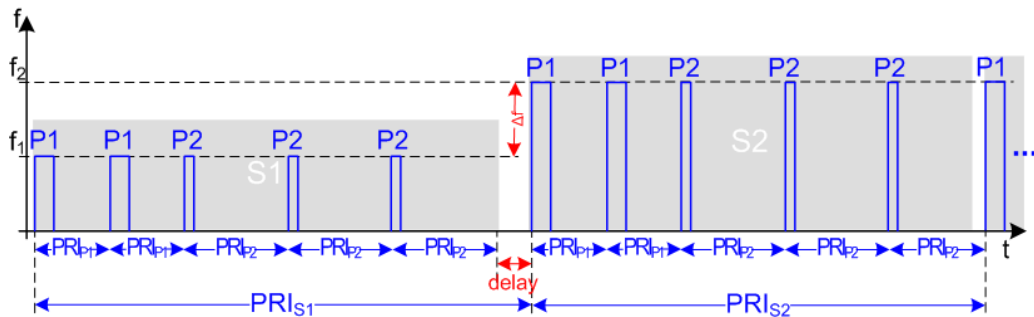


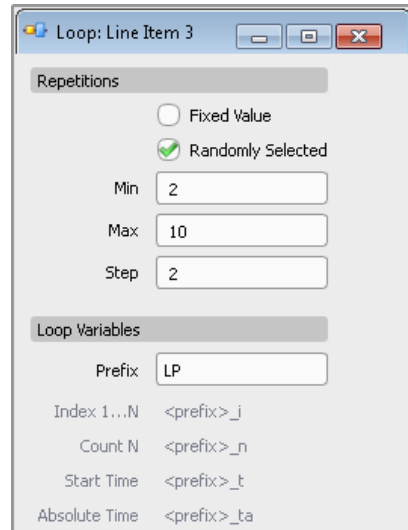
Figure 8-8: Example of pulse sequence composed of two subsequences, S1 and S2

S1, S2 = Both sequences follow the same pulse pattern (2xP1 and 3xP2)
 $PRI_{S1} = PRI_{S2} + \text{"Delay"}$
 $f_2 = f_1 + \text{"ΔFreq"}$

We assume that a sequence with four items has been created, as shown on the following figure.

1. In the "Sequence Description" table:

- select the third item and select "Insert Item Before Selection"
- Select "Type > Loop"
- select the "... (more)" icon and enable a random loop repetition.



The loop is repeated 2, 4, 6, 8 or 10 times.

- to include the last two items (Line items 3 and 5) in the loop, select the "Indent" icon.

No.	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
		Pulse	P1	2	Static	1 2 3 4 ...	0 Hz	0 dB	0 °	2.5 ms	0 s
		Pulse	P2	3	Static	1 2 3 4 ...	0 Hz	0 dB	0 °	5 ms	0 s
		Loop		var							
		Pulse	P1	2	Static	1 2 3 4 ...	500 kHz	-10 dB	0 °	2.5 ms	2 ms
		Pulse	P2	3	Static	1 2 3 4 ...	500 kHz	-10 dB	0 °	5 ms	0 s

The sequence graph in the "Sequence Description" confirms that the two items are included in the loop. The loop is repeated random number of times.

- To enable a different PRI for the looped items, select the first pulse in the loop and set "Delay = 2 ms".
 $PRI_{S1} = PRI_{S2} + 2 \text{ ms} = 22 \text{ ms}$.
- For the pulses in the loop, set the "Δ Freq = 500 KHz" and "Δ Level = -10 dB".
- Assign the sequence `My_S2_S1-F1_S1-F2` to a pulse train scenario, start the waveform calculation, and select "Volatile > View".

The "Data View" confirms that the waveform comprises of two subsequences with different PRI, where the second one is repeated four times.

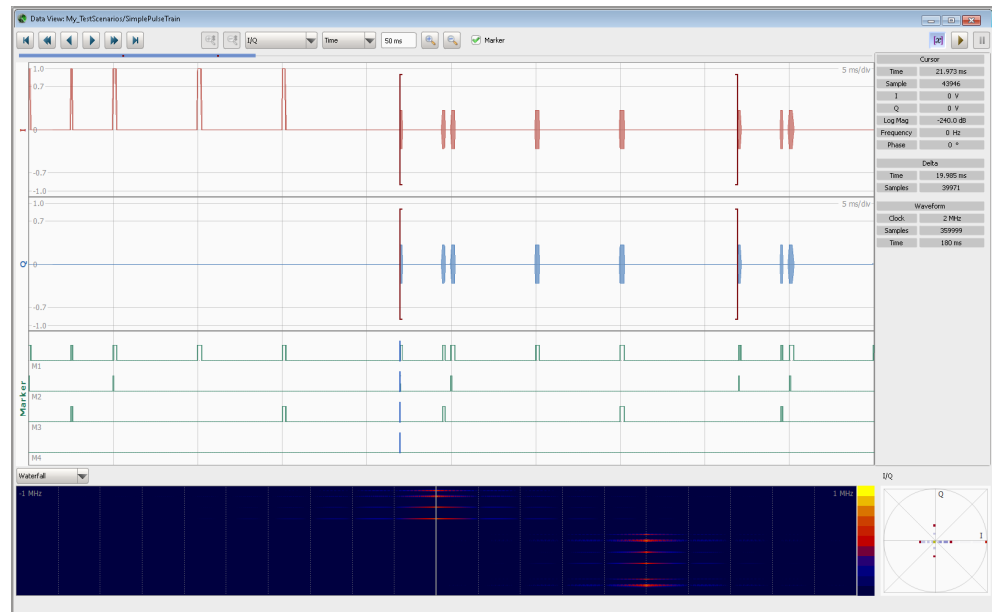


Figure 8-9: Data View: Example of a sequence

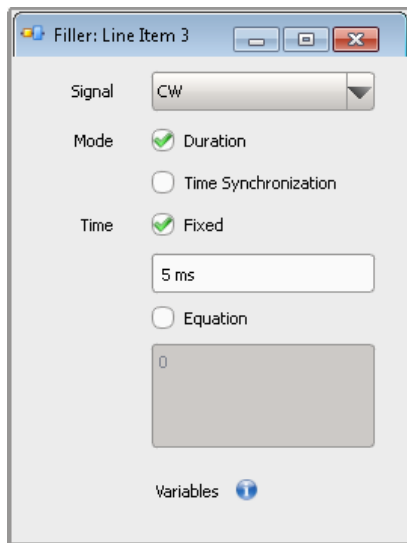
Select "View Mode > Frequency" and "View Mode > Average Power" to observe the frequency and level values.

For more information, see [Figure 16-1](#).

To create and add a CW segment

1. Open an existing sequence.
2. In the "Sequence Description" table, add an item, select "Type > Filler", and select "... (more)".
3. In the "Configure Filler" dialog:
 - a) select "Signal > CW"
 - b) select "Mode > Duration"

c) select "Time > Fixed = 5 ms"



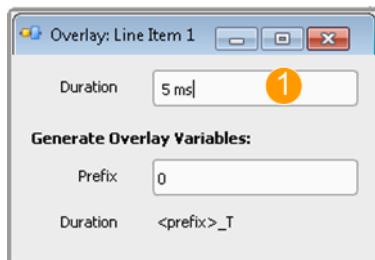
To define and enable overlaying segments

This example explains how to overlay three items (see [Figure 8-2](#)).

We assume that a sequence with four items has been created.

No		Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
<input type="checkbox"/>	1	< >	Pulse	P1	2	Static	1 2 3 4 ...	0 Hz	0 dB	0 °	1 ms	0 s
<input type="checkbox"/>	2	< >	Filler									
<input type="checkbox"/>	3	< >	Pulse	P1	5	Static	1 2 3 4 ...	0 Hz	0 dB	0 °	1.5 ms	0 s
<input type="checkbox"/>	4	< >	Pulse	P2	2	Static	1 2 3 4 ...	0 Hz	0 dB	0 °	2.5 ms	0 s

1. In the "Sequence Description" table of this sequence:
 - a) Select the first item
 - b) Select the "New/Select" icon and select "Insert Item Before Selection"
 - c) For the appended (first item), select "Type > Overlay" and select "... (more)".
2. In the "Configure Overlay" dialog, select "Overlay > Duration = 5 ms".



1 = Overlay **Duration**: sets the length of the resulting sum item

The block diagram confirms that an overlay is created. There are still no segments added to it.

Sequence Description

No	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
		Overlay		...							
		Pulse	P1	2	...	1 2 3 4 ...	0 Hz	0 dB	0°	1 ms	0 s
		Filler		...							
		Pulse	P1	5	...	1 2 3 4 ...	0 Hz	0 dB	0°	1.5 ms	0 s
		Pulse	P2	2	...	1 2 3 4 ...	0 Hz	0 dB	0°	2.5 ms	0 s

Block Diagram

- To include the first three items (Line items 2, 3 and 4) in the overlay, select the "Indent" icon.

Sequence Description

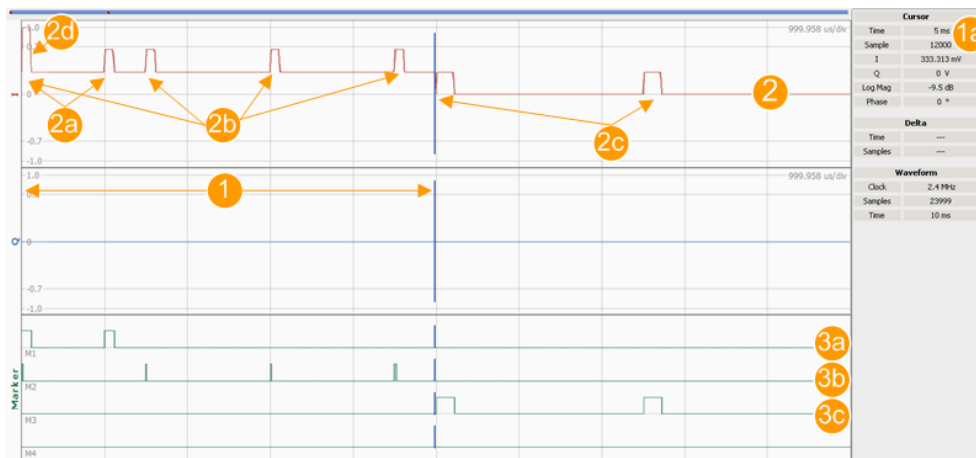
No	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
		Overlay		...							
		Pulse	P1	2	...	1 2 3 4 ...	0 Hz	0 dB	0°	1 ms	0 s
		Filler		...							
		Pulse	P1	5	...	1 2 3 4 ...	0 Hz	0 dB	0°	1.5 ms	0 s
		Pulse	P2	2	...	1 2 3 4 ...	0 Hz	0 dB	0°	2.5 ms	0 s

Block Diagram

The block diagram confirms that the three items are overlaid; the fourth item will be processed after elapsing of the overlay duration.

- To observe the effect of an enabled overlay, open the "Data View" dialog.

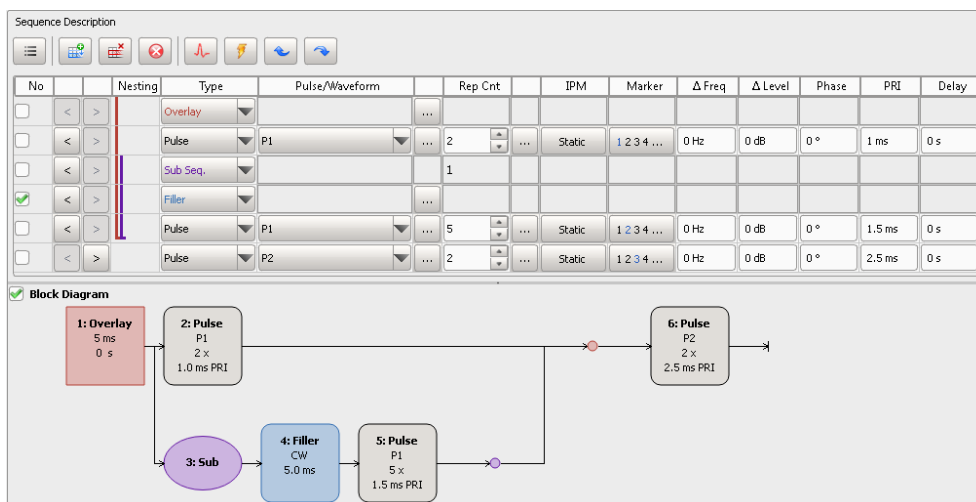
See also [Chapter 16.3, "How to Analyze the Content of Waveform Files and Files with I/Q Data"](#), on page 269.



- 1 = Overlay
- 1a = Overlay "Duration = 5 ms"
- 2 = Generated pulse sequence = sum segment followed by the item outside of the overlay
- 2a+2b = Resulting sum segment
- 2a = Short segment: 2x pulse P1 with "PRI = 1 ms" (remainder is left blank)
- 2b = Long segment: 4x pulse P1 with "PRI = 1.5 ms" (segment is truncated)
- 2c = Outside segment: 2xP2 (appended right after the overlay)
- 2d = The first P1 (short segment) and the first P2 (long segment) are superimposed
- 2a, 3b, 3c = Marker traces ("M1 = M2 = M3 = All"): indicate the pulse On time of each of the pulse

To include a subsequence in an existing overlay

1. Open the sequence you have created with ["To define and enable overlaying segments"](#) on page 129.
2. In the "Sequence Description" table, insert a new item after the second one and select "Type > Sub Sequence".
3. To build a parallel branch with the items 4 and 5, use the "Indent" icon.



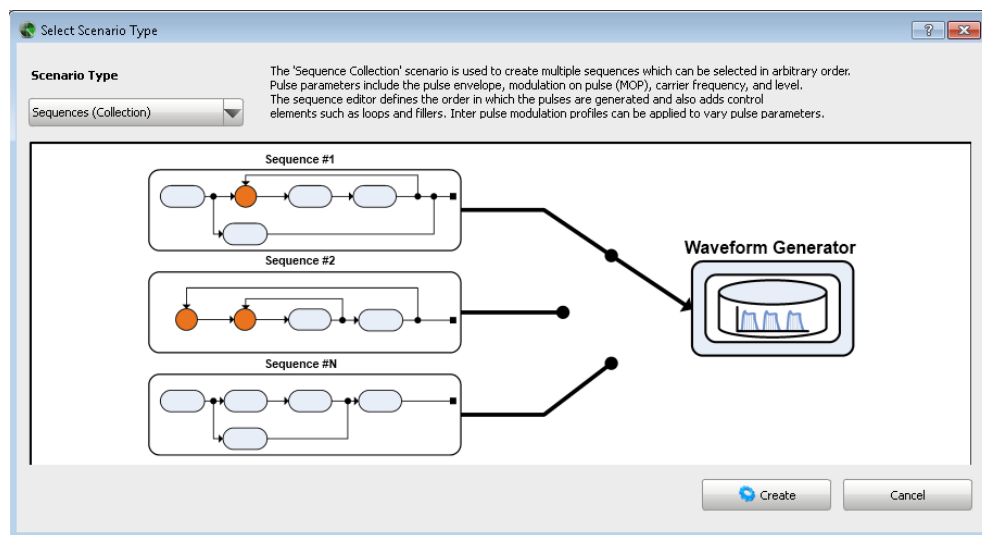
The block diagram confirms that the items are overlaid. The second parallel branch consists of two subsequent processed pulses; the fourth item will be processed after elapsing of the overlay duration.

To create a scenario with several pulse train sequences

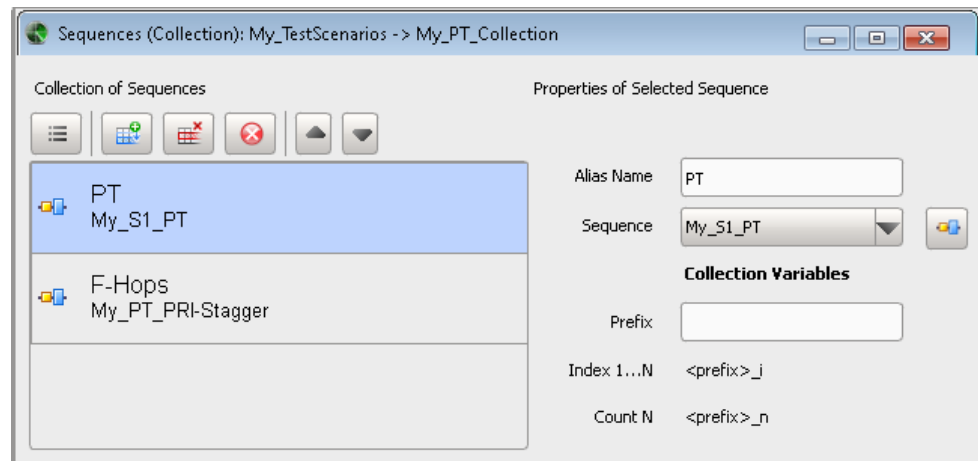
If your test situation requires to test the receiver's ability to detect different pulse sequences, you can use "Sequences (Collection)" scenario. In this dedicated scenario, you can configure several pulse train sequences and switch between them sequentially.

The following example creates a complex scenario with two pulse train sequences to explain the principle.

1. Select "Repository Tree > Scenario > New".
2. In the "Scenario Wizard", select "Scenario Type > Sequences (Collection)".

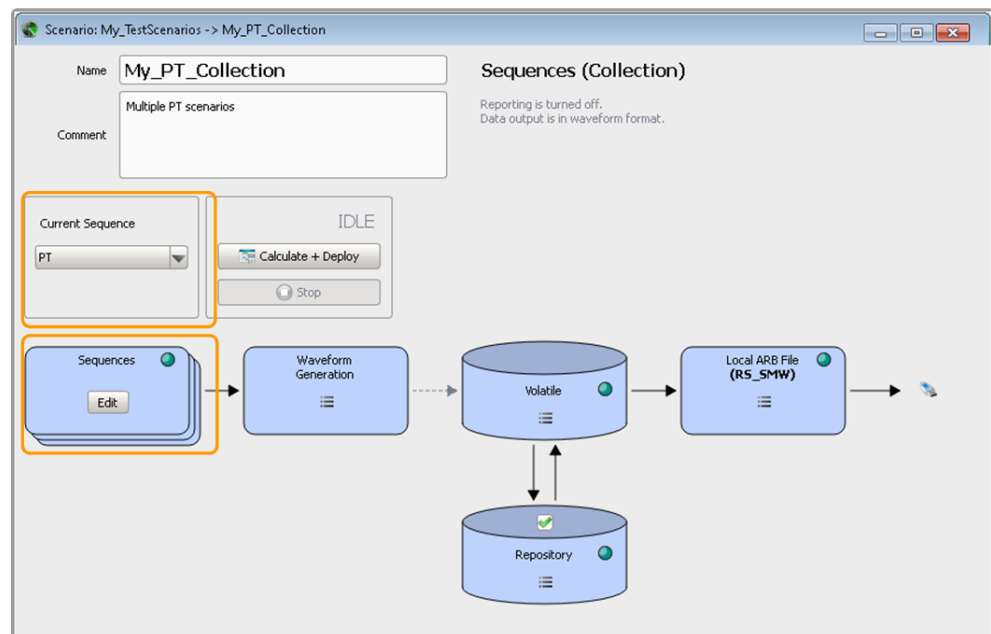


3. Confirm with "Create".
4. Enter a scenario name and add a description.
5. In the block diagram, select "Sequences > Edit".
The "Multiple Sequences" dialog opens.
6. Use the standard "New" function and insert the first sequence in the list.
You can select from the list of available sequences or create a new one.
See also ["To create a simple sequence"](#) on page 124.
7. In the list of sequences, select a "Sequence" and use the standard "New", "Append/Remove" and "Up/Down" functions to add new items to the list and reorder them.



For description of the provided settings, see [Chapter 8.2.6, "Lists with Multiple Sequences"](#), on page 123.

8. In the "Scenario" dialog, select the name of the sequence that is used, e.g. "PT".



For description of the provided scenario settings, see [Chapter 6.2, "Scenario Settings"](#), on page 55.

9 Defining and Enabling Inter-Pulse Modulation Effects

Per default, pulses calculated from the settings of one table row in the sequence description table are identical. To simulate a change of parameters from one pulse to one another, like PRI stagger for example, you can define and enable inter-pulse modulation (IPM) effects and assign them to the sequence elements.

Throughout this description, the IPM effects are commonly named parameter variations.

Pulse repetition patterns

The pulse repetition patterns are commonly used to:

- Avoid range ambiguities caused by echoes of far away objects, which fall into the detection period of the next pulse
- Protect the radar signal against jamming
- Deal with clutter (ground, sea, weather)
- Make the radar more difficult to be identified by frequency hopping

IPM effects

In the following, two common IPM technics, the Jittered PRI and Staggered PRI, are used to explain the principle:

• Jittered PRI

In general, a jitter is understood as the short-term variations of a significant pulse parameter, deviating from their ideal positions in time in either random or ordered way (see [Figure 9-1](#)).

You can, for example, use the random variation of the rising and falling pulse edge positions to simulate a technically imperfect trigger signal.

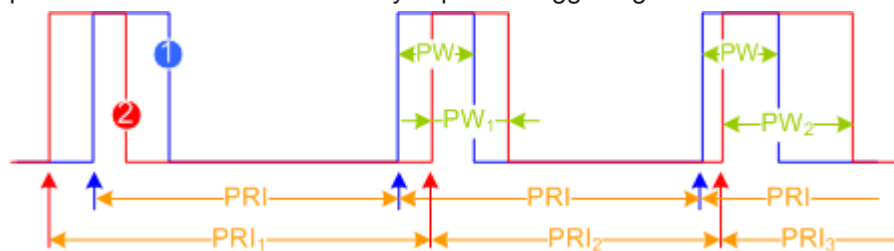


Figure 9-1: Example of Jitter effects

- 1 = Ideal pulse with constant pulse repetition interval (PRI) and pulse width (PW)
- 2 = Pulse affected by Jitters: Jitter PRI ($PRI_1 \neq PRI_2$) and varying pulse width ($PW_1 \neq PW_2$)

As illustrated on [Figure 9-1](#), the time between two subsequent pulses varies randomly, on a pulse-to-pulse basis, as does the pulse width.

• Staggered PRI

Staggered pulse trains are commonly used against range ambiguities and blind speeds, see also "[Pulse repetition patterns](#)" on page 134.

Staggered PRI is a transmission method based on pulse trains with constant pulse width (PW) and varying PRI. The principle of staggered PRI is illustrated on [Figure 9-2](#)

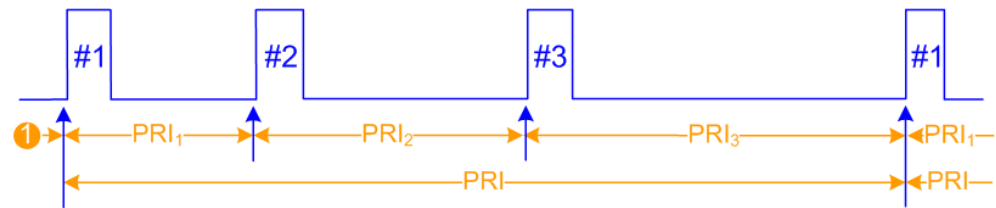


Figure 9-2: Example of an IPM effect: Staggered PRI

- 1 = Stagger start delay
- #1,#2,#3 = Pulse train group composed of three pulses
- PRI = Group PRI: Pulse repetition interval of the pulse train group
- PRI₁,PRI₂,PRI₃ = Pulses transmitted with different PRI

As illustrated on [Figure 9-2](#), within a pulse train group each successive pulse uses a different PRI; the pulse train group is slewing.

- **Frequency Hopping**
Frequency hopping and frequency agility is a further anti-jamming method. If frequency hopping is used, the radar signal switches rapidly and periodically between different carrier frequencies.

There are no predefined parameter variations in the software but IPM profiles can be assigned to the parameter listed in [Table 9-1](#).

Table 9-1: Pulse parameters to that IPM profiles can be assigned

Parameter group	Pulse Parameter
Level	Overshoot [%]
	Offset [dB]
	Δ Level
	Attenuation Top [dB]
	Attenuation > Top Power
	Attenuation Base [dB]
	Attenuation > Base Power
Modulation	Droop [%]
	Ripple [%]
	Ripple > Level
	Ripple Frequency [Hz]
	AM Modulation Depth [%]
	AM Frequency [Hz]
	FM Deviation [Hz]
	FM Frequency [Hz]
	Symbol Rate [Hz]
FSK Deviation [Hz]	

Parameter group	Pulse Parameter
	Chirp Deviation [Hz]
Timing	Rise Time [s] Pulse > Time > Rising Edge
	Pulse Width [s] Pulse > Time > Width
	Fall Time [s] Pulse > Time > Falling Edge
	Delay [s]
	PRI [s]
	PRF [Hz] Pulse repetition frequency, calculated from the selected PRI
Phase	Offset [°]
Δ Freq.	Offset [Hz]

Source profiles

Parameter variations are described as the combination of a source profile and the target element, pulse or sequence, the variation is assigned to. If you assign more than one variation to an element, these variations are superimposed (see [Chapter 9.1.4, "Configure Inter-Pulse Modulation \(IPM\) Settings"](#), on page 148).

Each jitter and IPM effect affects one of the parameters listed in [Table 9-1](#) and can use the following profiles:

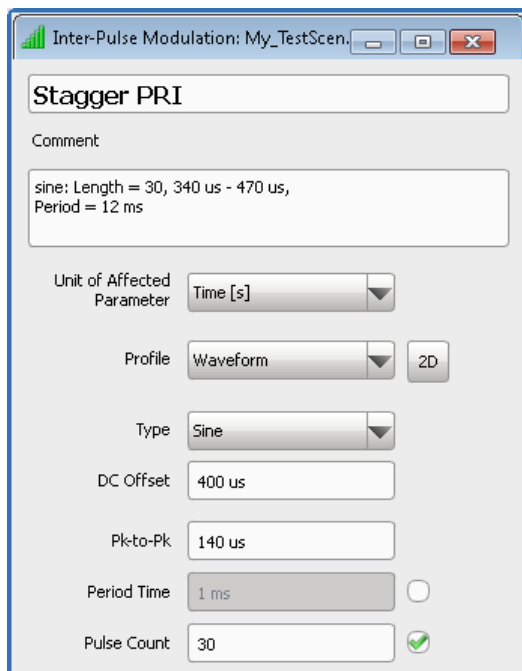
- [Steps](#)
- [Waveform](#)
- [Random List](#)
- [List](#)
- [Interpolated Shape](#)
- [Random](#)
- [Equation](#)

Source profiles are jitter and IPM profiles. You can use the same profiles for more than one variation; the R&S Pulse Sequencer calculates the profile values independent from each other.

9.1 IPM Profiles Settings

Access:

- ▶ Select "Repository Tree > IPM > New".



Available settings:

- [Common IPM Settings](#)..... 137
- [IPM Profiles Settings](#)..... 139
- [Edit List Settings](#)..... 147
- [Configure Inter-Pulse Modulation \(IPM\) Settings](#)..... 148

9.1.1 Common IPM Settings

IPM Profile Name	137
IPM Profile Comment	138
Unit of Affected Parameter	138
Profile	138
2D	138

IPM Profile Name

Enters the name of the profile.

Remote command:

[IPM:CREate](#) on page 329

[IPM:NAME](#) on page 330

[IPM:CATalog?](#) on page 329

[IPM:SElect](#) on page 330

[IPM:REMove](#) on page 331

IPM Profile Comment

Adds a description.

Remote command:

[IPM:COMMeNt](#) on page 331

Unit of Affected Parameter

Sets the units of the [IPM](#) parameters. Available are: "Time [s]", "Frequency [Hz]", "Level [dB]", "Phase [deg]".

Remote command:

[IPM:UNIT](#) on page 376

Profile

Sets the shape of the profile:

- [Steps](#)
- [Waveform](#)
- [Random List](#)
- [List](#)
- [Interpolated Shape](#)
- [Random](#)
- [Equation](#)

Further available parameters depend on the selected shape.

Remote command:

[IPM:TYPE](#) on page 375

2D

Opens a preview diagram of the selected profile.

Two views are available:

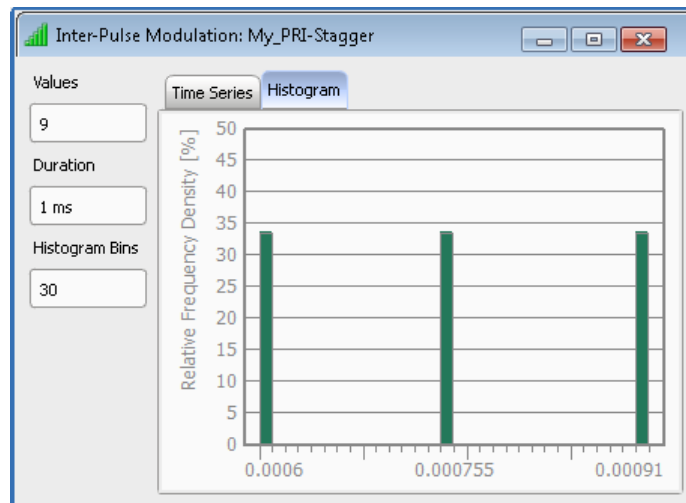
- "Time Series"
That is a visualization of the profile variation over time
- "Histogram"
That is a statistical representation of the relative frequency density, calculated for the selected number of "Histogram Bins".
The histogram plots the number of times data points appear in a particular bin.

The displayed characteristics on both views are defined by:

"Values" Sets the number of values to be displayed.

"Duration" Sets the displayed duration.

"Histogram Bins"



Sets the number of histogram bins, i.e. the histogram sets used to group the obtained data.

Used are bins with equal size. The bin size is calculated from the value range used on the y axis in the time series display and the number of bins.

9.1.2 IPM Profiles Settings

The following IPM shapes can be defined:

Steps.....	139
List.....	140
Waveform.....	141
Interpolated Shape.....	142
Equation.....	143
Random List.....	143
Random Steps.....	144
Random.....	145
Plugin.....	146

Steps

This IPM profile follows a staircase shape and creates a sequence of identical values before it moves on the next one.

The "Burst Length" defines often a certain step is repeated, i.e. how many identical values are created. The profile shape is defined as function of "Number of Steps", "Start" and "Increment" values; the stop value is calculated automatically.

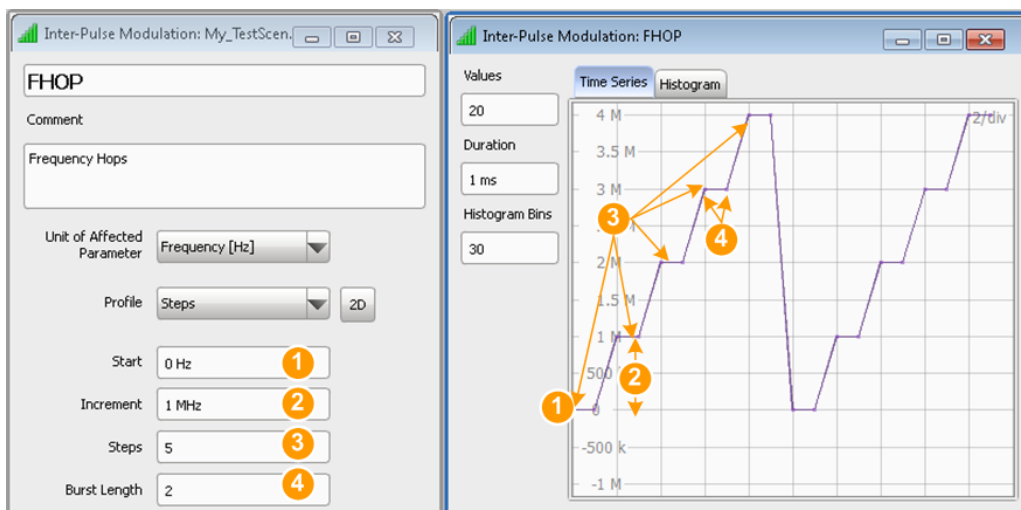


Figure 9-3: IPM with shape "Profile = Steps"

- 1 = "Start" = 0 Hz
- 2 = "Increment" = 1 MHz
- 3 = "Number of Steps" = 5
- 4 = "Burst Length" = 2

Remote command:

- [IPM:STEP:BURSt](#) on page 381
- [IPM:STEP:STEPs](#) on page 381
- [IPM:STEP:START](#) on page 381
- [IPM:STEP:INCRement](#) on page 381

List

The IPM shape is a sequence of discrete values defined in a list. The list items are processed sequentially and cyclically; when the list end is reached, the list is processed from the beginning.

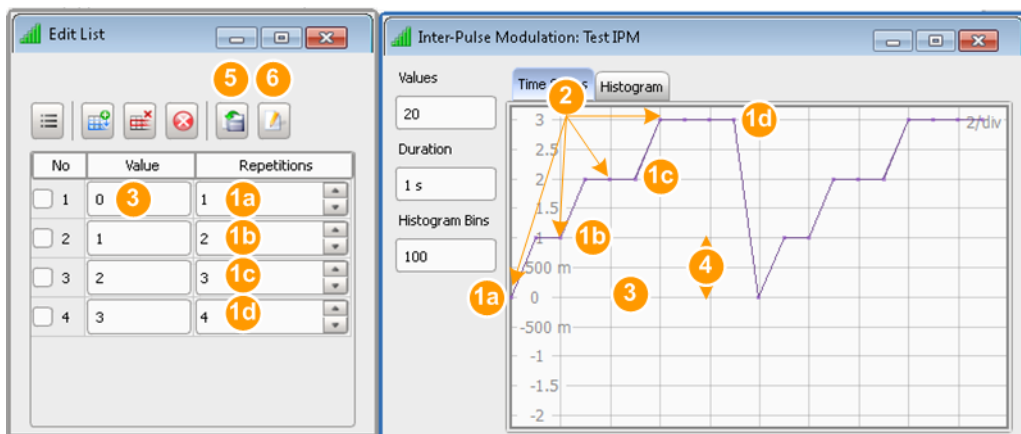


Figure 9-4: IPM with shape "Profile = List"

- 1a, 1b, 1c, 1d, 1e = The increments are repeated different number of times: "Count#1 = 1", "Count#2 = 2", "Count#3 = 1", etc.
- 2 = "List Count" = 4
- 3 = "List Start" = 0

- 4 = "List Increment" = 1
 5 = "Import Data from File" icon
 6 = "Populate list" icon

- "List" Accesses the standard list editor to define a list in table form, see ["Edit List"](#) on page 147.
- To create a list manually, use the standard icons and functions in the context menu (see ["Standard function in the context menus"](#) on page 24). Enter the list values manually.
 - To import a list in ASCII format, select the "Import Data from File" icon.
 - To create a list automatically and fill it in with values, select the "Populate list" icon. Enter the "Start" and "Increment" values and the number of list elements ("Count") and select "Populate".

Remote command:

[IPM:LIST:ITEM:ADD](#) on page 332

[IPM:LIST:ITEM:COUNT?](#) on page 333

[IPM:LIST:ITEM:SELEct](#) on page 333

[IPM:LIST:ITEM:VALue](#) on page 377

[IPM:LIST:ITEM:REPetition](#) on page 377

[IPM:LIST:ITEM:DELete](#) on page 334

Waveform

Sets an IPM shape that follows a ramp, sine, or triangular waveform function. The waveform amplitude is defined by the "Peak-to-Peak" value. A constant "Offset" can be applied to the entire waveform. The waveform period is set as "Period Time" or as number of pulses ("Pulse Count") it lasts.

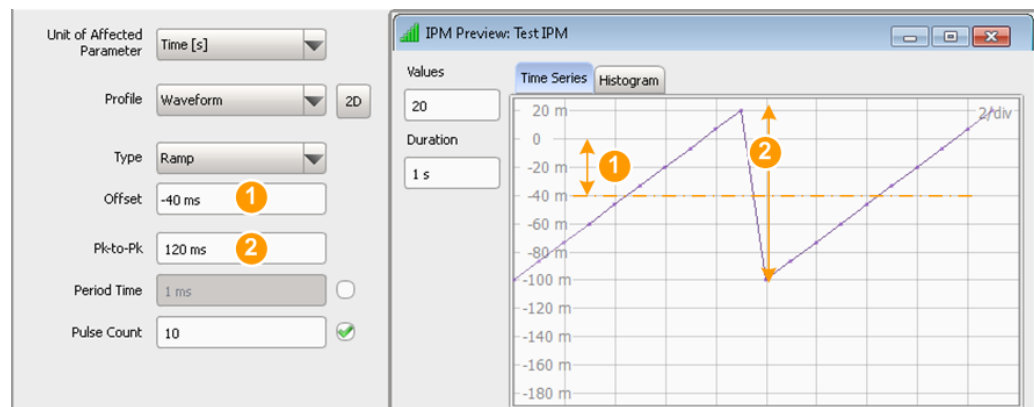


Figure 9-5: IPM with shape "Profile = Waveform" (Type = Ramp)

1 = "Offset" = -40 ms

2 = "Peak-to-Peak" = 120 ms

- "Linear Ramp" The linear ramp is a profile with a sawtooth shape. The generated values monotonically increase from ("Offset"- "Pk-to-Pk"/2) to ("Offset"+"Pk-to-Pk"/2).

- "Sine" The sine profile creates values that follow one period of a sine wave. Enable a "Phase Offset" to change the start phase of the sine wave.
- "Triangular" The triangular profile changes a parameter from a minimum to a maximum value and back to the minimum value, following a triangular shape.

Remote command:

[IPM:WAVeform:BASE](#) on page 383

[IPM:WAVeform:TYPE](#) on page 382

[IPM:WAVeform:PKPK](#) on page 384

[IPM:WAVeform:OFFSet](#) on page 383

[IPM:WAVeform:PHASe](#) on page 383

[IPM:WAVeform:COUNT](#) on page 383

[IPM:WAVeform:PERiod](#) on page 384

Interpolated Shape

The IPM shape is a sequence of values defined in a list. The list items are mapped to several repetitions ("Pulse Count") or equally distributed over a selected period of time ("Period Time").

The transition between the increments can also be linearly interpolated.

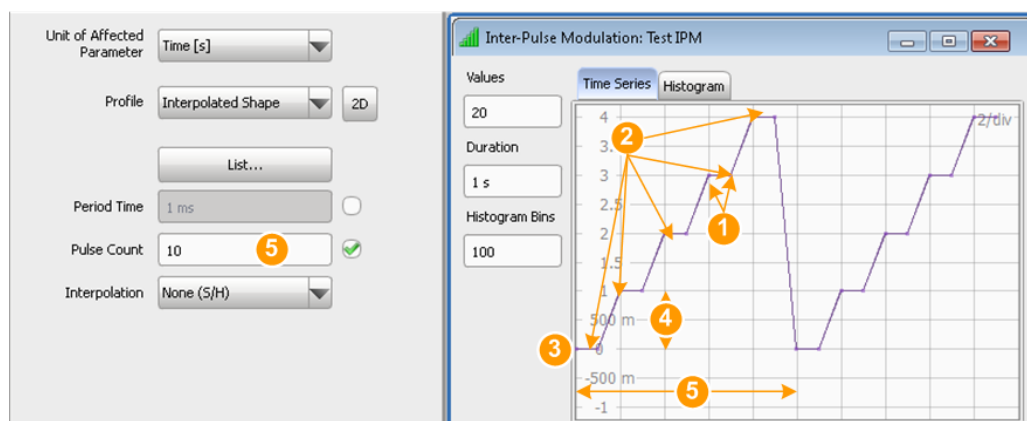


Figure 9-6: IPM with shape "Profile = Interpolated Shape" ("Interpolation = None (S/H)")

1 = Each increment is repeated two times = "Pulse Count"/"List Count"

2 = "List Count" = 5

3 = "List Start" = 0

4 = "List Increment" = 1

5 = "Pulse Count" = 10

"Period Time" Sets the period of time over that the list items are equally distributed.

"Pulse Count" Sets the number of pulses for that the data from the list is used.

Remote command:

[IPM:SHAPe:BASE](#) on page 380

[IPM:SHAPe:PERiod](#) on page 381

[IPM:SHAPe:COUNT](#) on page 380

[IPM:SHAPe:INTerpol](#) on page 380

Equation

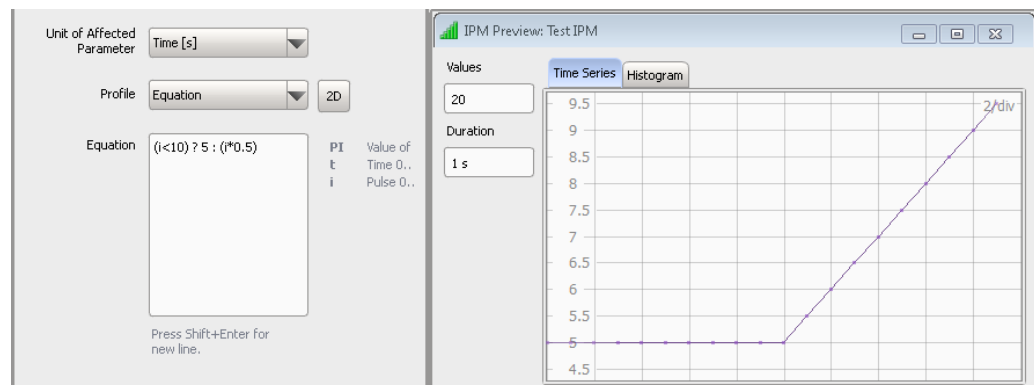
Defines the IPM shape as a function of:

$t = 0 \dots T$ is the absolute time

$i = 0 \dots N-1$ is a pulse counter, where N is the maximum number of pulses

Example:

$(i < 10) ? 5 : (i * 0.5)$



See [Chapter B, "Formula Syntax"](#), on page 492.

Remote command:

[IPM:EQUation](#) on page 377

Random List

The IPM shape is a sequence of discrete values that are randomly selected from a list with user defined values.

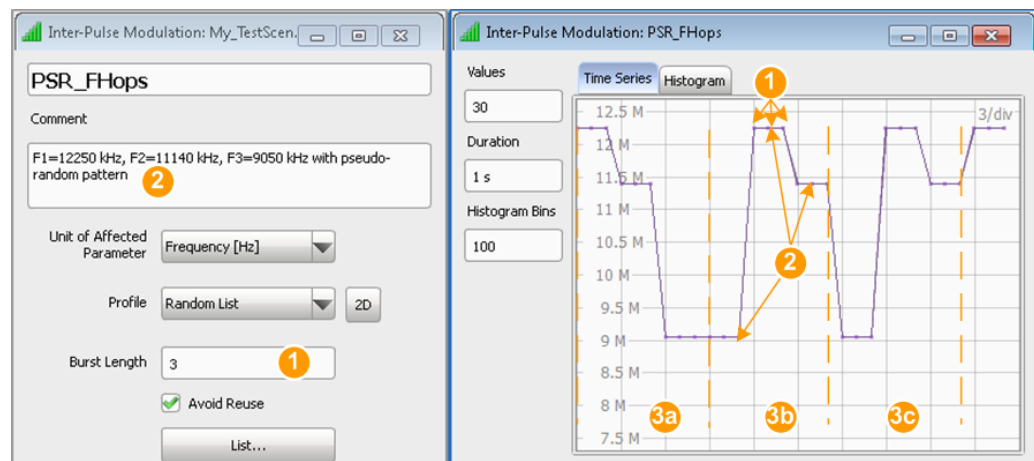


Figure 9-7: IPM with shape "Profile = Random List"

1 = "Burst Length" = 3

2 = "List Items" = 3

3a, 3b, 3c = Subsequent patterns with values that are selected on a pseudo-random manner from the defined 3 list items

"Burst Length" Defines how often an increment is repeated.

- "Avoid Reuse" Ensures that each value is used only once per burst.
- "List" Accesses the standard list editor to define a list in table form, see ["Edit List"](#) on page 147
- To create a list manually, use the standard icons and functions in the context menu (see ["Standard function in the context menus"](#) on page 24). Enter the list values manually.
 - To import a list in ASCII format, select "Import Data from file" icon, navigate to a suitable file, select it and import it.
 - To create a list automatically and fill it in with values, select the "Populate List" icon.
Enter the "Start" and "Increment" values and the number of list elements ("Count") and select "Populate".

Remote command:

[IPM:RLIST:BURSt](#) on page 379

[IPM:RLIST:REUSE](#) on page 380

Random Steps

The IPM shape is a sequence of discrete values and step size that are randomly selected from a user defined value ranges.

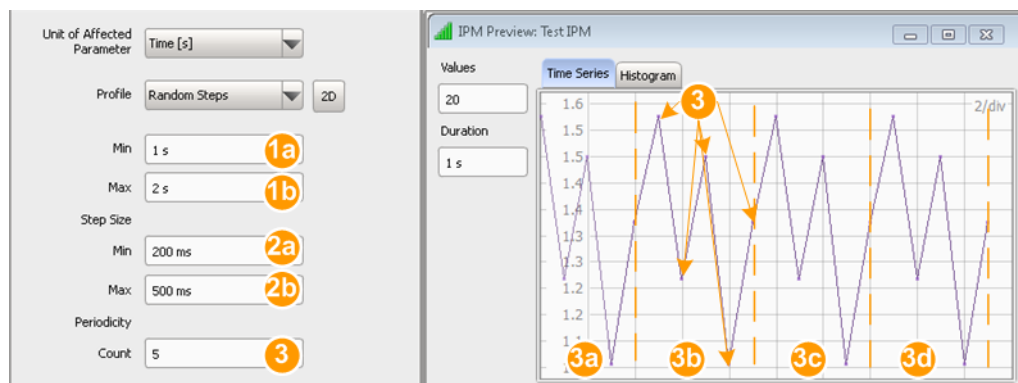


Figure 9-8: IPM with shape "Profile = Random Steps"

1a, 1b = User defined value range

2a, 2b = User defined step size

3 = "Periodicity > Count" = 5, i.e. selected are 5 random values

3a to 3d = Subsequent patterns

"Periodicity > Count"

Sets the pattern length, i.e. the number of values that are selected on a pseudo-random manner

Remote command:

[IPM:RSTep:MINimum](#) on page 382

[IPM:RSTep:MAXimum](#) on page 382

[IPM:RSTep:STEP:MINimum](#) on page 382

[IPM:RSTep:STEP:MAXimum](#) on page 382

[IPM:RSTep:PERiod](#) on page 382

Random

Random jitter is unlimited in its value range and is caused by thermal noise and similar effects. The IPM shape is a sequence of random values according to the select "Distribution" function.

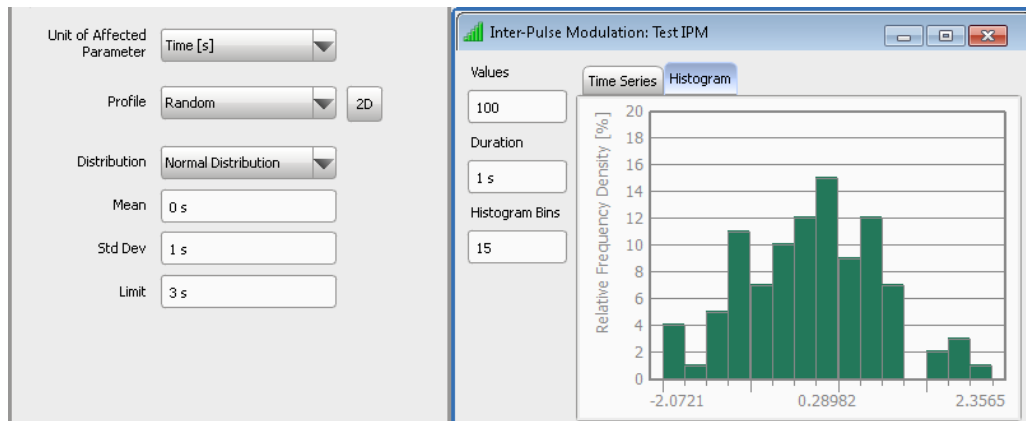


Figure 9-9: IPM with shape "Profile = Random" and "Distribution = Normal Distribution"

"Uniform Distribution"

With this distribution, values occur with the same probability in the range between the minimum and maximum level. The granularity is the "Step" value.

"Normal Distribution"

The Gauss probability density function or normal distribution is characterized by the static parameters "Mean" (μ), "Standard Deviation" (σ) and "Limit".

The Figure 9-10 illustrates the probability at which values would occur related to the standard deviation if no limit was set.

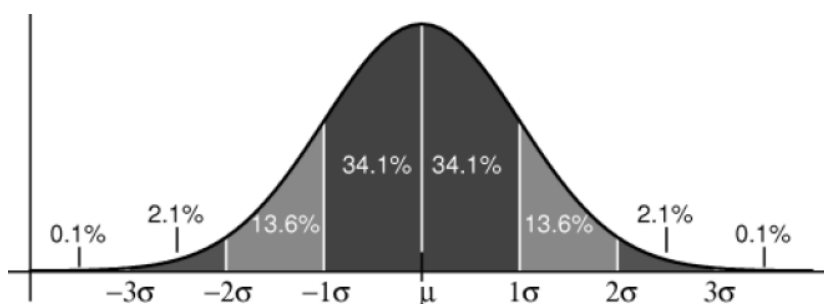


Figure 9-10: Normal distribution

"U Distribution" The U distribution is characterized by a function with lower limit and upper limit that are calculated from selected "Center" and "Range" values.

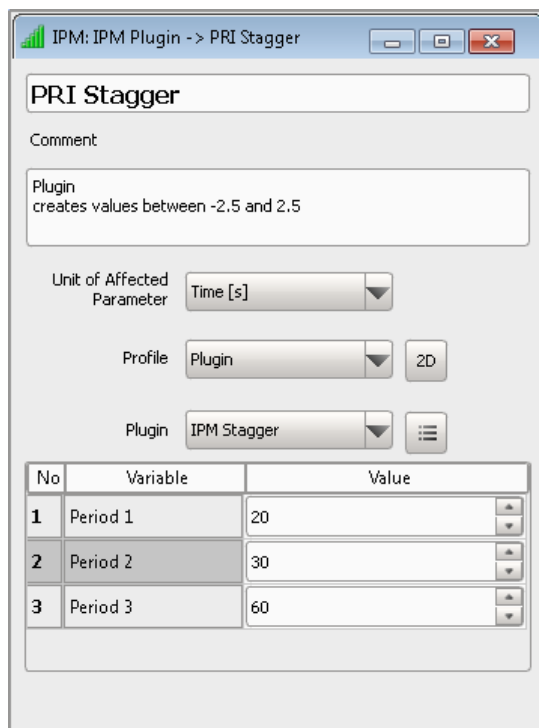
Remote command:

- [IPM:RANDOM:DIStRiBution](#) on page 378
- [IPM:RANDOM:UNIForm:MINimum](#) on page 379
- [IPM:RANDOM:UNIForm:MAXimum](#) on page 379
- [IPM:RANDOM:UNIForm:STEP](#) on page 379

[IPM:RANDom:NORMAl:MEAN](#) on page 378
[IPM:RANDom:NORMAl:STD](#) on page 378
[IPM:RANDom:NORMAl:LIMit](#) on page 378
[IPM:RANDom:U:CENTer](#) on page 379
[IPM:RANDom:U:RANGe](#) on page 379

Plugin

Custom IPM profiles are used in the same way as the internal IPM profiles.



See [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.

"Plugin Name" Selects and loads a user-defined IPM profile from a file. The IPM profile must exist in the "Plugin" library of the repository.

"Plugin Variables" Sets the values of the variables defined and used in the plugin. Use the standard context menu functions to:

- Reset the plugin variables
- Access the load "Plugin" settings, see [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297

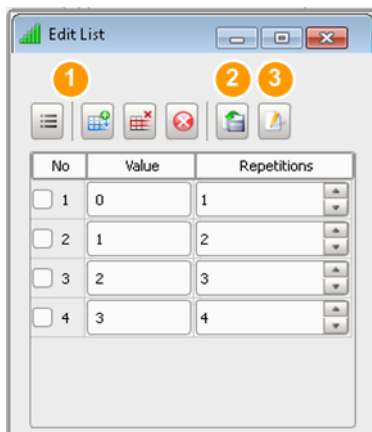
Remote command:

[IPM:PLUGin:NAME](#) on page 330
[IPM:PLUGin:VARiable:CATalog?](#) on page 393
[IPM:PLUGin:VARiable:SElect](#) on page 393
[IPM:PLUGin:VARiable:VALue](#) on page 393

9.1.3 Edit List Settings

Edit List

List editor with standard functions.



1 = Standard functions for item handling

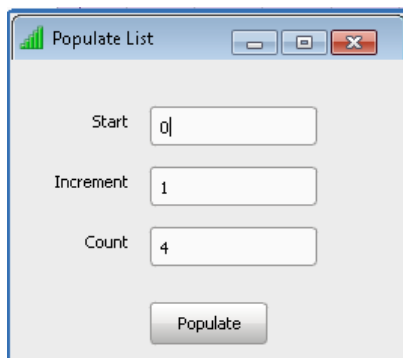
2 = "Import Data from File" icon

3 = "Populate list" icon

"New/Select" Accesses standard functions for item handling like "Select All", "Remove All", "Insert After", etc.
See [Table 3-4](#).

"Import Data from File"
Opens the Windows Explorer to navigate and select an ASCII file with list data.

"Populate List" Accesses the "Populate List" dialog.



To fill in a list with values automatically, set:

- "Start": start value
- "Increment": step with that the values are incremented
- "Count": number of list items.

"Item" Selects an item.

"No" Automatically assigned number to each list item.

"Value" Enters the value of the list item.

"Repetitions" Defines how many times a particular list item is repeated.

Remote command:

`IPM:LIST:LOAD` on page 377

`IPM:LIST:SAVE` on page 377

`IPM:LIST:CLEAr` on page 335

9.1.4 Configure Inter-Pulse Modulation (IPM) Settings

Access:

- ▶ In the "Sequence" dialog, select a pulse item and select "IPM > Static".

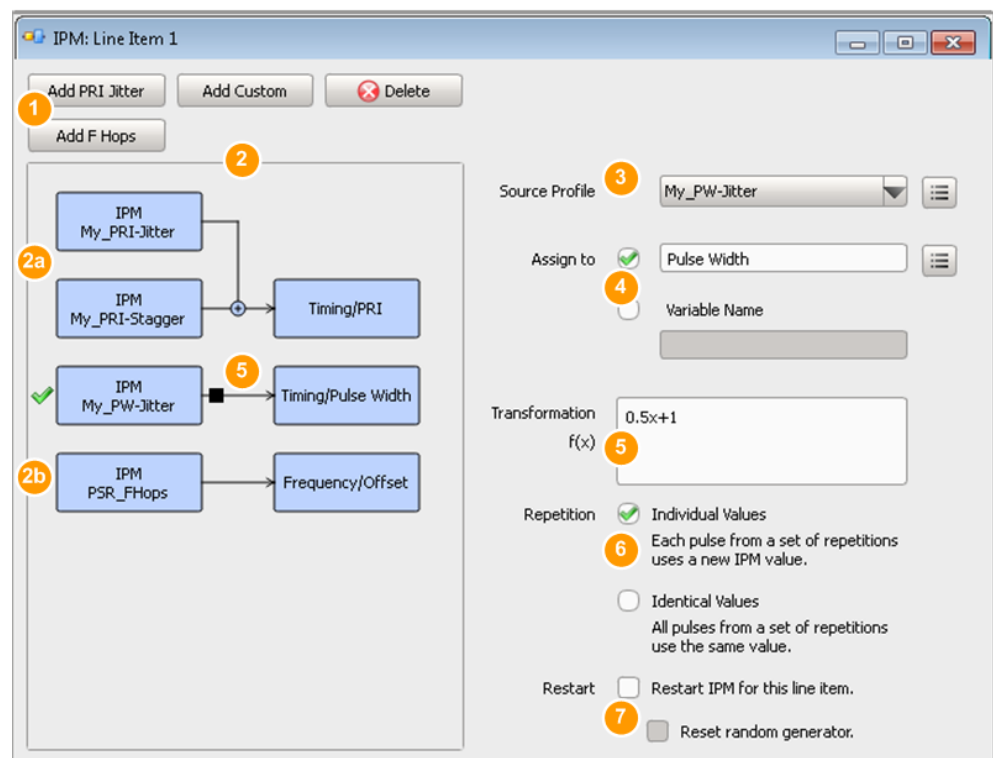


Figure 9-11: Configure Inter-Pulse Modulation: Understanding the displayed information

- 1 = Standard add/delete functions to create IPM of the selected type or remove the selected IPM from the repository
- 2 = Graphical representation of the configured variations and the parameters on that they are applied
- 2a = Two variations are assigned the same element ("Assign to > Timing > PRI"); these variations are added
- 2b = Second variation is assigned to a different element ("Assign to > Frequency > Offset")
- 3 = Variation profile
- 4 = Defines the pulse parameter or the variable on that the IPM is applied
- 5 = Enables and defines a transformation
- 6 = Sets the way repeating pulses are processed
- 7 = Configures the random generator

Available settings:

Add PRI Jitter/Add Frequency Hoping/Add Custom/Delete.....	149
IPM Graph.....	149
Source Profile.....	149
Assign to.....	149
Transformation f(x).....	149
Repetition.....	150
Restart.....	150

Add PRI Jitter/Add Frequency Hoping/Add Custom/Delete

Standard functions to create *IPM* of the selected type, to disable the selected *IPM*, or to remove an automatically created one from the repository.

Remote command:

`SEquence:ITEM:IPM:ADD` on page 332
`SEquence:ITEM:IPM:COUNT?` on page 333
`SEquence:ITEM:IPM:SElect` on page 333
`SEquence:ITEM:IPM:DElete` on page 335

IPM Graph

Graphical representation of the configured variations and the elements they are applied on.

Source Profile

Lists the available Jitter and *IPM* profiles in the repository.

To create a profile or to edit an existing one, select the "New/Select" icon.

Remote command:

`SEquence:ITEM:IPM:SOURce` on page 385

Assign to

Profiles can be assigned to an existing pulse parameter or to a variable.

- To select a parameter, use the "New/Select" icon
- To select a variable, enter the variable name. This variable is available within the current sequence segment (table row).
The variable prefix is a case-sensitive entry.

Parameter variations can be applied on the pulse parameters listed in [Table 9-1](#).

Remote command:

`SEquence:ITEM:IPM:TARGet:TYPE` on page 376
`SEquence:ITEM:IPM:TARGet:PARAmeter` on page 376
`SEquence:ITEM:IPM:TARGet:VARiable` on page 376

Transformation f(x)

Transformations are an alternative way to describe the output value of the *IPM* as a mathematical function of the raw value from the *IPM* profile.

Example:

$$0.5*x + 1$$

The example uses the equation $f(x) = a*x+b$, where the multiplier $a = 0.5$ and the offset $b = 1$.

See also [Chapter B, "Formula Syntax"](#), on page 492.

Remote command:

[SEquence:ITEM:IPM:EQUation](#) on page 384

Repetition

Defines the way the variations are applied on repeating pulses:

"Individual value"

A new value is calculated per pulse.

"Identical value"

The same values are used for all repetition of a particular item in the [Sequence > Sequence Description Table](#).

New values are selected each time the same IPM profile is applied to a different item.

Use the "Restart" options to reset an IPM profile.

Remote command:

[SEquence:ITEM:IPM:MODE](#) on page 384

Restart

If the IPM profile uses a random generator, you can:

- Restart the IPM per sequence line item, i.e.:
 - Resets incremented values
 - Starts processing of list from the first item
 - Restarts plugins
- Reset the start seed of random generator to reproduce the generated random sequence at a latter time.

Remote command:

[SEquence:ITEM:IPM:REStart](#) on page 385

[SEquence:ITEM:IPM:RANDom:RESet](#) on page 384

9.2 How to Create IPM Profiles and Use Them to Vary Pulse Parameters

See:

- ["To create an IPM with the "Startup Assistant" on page 151](#)
- ["To access the IPM profile dialog" on page 151](#)
- ["To create a simple staggered PRI IPM profile" on page 151](#)
- ["To assign the staggered PRI profile to the PRI parameter of a pulse" on page 153](#)
- ["To visualize the impact of the IPM profiles" on page 155](#)
- ["To create and assign a PRI Jitter" on page 158](#)

- ["To vary pulse parameters simultaneously"](#) on page 159
- ["To create and apply an F Hop \(frequency hopping\) pattern"](#) on page 156

To create an IPM with the "Startup Assistant"

1. In the menu bar, select "Help > Wizard".
2. Select "Create a new inter-pulse modulation > On".
3. Select "Next".
4. Select the "Unit" and the "Profile" type.
See [Unit of Affected Parameter](#) and [Profile](#).

To access the IPM profile dialog

Perform one of the following:

1. Select "Repository Tree > IPM > New".
2. Open the "Repository Tree > Sequence > Sequence Description > IPM > IPM/Static > Configure Inter-Pulse Modulation" dialog and:
 - a) On the right of "Source Profile" parameter, open the context menu
 - b) Select "New IPM".

A new empty IPM profile is created and added to the project tree.

New profiles are named IPM <n> where n is a number starting at one.

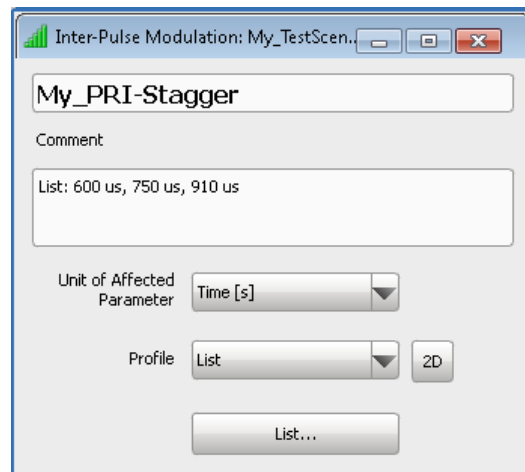
You can add information to describe and identify the profile, like a name or a comment.

To create a simple staggered PRI IPM profile

In this example, we create an IPM profile that contains three time values, defined in a list form. We assign this IPM profile to the PRI parameter of a pulse, see ["To assign the staggered PRI profile to the PRI parameter of a pulse"](#) on page 153.

Generated is a pulse train containing three pulses, as for example the pulse train on [Figure 9-2](#).

1. Open an IPM profile dialog.
See ["To access the IPM profile dialog"](#) on page 151.
2. Select "Unit >Time [s]" and "Profile > List"



See also [Chapter 9.1.2, "IPM Profiles Settings"](#), on page 139.

3. Select "List..." and create a list with three items. The value of each item is set to the required PRI; items are processed once.

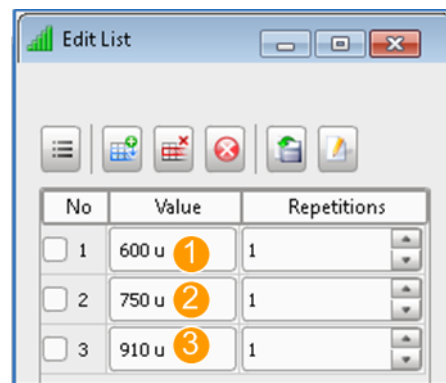
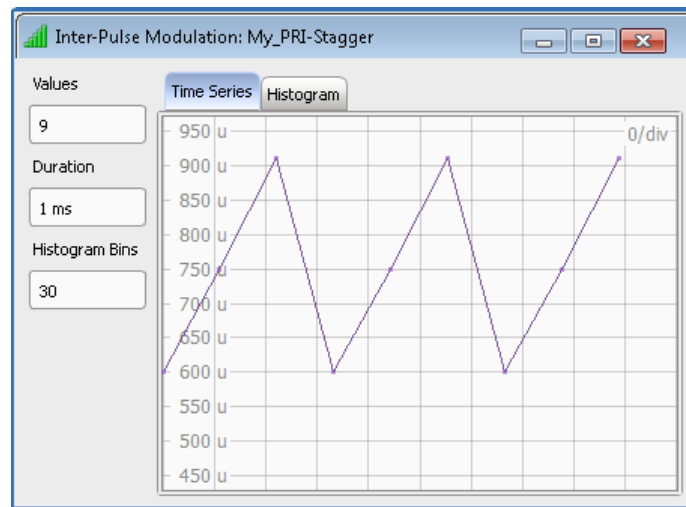


Figure 9-12: Example of a simple staggered PRI IPM profile

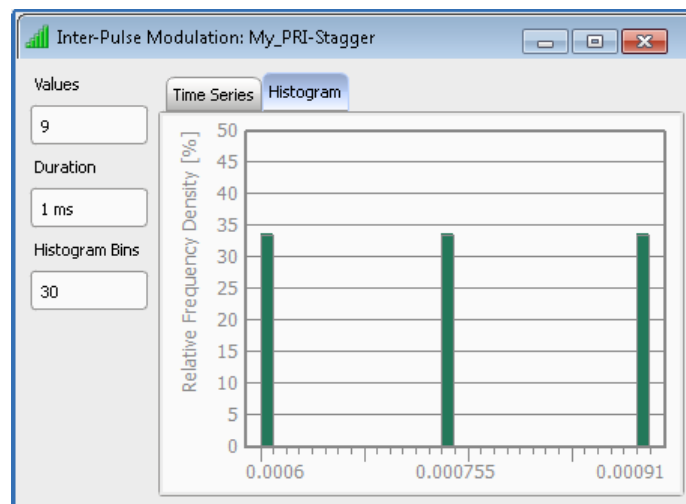
1 = PRI₁ = 600 us
 2 = PRI₂ = 750 us
 3 = PRI₃ = 910 us

4. In the "IPM" dialog, select "2D > Time Series" to visualize the configured IPM profile in the time domain.



Created is a simple profile with three values only.

5. Select "Histogram" to retrieve statistical information on the IPM profile.

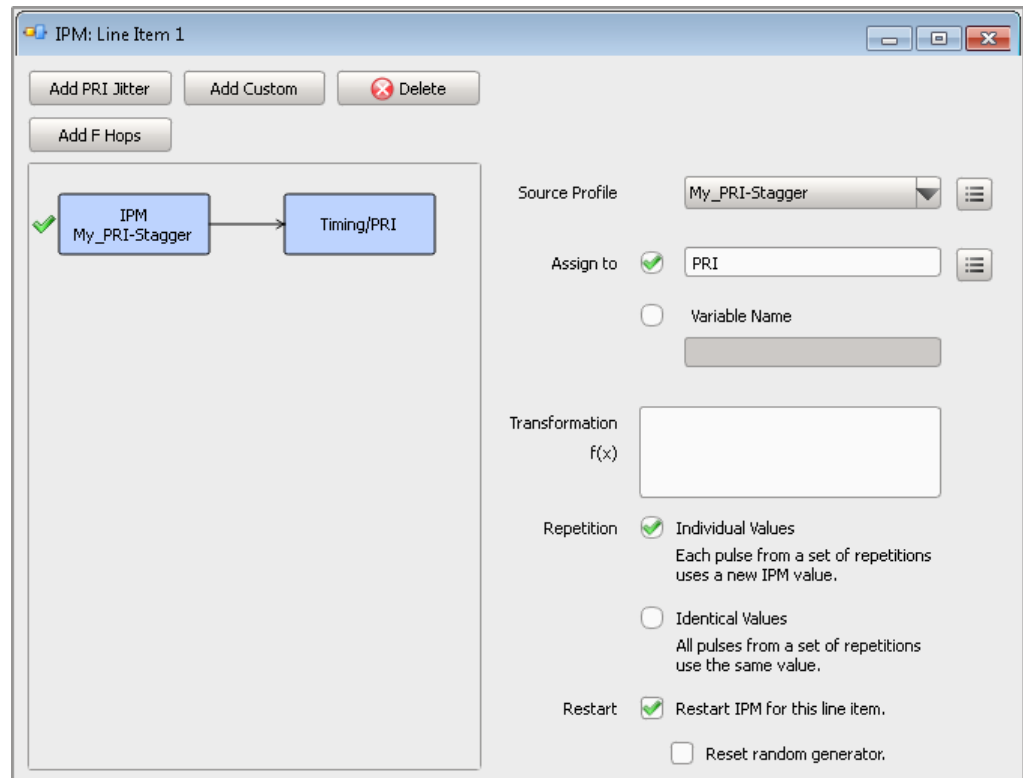


In practice, a staggered PRI profile follows more complex structure and comprises of a larger number of pulses, see for example [Figure 9-4](#).

To assign the staggered PRI profile to the PRI parameter of a pulse

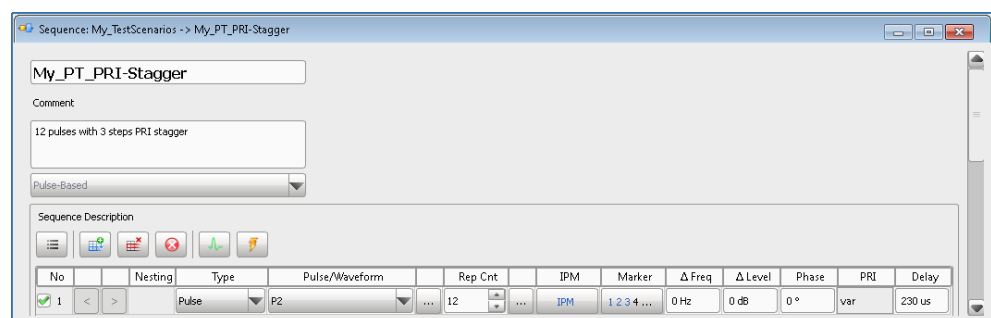
1. Open the pulse sequence on that the created profile should be applied, e.g. "My_PT_PRI-Stagger".
See also ["To create a simple sequence"](#) on page 124.
2. In the "Sequence" dialog, select a pulse item and select "IPM > Static".
The "Configure Inter-Pulse Modulation" dialog opens.
Per default, there are no predefined parameter variations.
3. To assign the created IPM profile:
 - a) Select "Add Custom".
New empty profile "IPM NONE" is created.

- b) Select "Source Profile > My_PRI_Stagger".
- c) On the right of "Assign to" parameter, open the context menu and select "Timing > PRI".



For description of the provided settings, see [Chapter 9.1.4, "Configure Inter-Pulse Modulation \(IPM\) Settings"](#), on page 148.

4. In the "Sequence" dialog:
 - a) set "Repetition Count = 12"
 - b) set "Delay = 230 us" to enable a stagger start delay, see [Figure 9-2](#)



The "Sequence" dialog confirms, that a parameter variation varying the PRI is applied ("IPM = IPM" and "PRI = var").

To visualize the impact of the IPM profiles

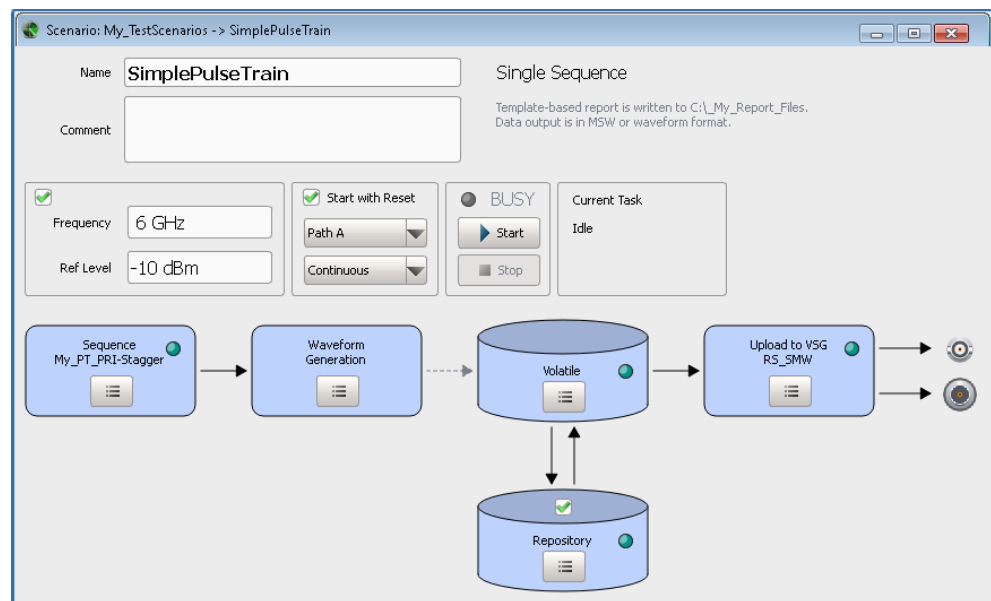
We assume, that a PRI Stagger has been configured and applied to a pulse train, as described in ["To assign the staggered PRI profile to the PRI parameter of a pulse"](#) on page 153.

We also assume, that a "Pulse Train" or a "Pulse Train (Collection)" scenario is configured, a signal generator is selected, and the sequence is assigned to this scenario (see [Chapter 6.3, "How to Select and Create a Test Scenario"](#), on page 68).

To visualize the resulting sequence:

1. In the "Scenario" dialog, select "Start".

The software calculates the waveform. Green LEDs indicate that the processing is completed.



2. Select "Volatile > View".

The "Waveform Viewer" confirms the configured pulse sequence with enabled PRI Stagger.

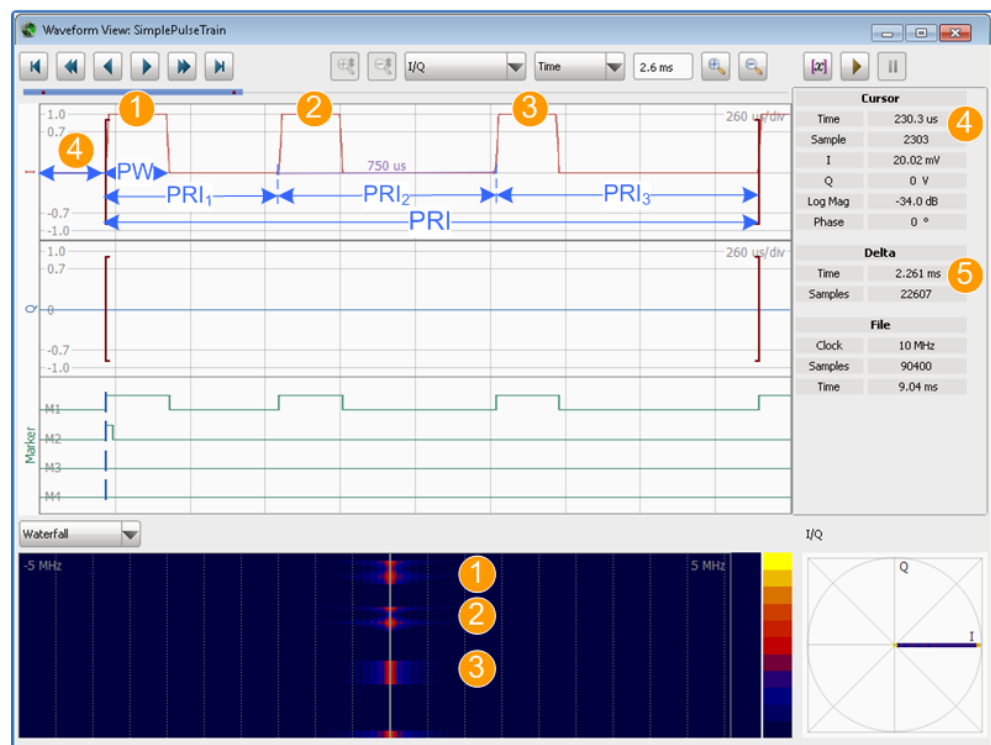


Figure 9-13: Waveform Viewer: Example of a pulse train (PW = 200 us, PRI stagger)

- 1, 2, 3 = Three unmodulated pulses, also observed on the Waterfall diagram
- 4 = "Delay = 230 ms"
- PRI₁ = 600 us
- PRI₂ = 750 us
- PRI₃ = 910 us
- 5 = PRI = 2260 us

For detailed description of the provided settings, see [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264.

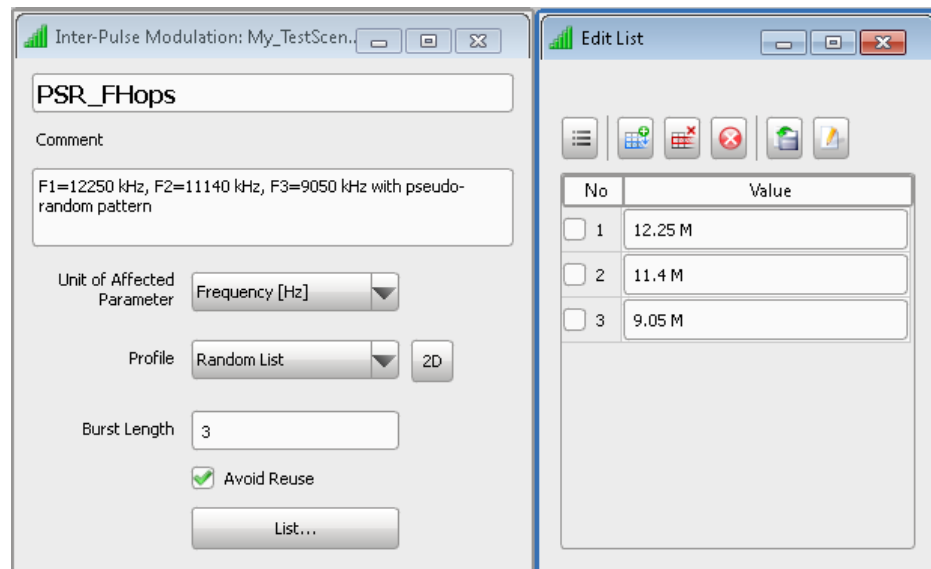
To create and apply an F Hop (frequency hopping) pattern

1. In the "Sequence" dialog, select "IPM" to open the "Configure Inter-Pulse Modulation" dialog.

2. Select "Add F Hops".

A new IPM profile is created, automatically named and assigned to the "Frequency/Offset" parameter.

3. Configure the parameters for example as shown on the following figure.



- Use the 2D view to display the configured pattern, see also [Figure 9-7](#).
- Observe the "Sequence" dialog.

The sequence configuration settings confirm the configuration.

No	Nesting	Type	Pulse/Waveform	Rep Cnt	IPM	Marker	Δ Freq	Δ Level	Phase	PRI	Delay
<input type="checkbox"/> 1	< >	Pulse	P1	30	IPM	1 2 3 4 ...	var	0 dB	0 °	1 ms	0 s

- Use the "Waveform Viewer" to visualize the generated waveform.

See also:

- ["To visualize the impact of the IPM profiles"](#) on page 155

- [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264

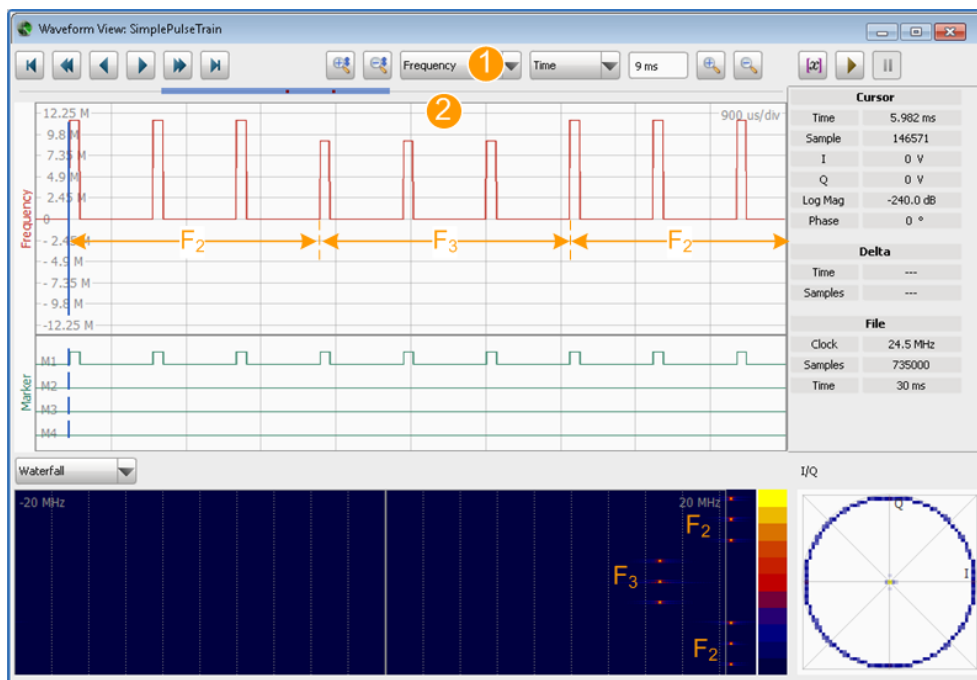


Figure 9-14: Waveform Viewer: Example of a pulse train (PW = 100 us, Frequency hopping)

- 1 = "View Mode > Frequency"
- 2 = 30 standard pulses; 3 consecutive pulses per hopping frequency
- F1 = 12250 kHz (not on the display)
- F2 = 11140 kHz
- F3 = 9050 kHz

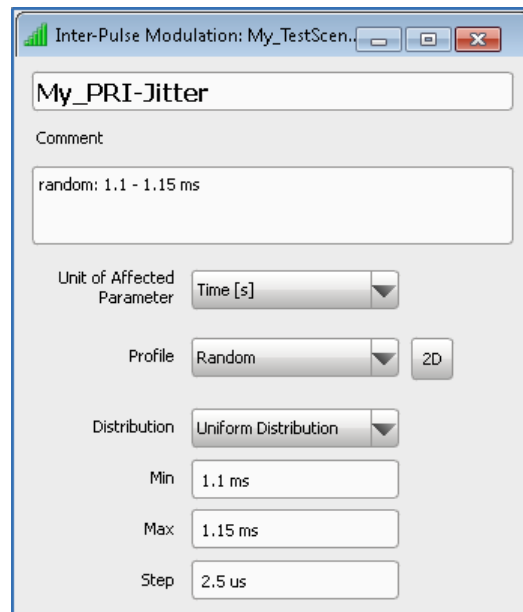
The signal will hop the three frequencies according to a pseudo-random pattern. Each hop frequency is used three times.

To create and assign a PRI Jitter

There is a dedicated function for creating PRI Jitters, the "PRI Jitter" in the "Configure IPM" dialog (see also ["To create and apply an F Hop \(frequency hopping\) pattern"](#) on page 156).

In this example, however, we create manually a randomly varying IPM profile with 20 different values. Within 20 ms, the PRI should vary in the range 1.1 ms to 1.15 ms.

1. Create an IPM Profile that uses "Profile > Random" and for example a "Distribution > Uniform Distribution".
See also ["To access the IPM profile dialog"](#) on page 151.



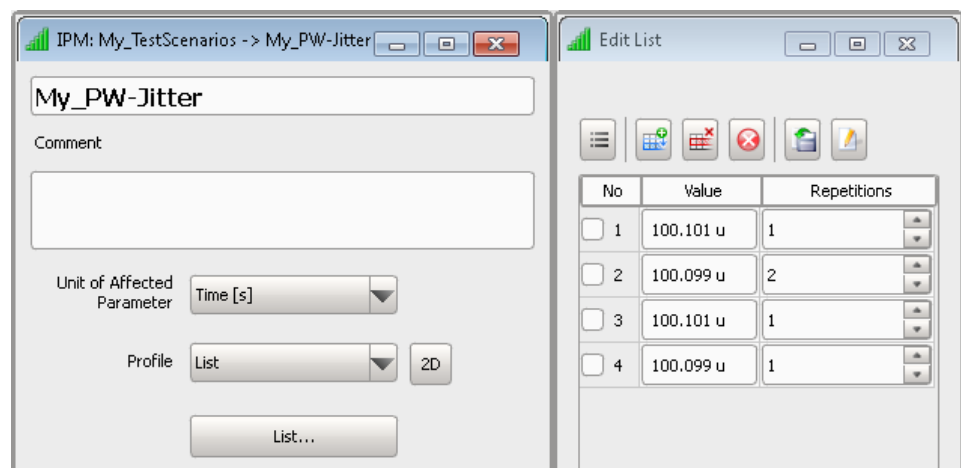
2. Select "2D" to visualize the function.
3. Open the required sequence and assign the IPM profile to a pulse train.
See ["To create and apply an F Hop \(frequency hopping\) pattern"](#) on page 156.

To vary pulse parameters simultaneously

We assume, that a PRI Jitter has been configured and applied to a pulse train, as described in ["To create and assign a PRI Jitter"](#) on page 158.

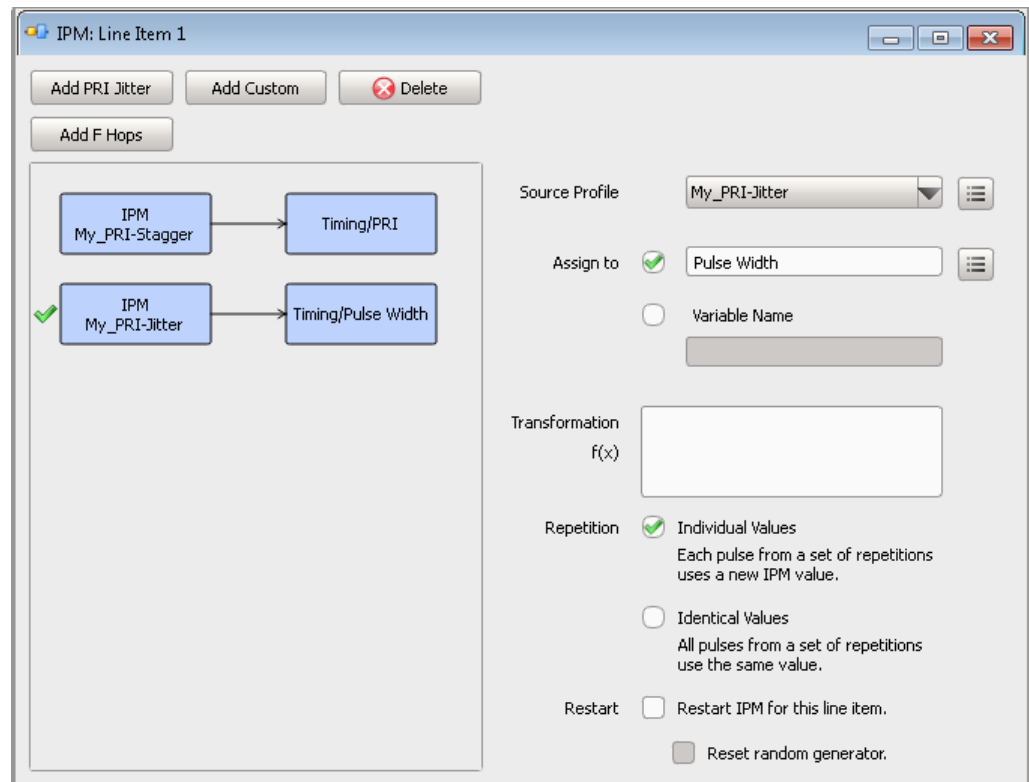
To apply a second variation, for example a PW Jitter, perform the following:

1. Create an IPM Profile, for example as shown on the following figure.



Configured is a pulse width (PW) pattern with 5 values.

2. Open the required sequence and assign the IPM profile to the same pulse train as the PRI Jitter.



Tip: You can vary one pulse parameter with more than one profile, see [Figure 9-11](#).

Configured is a pulse train, consisting of 20 pulses, each with different PRI value. The PW pattern is applied on 5 consequent pulses and is repeated 4 times. In this example, the PRI varies in the range 1.1 ms to 1.15 ms; the PW between 100.101 us and 100.099 us.

10 Defining Antenna Patterns and Antenna Scans

This section informs you about the provided antenna pattern types and antenna scan types. It also helps you select and configure a subset of antenna characteristics as basis for the emitter configuration and its different operating modes.

10.1 Overview of the Main Antenna Parameters

Antennas are elements that radiate energy. Radiated energy focused in one defined direction is referred to as a main beam or main lobe (see [Figure 10-1](#)). Radiated energy in other than the required direction form the side lobes and the back lobe.

The angle at that the radiated energy falls down to the half of the maximum is referred to as half-power beam width (HPBW).

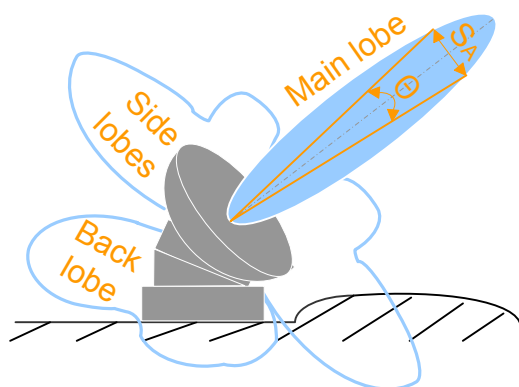


Figure 10-1: Main characteristics of an antenna

Main lobe/beam	= The region around the direction with maximum radiation
Side lobes	= Energy radiated in undesired direction
S_A	= Angular resolution
Θ (Theta)	= Antenna beam width or the -3 dB angle, defined by the half-power (HPBW)

Antenna pattern

Antennas of different type and with different purpose use more wider or more directed beams, for example pencil beams have a typical HPBW of about 2 deg.

The R&S Pulse Sequencer provides a wide range of commonly used antenna patterns, such as Parabolic, Cosecant Squared, or Planar Phased Array antennas. Moreover you can:

- Adjust the antenna pattern settings to simulate for example a directive antenna that concentrates the radiated energy in a particular direction

- Simulate a back lobe pattern
- Create a custom antenna pattern based on the required antenna characteristics (see ["To create a custom antenna pattern based on the required antenna characteristics"](#) on page 190)
- Import your custom antenna pattern (see ["To import a user-defined antenna pattern from file in one of the predefined file formats"](#) on page 191 and ["To import an antenna pattern file in custom file format"](#) on page 193)
- Design your phased array antenna
- Visualize the antenna pattern on a 2D and a 3D preview, see [Figure 10-8](#).

Antenna scan

Antenna scans describe the movement of the antenna beam in three-dimensional space. A conical scan for example is a movement where the beam points in the direction of the target and is rotated in a circle, i.e. the half-power points of the beam draw a circle.

The R&S Pulse Sequencer provides a wide range of commonly used antenna scans, such as a raster scan, helical scan, or conical scan. You can also:

- Adjust the antenna scan settings
- Visualize the configured antenna scan on a 3D preview
- Observe the signal received by a receiver for a particular antenna pattern and antenna scan configuration

See:

- ["Received signal amplitude"](#) on page 211
- ["To visualize the signal received by a static receiver"](#) on page 207

Antenna polarization and polarization loss

The polarization of an antenna describes the orientation of the electric wave radiated by the antenna. Polarization can be constant, or it can rotate with each wave cycle.

The [Table 10-1](#) gives an overview and a short description of the supported polarization types.

Table 10-1: Polarization types

Polarization	Type	Short description
Vertical (V)	Linear polarization	The electrical field is vertical to the propagation direction (Y-axis).
Horizontal (H)		The electrical field is horizontal to the propagation direction (Y-axis).
Circuit Right (CR)	Circular polarization	The electrical field rotates clockwise (CW) relative to the propagation direction.
Circuit Left (CL)		The electrical field rotates counter clockwise (CCW) relative to the propagation direction.

Polarization	Type	Short description
Slant Right (SR)	Linear polarization	The electrical field is rotated at 45° from the vertical case.
Slant Left (SL)		The electrical field is rotated at -45° from the vertical case

The theoretical maximum received power is achieved, if the transmit and the receive antenna are co-polarized and aligned; that is, if the antenna at the emitter and the receiver use the same polarization and point to each other. If the antennas use different polarizations, the received power decreases. The polarization loss is given by the ratio of the received power to the maximum power. Theoretically, if the transmit and the receive antennas are cross-polarized, the polarization loss is infinite. In the practice, however, this value is not reached. In this implementation, a polarization loss of -30 dB is used, if the antennas are cross-polarized.

[Table 10-2](#) lists the polarization loss for any combination of transmit and receive antenna polarizations.

Table 10-2: Polarization loss values depending on the different antenna combinations

Polarization of the transmit antenna	Polarization of the receive antenna	Polarization loss, dB
Vertical (V)	Vertical (V) (co-polarized antennas)	0
	Horizontal (H) (cross-polarized antennas)	-∞ (Implemented as -30 dB)
	Slant Right (SR) or Slant Left (SL)	-3
	Circuit Right (CR) or Circuit Left (CL)	-3
Horizontal (H)	Horizontal (H) (co-polarized antennas)	0
	Slant Right (SR) or Slant Left (SL)	-3
	Circuit Right (CR) or Circuit Left (CL)	-3
Circuit Right (CR)	Circuit Right (CR) (co-polarized antennas)	0
	Circuit Left (CL) (cross-polarized antennas)	-∞ (Implemented as -30 dB)
Circuit Right (CR) or Circuit Left (CL)	Slant Right (SR) or Slant Left (SL)	-3

See:

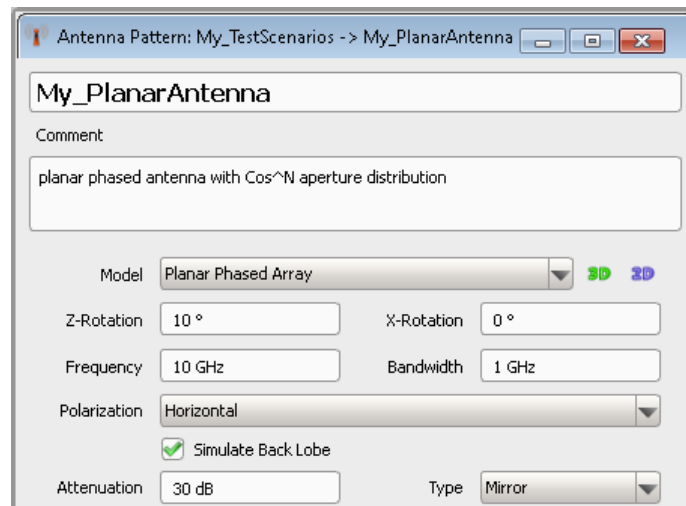
- ["Polarization"](#) on page 166
- ["Polarization"](#) on page 168

- "To observe the effect of polarization" on page 193

10.2 Antenna Pattern Settings

Access:

- ▶ Select "Repository Tree > Antenna Pattern > New".



Settings

Antenna patterns are characterized with the following common and model specific settings:

Antenna Pattern Name.....	165
Comment.....	165
Antenna Model.....	165
Z-Rotation, X-Rotation.....	165
Frequency, Bandwidth.....	166
Polarization.....	166
Simulate Back Lobe, Attenuation, Type	167
2D and 3D diagrams.....	168
L 2D.....	168
L 3D.....	168
L Minimum Level.....	168
L Polarization.....	168
L Pattern.....	168
L Electronic Steering > Z-/X-Rotation.....	169
Dipole Antenna Settings.....	169
Parabolic Antenna Settings.....	169
Gaussian Antenna Settings.....	169
Sin(x)/x Antenna Settings.....	170
Pyramidal Horn Antenna Settings.....	170
Cosecant Squared Antenna Settings.....	171

Planar Phased Array Antenna Settings.....	171
Custom Antenna Settings.....	172
Custom Phased Array Settings.....	173
Cardioid Antenna Settings.....	176
Imported Antenna Settings.....	177
L Import Filter.....	177
L Load/Delete.....	177
L Number of Patterns and Points.....	177
L Import Wizard.....	178

Antenna Pattern Name

Enter the name of the antenna pattern.

Remote command:

[ANTenna:NAME](#) on page 330

[ANTenna:CATalog?](#) on page 329

[ANTenna:CREate](#) on page 329

[ANTenna:SElect](#) on page 330

[ANTenna:REMove](#) on page 331

Comment

Enter a short description.

Remote command:

[ANTenna:COMMeNt](#) on page 331

Antenna Model

Selects the antenna pattern. Use the "2D" and "3D" diagrams to visualize the polar antenna gain patterns.

Further provided antenna pattern settings depend on the selected antenna model, see:

- [Dipole Antenna Settings](#)
- [Parabolic Antenna Settings](#)
- [Gaussian Antenna Settings](#)
- [Sin\(x\)/x Antenna Settings](#)
- [Pyramidal Horn Antenna Settings](#)
- [Cosecant Squared Antenna Settings](#)
- [Planar Phased Array Antenna Settings](#)
- [Imported Antenna Settings](#)
- [Custom Antenna Settings](#)
- [Custom Phased Array Settings](#)
- [Cardioid Antenna Settings](#)

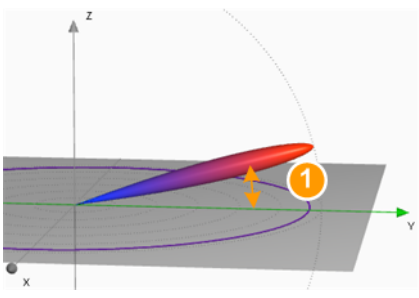
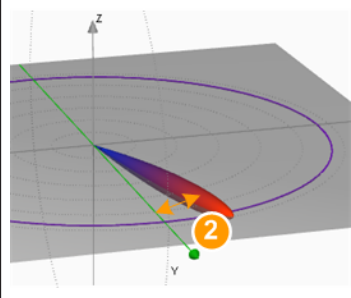
Remote command:

[ANTenna:MODeL:TYPE](#) on page 339

Z-Rotation, X-Rotation

Rotates the antenna pattern around the selected axis.

Table 10-3: Effect of antenna pattern rotation

"X-Rotation = 15°"	"Z-Rotation = 15°"
	
<p>1) Rotation around the x-axis Positive values rotate the antenna boresight towards the positive z-axis.</p>	<p>2) Rotation around the z-axis Positive values rotate the antenna boresight counter clockwise towards the x-axis.</p>

This parameter is useful for example:

- To simulate a radar boresight displacement
- To align a user defined antenna pattern to the boresight;
(see ["To import a user-defined antenna pattern from file in one of the predefined file formats"](#) on page 191).

Remote command:

`ANTenna:MODEl:ROTation:X` on page 346

`ANTenna:MODEl:ROTation:Z` on page 346

Frequency, Bandwidth

Sets the frequency and bandwidth the antenna pattern is calculated for.

See also:

- [Frequency](#)
- [Emitter Modes](#)

Remote command:

`ANTenna:MODEl:FREQuency` on page 344

`ANTenna:MODEl:BANDwidth` on page 344

Polarization

Sets the antenna polarization.

The polarization of an antenna describes the orientation of the electric wave radiated by the antenna. Polarization can be constant, or it can rotate with each wave cycle.

The selected polarization is indicated on the 3D diagram.

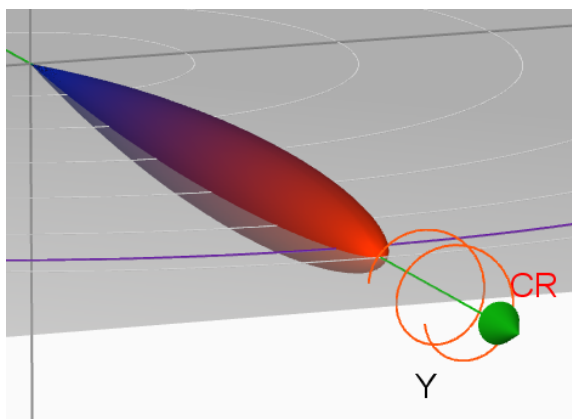


Figure 10-2: Indication of the selected polarization on the 3D diagram

See also "[Antenna polarization and polarization loss](#)" on page 162.

- "Vertical (V)" Linear polarization.
The electrical field is vertical to the propagation direction (Y-axis).
- "Horizontal (H)" Linear polarization.
The electrical field is horizontal to the propagation direction (Y-axis).
- "Circuit Right (CR)" Circular polarization.
The electrical field rotates clockwise (CW) relative to the propagation direction.
- "Circuit Left (CL)" Circular polarization.
The electrical field rotates counter clockwise (CCW) relative to the propagation direction.
- "Slant Right (SR)" Linear polarization.
The electrical field is rotated at 45° from the vertical case.
- "Slant Left (SL)" Linear polarization.
The electrical field is rotated at -45° from the vertical case.

Remote command:

`ANTenna:MODEl:POLarization` on page 339

Simulate Back Lobe, Attenuation, Type

Enables the simulation of a back lobe and sets its power level and shape.

The parameter "Type" defines the shape of the back lobe pattern:

- "Mirror": the back lobe pattern is the mirrored antenna pattern, attenuated with the selected value, see [Figure 10-9](#).
- "Omnidirectional": the back lobe pattern is a hemisphere, attenuated with the selected value.

Remote command:

[ANTenna:MODEl:BACKlobe:ENABle](#) on page 341

[ANTenna:MODEl:BACKlobe:TYPE](#) on page 342

[ANTenna:MODEl:BACKlobe:ATTenuation](#) on page 342

2D and 3D diagrams

You can visualize the antenna characteristics as 2D and 3D interactive diagrams.

2D ← 2D and 3D diagrams

This diagram is a plot of the radiated energy, measured at different angles and at a constant distance from the antenna. The 2D diagram indicates the calculated [HPBW](#) and the 3 dB line.

The 2D diagram display the antenna pattern as a XY (azimuth pattern) or YZ (elevation pattern) graph with polar or with Cartesian coordinates.

- The polar diagram shows the radiation in all possible directions, i.e. the main, side and back lobes, see [Figure 10-8](#).
- The Cartesian diagram is useful to determine the main lobe to back lobe and main lobe to side lobes ratio of highly directive antennas.

Remote command:

[PLOT:POLar:TYPE](#) on page 347

[PLOT:POLar:CUT](#) on page 347

[PLOT:POLar:LOG:MIN](#) on page 347

3D ← 2D and 3D diagrams

This diagram displays the radiated energy as a 3D spatial radiation model.

Minimum Level ← 2D and 3D diagrams

Defines the minimum displayed side lobes level.

Polarization ← 2D and 3D diagrams

Shows the co-polarized and the cross-polarized antenna pattern.

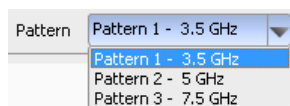
The co-polarized and cross-polarized antennas are selected automatically for the current antenna, see [Table 10-2](#). The cross-polarization is implemented as polarization lose of - 30 dB.

See "[To observe the effect of polarization](#)" on page 193.

This function cannot be used for user-defined antenna patterns imported as a file.

Pattern ← 2D and 3D diagrams

A file describing custom antenna pattern may contain more than one antenna patterns. If a custom antenna pattern is loaded, this parameter selects the antenna pattern number that is currently displayed.



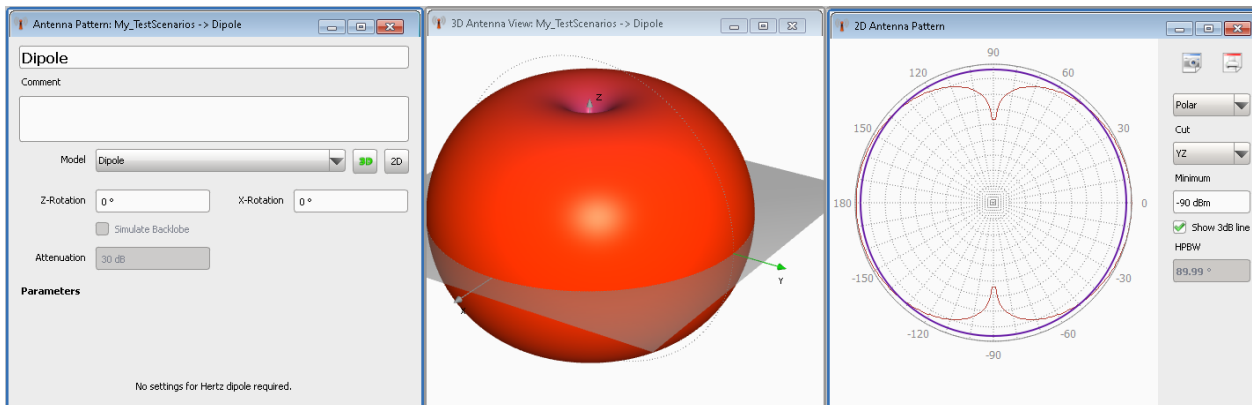
The parameter also indicates the frequency for that the antenna pattern is designed.

Electronic Steering > Z-/X-Rotation ← 2D and 3D diagrams

For "Antenna Model > Custom Phased Array", turns the antenna diagram around the selected axis.

Dipole Antenna Settings

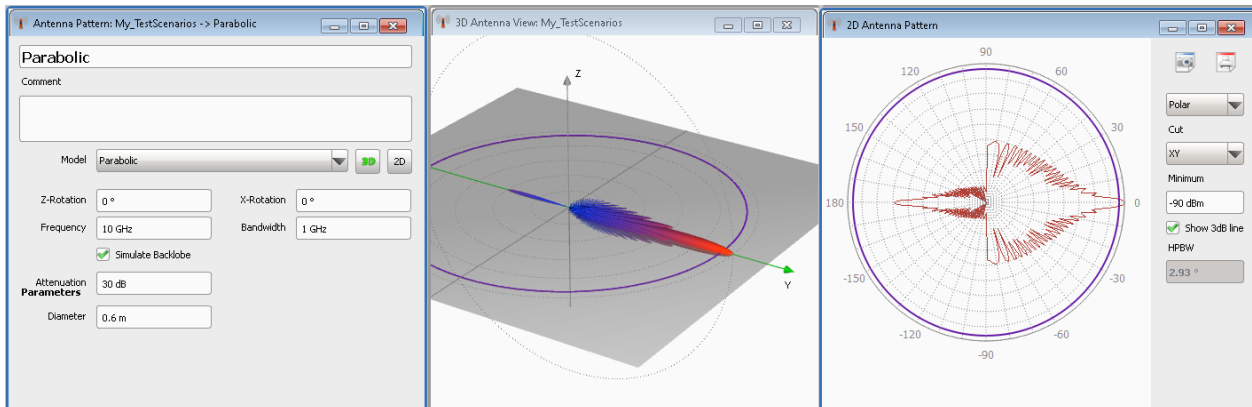
A dipole antenna is a simple antenna with a radiation pattern, shaped like a toroid (doughnut) symmetrical about the axis of the dipole.



Parabolic Antenna Settings

A parabolic antenna has a curved surface with the cross-sectional shape of a parabola with a user defined "Diameter".

Typically, a parabolic antenna radiates the power in a narrow main lobe along the antenna's y-axis.



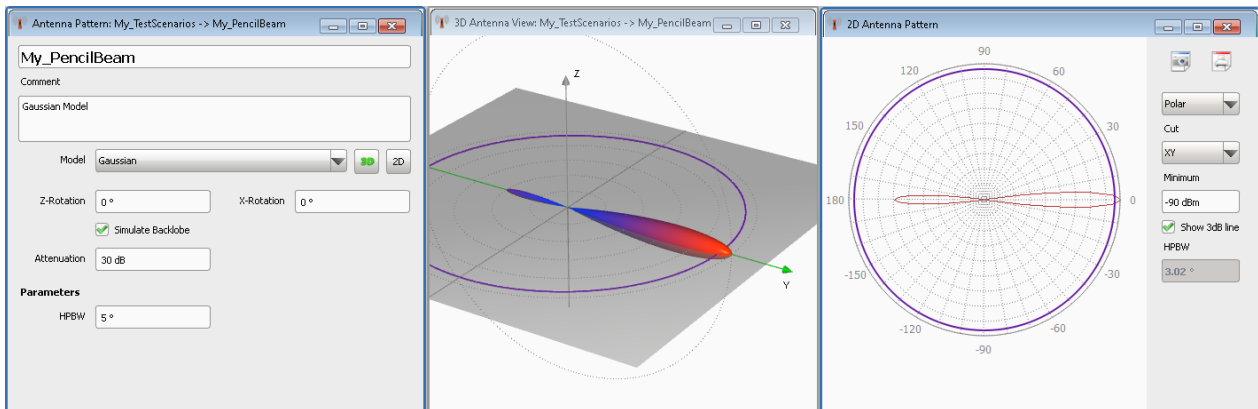
Remote command:

[ANTenna:MODEL:PARabolic:DIAMeter](#) on page 345

Gaussian Antenna Settings

An antenna with radiation pattern that follows the Gaussian distribution.

The main parameter is the **HPBW** that specifies the angular width within which the antenna is most sensitive.



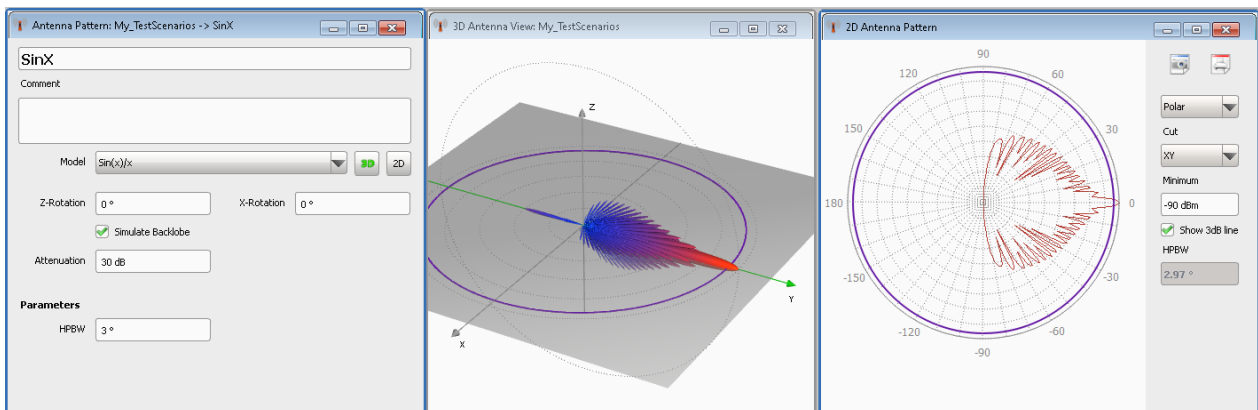
Remote command:

[ANTenna:MODEL:GAUSSian:HPBW](#) on page 345

Sin(x)/x Antenna Settings

An antenna with radiation pattern that follows the Sin(x)/x distribution.

The antenna is characterized by the HPBW.

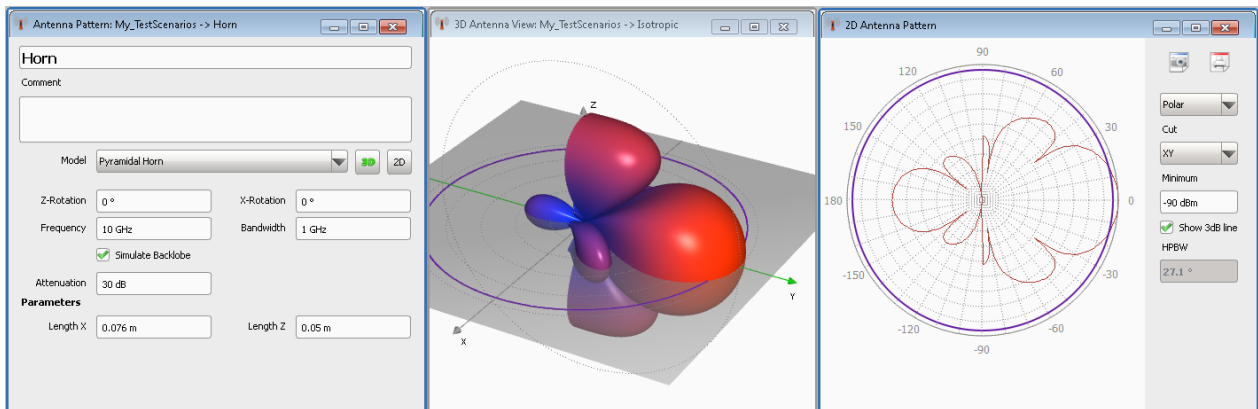


Remote command:

[ANTenna:MODEL:SINC:HPBW](#) on page 345

Pyramidal Horn Antenna Settings

A pyramidal horn antenna has a horn shape in the form of a four-sided pyramid and a rectangular cross section. The length of the rectangular sides "Length X, Length Z" are configurable.



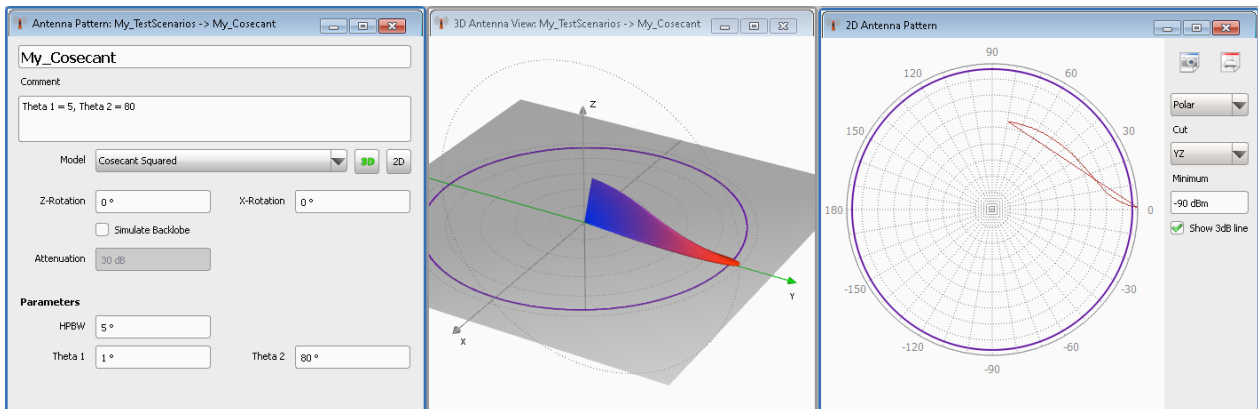
Remote command:

[ANTenna:MODEL:HORN:LX](#) on page 345

[ANTenna:MODEL:HORN:LZ](#) on page 345

Cosecant Squared Antenna Settings

Cosecant squared antennas are designed for air-surveillance radar sets.



Remote command:

[ANTenna:MODEL:COSECANT:HPBW](#) on page 345

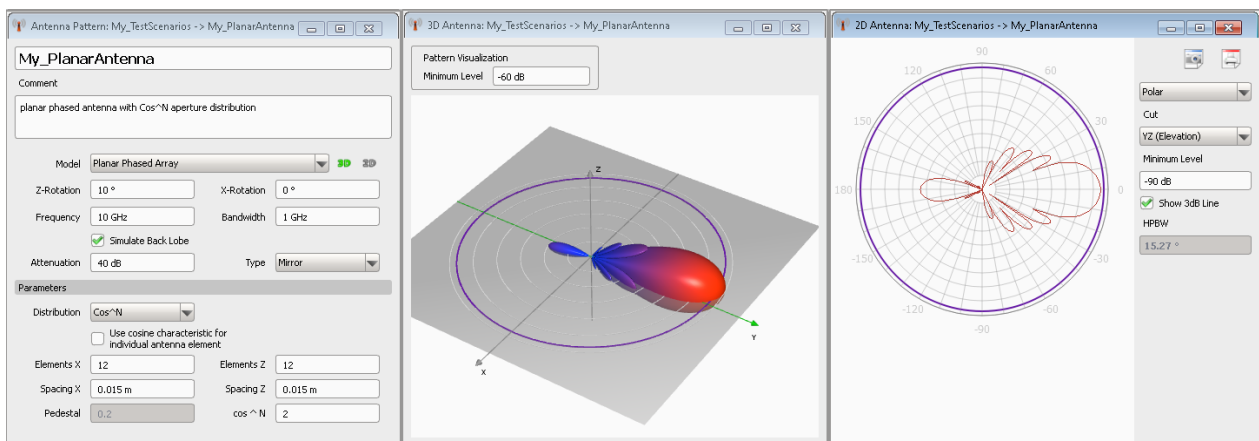
[ANTenna:MODEL:COSECANT:T1](#) on page 342

[ANTenna:MODEL:COSECANT:T2](#) on page 342

Planar Phased Array Antenna Settings

A planar phased array antenna consists of several antenna elements.

The number of elements and the spacing between them is user-definable ("Elements, Spacing"). The exact antenna pattern is defined by the selected "Distribution", the pedestal level and the N value in the \cos^N distribution.



Per default, the individual antenna elements use omnidirectional characteristic. If the parameter "Use cosine characteristic for individual antenna element" is enabled, the cosine characteristic is used.

The characteristics of the individual antenna elements influence the transition between the main/side lobes and the back lobes of the phased array antenna, see [Table 10-4](#).

See also ["To create an antenna pattern"](#) on page 186.

Remote command:

[ANTenna:MODEL:ARRay:DIStribution](#) on page 340

[ANTenna:MODEL:ARRay:ELEMent:COStine](#) on page 340

[ANTenna:MODEL:ARRay:NX](#) on page 340

[ANTenna:MODEL:ARRay:NZ](#) on page 340

[ANTenna:MODEL:ARRay:XDIStance](#) on page 341

[ANTenna:MODEL:ARRay:ZDIStance](#) on page 341

[ANTenna:MODEL:ARRay:PEDestal](#) on page 341

[ANTenna:MODEL:ARRay:COStN](#) on page 340

Custom Antenna Settings

You can define an antenna pattern based on the required antenna characteristics.

See [Figure 10-11](#).

"HPBW XY, HPBW YZ"

HPBW specifies the angular width within which the antenna is most sensitive.

"Side lobe level"

Attenuation applied on the first pair of side lobes compared to the power level of main lobe

"Rolloff factor"

Step size, used to calculate the attenuation of the subsequent side lobes

"Side lobe scale"

Scale factor, used to calculate the HPBW of the side lobes, i.e. how much smaller or wider the side lobes are compared to the main lobe

Remote command:

[ANTenna:MODEL:CUSTom:HPBW:XY](#) on page 343

[ANTenna:MODEL:CUSTom:HPBW:YZ](#) on page 343

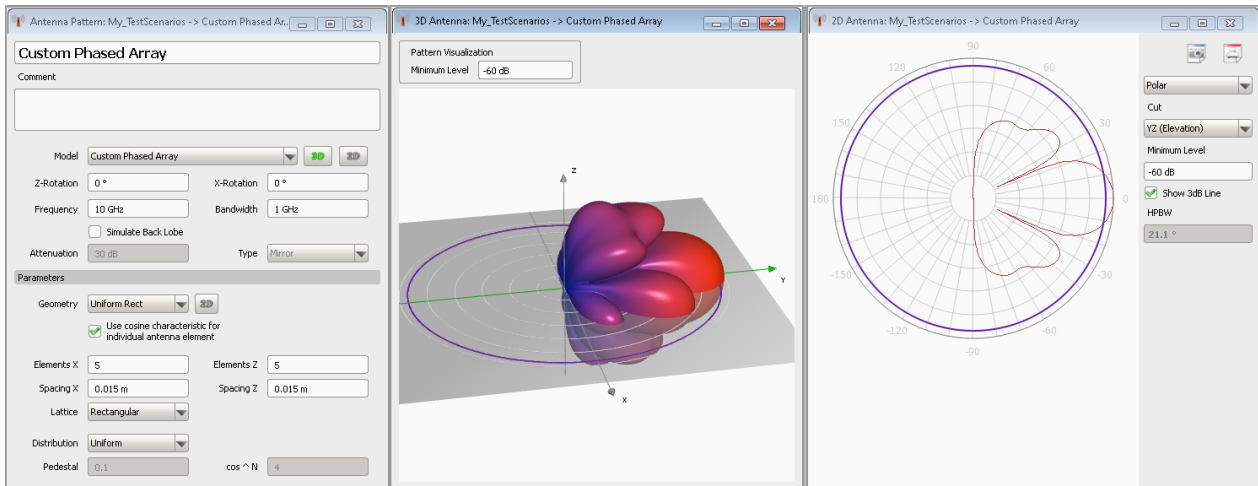
[ANTenna:MODEL:CUSTOM:SLStart](#) on page 344

[ANTenna:MODEL:CUSTOM:SLRolloff](#) on page 344

[ANTenna:MODEL:CUSTOM:SLScale](#) on page 344

Custom Phased Array Settings

You can define a phased array antenna with user-definable geometry, number of elements and spacing between them.



"Geometry" An antenna array is a set of two or more antennas, called antenna elements.

The geometry describes how the single antenna elements of the array are arranged:

- "Uniform Linear" - in a row
- "Uniform Rectangular" - in a rectangular grid
- "Circular Planar" - in a circle
- "Uniform Hex" - as a hexagon

Remote command:

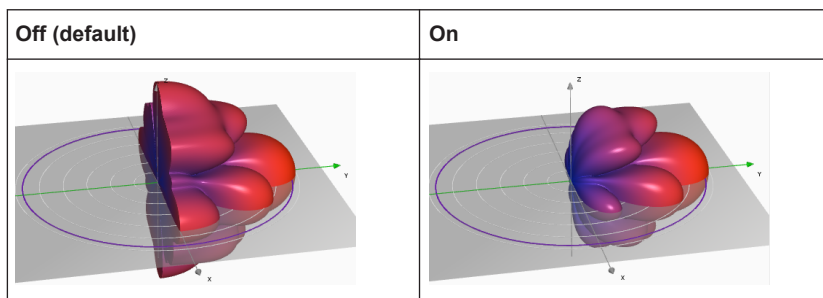
[ANTenna:MODEl:CARRay:GEOMetry](#) on page 343

"Use cosine characteristic for individual antenna element"

Per default, the individual antenna elements use omnidirectional characteristic. If the parameter is enabled, the cosine characteristic is used.

The characteristics of the individual antenna elements influence the transition between the main/side lobes and the back lobes of the phased array antenna, see [Table 10-4](#).

Table 10-4: Effect of the parameter "Use cosine characteristic for individual antenna element"



Remote command:

[ANTenna:MODEl:CARRay:ELEMent:COSSine](#) on page 340

"2D"

Illustrates the array geometry. Click on an antenna element to activate or deactivate it.

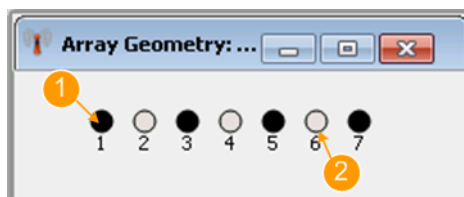


Figure 10-3: Example: Linear array antenna with 7 elements

- 1 = Active element
- 2 = Deactivated element

Remote command:

[ANTenna:MODEl:CARRay:ELEMent](#) on page 343

"Uniform rectangular antenna array"

A rectangular array is a planar antenna array with X number of columns and Z number of rows. The antenna elements can be placed on rectangularly or triangularly lattice.

Per default, the number of elements in the rows and in the columns is the same and the elements are uniformly distributed. You can change the spacing and use unequal number of elements.

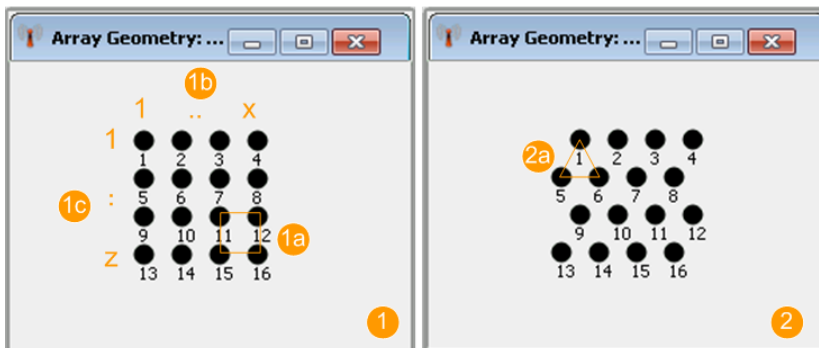


Figure 10-4: Example: Rectangle array antenna with different grids

1, 1a = Rectangular lattice

2, 2a = Triangular lattice

1b, 1c = Number of antenna elements

Remote command:

[ANTenna:MODEl:CARRay:RECTangular:NX](#) on page 340

[ANTenna:MODEl:CARRay:RECTangular:NZ](#) on page 340

[ANTenna:MODEl:CARRay:RECTangular:XDIStance](#) on page 341

[ANTenna:MODEl:CARRay:RECTangular:ZDIStance](#) on page 341

[ANTenna:MODEl:CARRay:RECTangular:LATTice](#) on page 342

"Uniform Linear"

A linear array is a planar array where the antenna elements are placed in a row, see [Figure 10-3](#).

You can change the number of antenna elements and the distance between them. The elements are uniformly distributed.

Remote command:

[ANTenna:MODEl:CARRay:LINear:N](#) on page 340

[ANTenna:MODEl:CARRay:LINear:DIStance](#) on page 341

"Uniform hexagonal antenna array"

This is an array with a hexagonal shape, where each of the six sides has the selected number of antenna elements.

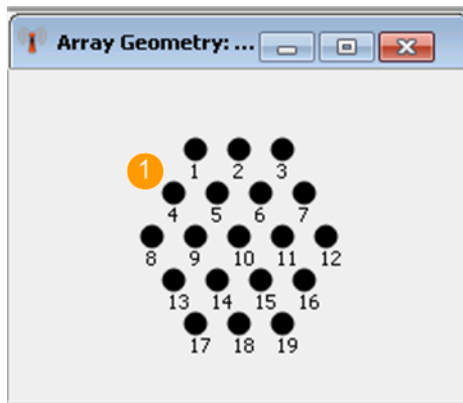


Figure 10-5: Example: Hexagonal array antenna with 3 elements per side

1 = Elements per side

Remote command:

[ANTenna:MODeL:CARRay:HEXagonal:N](#) on page 340

[ANTenna:MODeL:CARRay:HEXagonal:DIStance](#) on page 341

"Planar Circular"

This is an array with a shape similar to a circle, selectable radius size, and lattice.

Remote command:

[ANTenna:MODeL:CARRay:CIRCular:RADius](#) on page 342

[ANTenna:MODeL:CARRay:CIRCular:DIStance](#) on page 341

[ANTenna:MODeL:CARRay:CIRCular:LATTice](#) on page 342

"Distribution" Selects a distribution function.

Remote command:

[ANTenna:MODeL:CARRay:DIStribution](#) on page 340

"Pedestal" Sets the pedestal level.

Remote command:

[ANTenna:MODeL:CARRay:PEDestal](#) on page 341

"cos^N" Sets the N in the cos^N distribution.

Remote command:

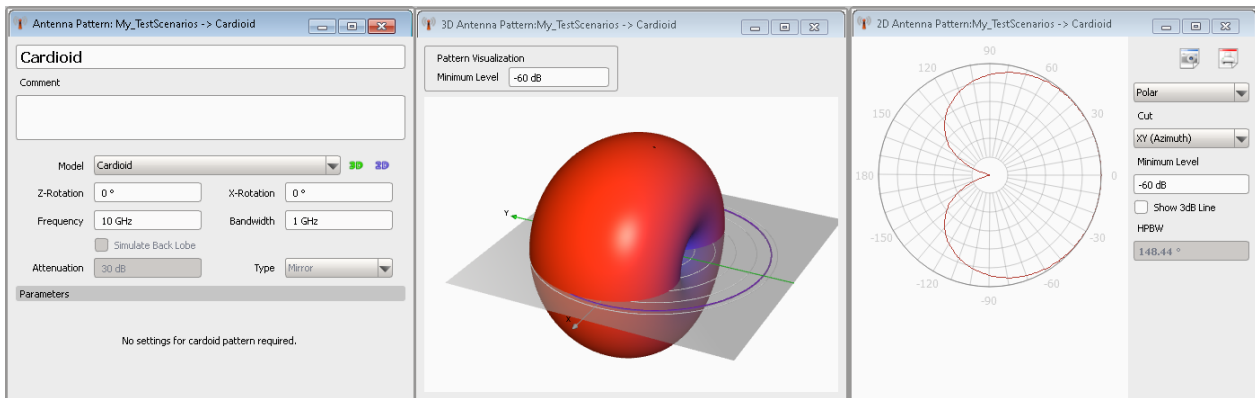
[ANTenna:MODeL:CARRay:COsN](#) on page 340

Cardioid Antenna Settings

A cardioid antenna is an antenna with a heart-like radiation pattern.

The cardioid equation expressed in polar coordinates is:

$r(\varphi) = 1 + \cos(\varphi)$, where φ is the rotation angle.



Imported Antenna Settings

You can define and load your custom antenna patterns.

See:

- [Chapter A.2, "Antenna Pattern File Formats"](#), on page 485 for description of the file formats.
- ["To import a user-defined antenna pattern from file in one of the predefined file formats"](#) on page 191.

Import Filter ← Imported Antenna Settings

Selects the import filter used to parse the antenna pattern file.

"Internal"

Parses antenna patterns in the following file formats:

- FEKO Far Field (*.ffe) files
- *.ant_pat files in the Rohde & Schwarz proprietary format
- Antenna Magus *.tsv files
- ANSYS HFSS *.ffd files
- Antenna patterns in *.csv file format.

The [Import Wizard](#) assists you to import *.csv files.

"Plugin Name"

Any selection other than "Internal" selects a user-defined antenna pattern import filter (plugin). The import filter must exist in the "Plugin" library.

See [Chapter C.4, "Custom Antenna Pattern Import Functions"](#), on page 506.

Antenna patterns in custom file format are files with extension *.txt.

Load/Delete ← Imported Antenna Settings

Standard functions for file management.

Remote command:

[ANTenna:MODEL:USER:LOAD](#) on page 346

[ANTenna:MODEL:USER:CLEAR](#) on page 335

Number of Patterns and Points ← Imported Antenna Settings

Indicates the number of patterns and points as retrieved from the load file.

Import Wizard ← Imported Antenna Settings

When a *.csv file is loaded, the "Import Wizard" opens and tries to determine:

- The used column separator
- The decimal delimiter
- The first row with data
- The data content of each column.

You can change the values afterwards.

Create Waveform From Data

Column Separator: Comma (,)

Decimal Point: Point (.)

First Row: 14

1st Column: Theta

2nd Column: Phi

3rd Column: E Theta Re

4th Column: E Theta Im

5th Column: E Phi Re

6th Column: E Phi Im

```

FRQ=3.0E+9
PIN=0.171146799151093
THETA_SYM_0=0
THETA_SYM_90=0
PHI_SYM_0=0
PHI_SYM_90=0
INTERP_DBI=0
Theta(,deg) Phi(,deg) E_Theta(Re,) E_Theta(Im,) E_Phi(Re,) E_Phi(Im,)

# Far-field exported from Antenna Magus 5.3.0.1686
# On Friday, March 06, 2015 at 1:17:55 PM
# Theta Samples: 181
# Phi Samples: 361
0.000000000000E+000,0.000000000000E+000,4.303687774512E-016,0.000000000000E+0
00,-2.142424624562E+000,-1.426268526806E+000

```

Abort Import

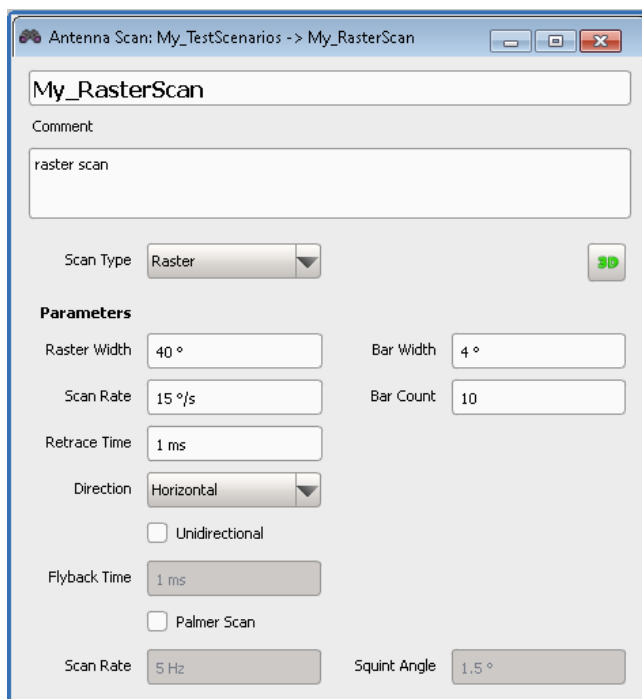
Remote command:

[ANTenna:MODEl:USER:CSV:FORMat](#) on page 346

10.3 Antenna Scans Settings

Access:

- ▶ Select "Repository Tree > Antenna Scan > New".



Available settings:

- Antenna Scan Name..... 180
- Comment..... 180
- Scan Type..... 180
- 3D Scan View..... 180
 - └ Scan Animation Time..... 180
 - └ Scan and Pattern Visualization..... 180
- Circular Scan Settings..... 180
- Sector Scan Settings..... 181
- Raster Scan Settings..... 182
- Conical Scan Settings..... 183
- Helical Scan Settings..... 184
- Spiral Scan Settings..... 184
- Lobe Switching Scan Settings..... 185

Antenna Scan Name

Enter the name of the antenna scan.

Remote command:

[SCAN:NAME](#) on page 331

[SCAN:CATalog?](#) on page 329

[SCAN:SElect](#) on page 330

[SCAN:CREate](#) on page 329

[SCAN:REMove](#) on page 331

Comment

Enter a short description.

Remote command:

[SCAN:COMment](#) on page 331

Scan Type

Defines the scan type.

Use the "3D Scan View" diagram to visualize the selected scan.

Remote command:

[SCAN:TYPE](#) on page 354

3D Scan View

Visualizes the antenna scan in a 3D preview.

See the example on [Figure 10-10](#).

Scan Animation Time ← 3D Scan View

Sets the time it takes the animation to complete a scan.

The "Real Time" is also displayed.

Scan and Pattern Visualization ← 3D Scan View

Defines the way the scan is represented:

- "Scan Line = On": the scan is represented by a line; the antenna pattern is disregarded.
- "Pattern Visualization > Pattern": the visualization uses a Gaussian pattern. If you have opened this dialog from the "Emitter" dialog or the 2D map, the visualization corresponds to the current selected antenna pattern.
- "Pattern Visualization > HPBW": the antenna is represented by its [HPBW](#).
- "Minimum Displayed Level": the minimum displayed side lobes level.

Circular Scan Settings

A circular antenna turns in a circle and scans 360°.

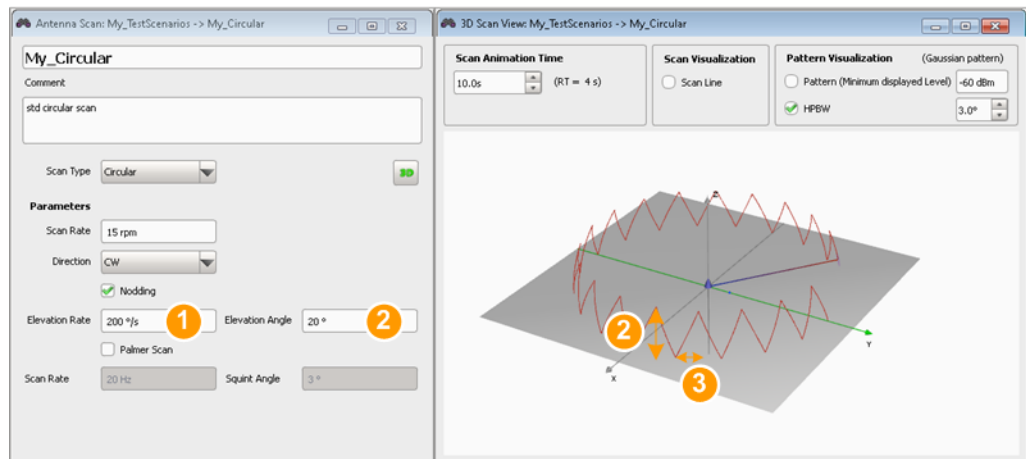


Figure 10-6: Circular scan with enabled nodding: understanding the displayed information

1 = "Elevation Rate"

2 = "Elevation Angle"

3 = Rise Time = "Elevation Angle"/"Elevation Rate"

"Scan Rate" Sets speed the antenna is turning with.

"Direction" Sets the direction, clockwise ("CW") or counter clockwise ("CCW"), the antenna is turning in.

"Nodding" If enabled, superimposes a horizontal nodding on the circular scan. The height of the nods and their rise time are defined with the parameters "Elevation Angle" and "Elevation Rate", see [Figure 10-6](#).

"Palmer Scan" If enabled, superimposes a conical scan on the circular scan, see [Figure 10-7](#).

Remote command:

[SCAN:CIRCular:RPM](#) on page 349

[SCAN:CIRCular:ROtation](#) on page 349

[SCAN:CIRCular:NODDing](#) on page 354

[SCAN:CIRCular:NRATe](#) on page 354

[SCAN:CIRCular:NElevation](#) on page 354

[SCAN:CIRCular:PALMer](#) on page 355

[SCAN:CIRCular:PRATe](#) on page 355

[SCAN:CIRCular:PSQuint](#) on page 355

Sector Scan Settings

A Sector antenna scans only a sector with defined "Sector Width".

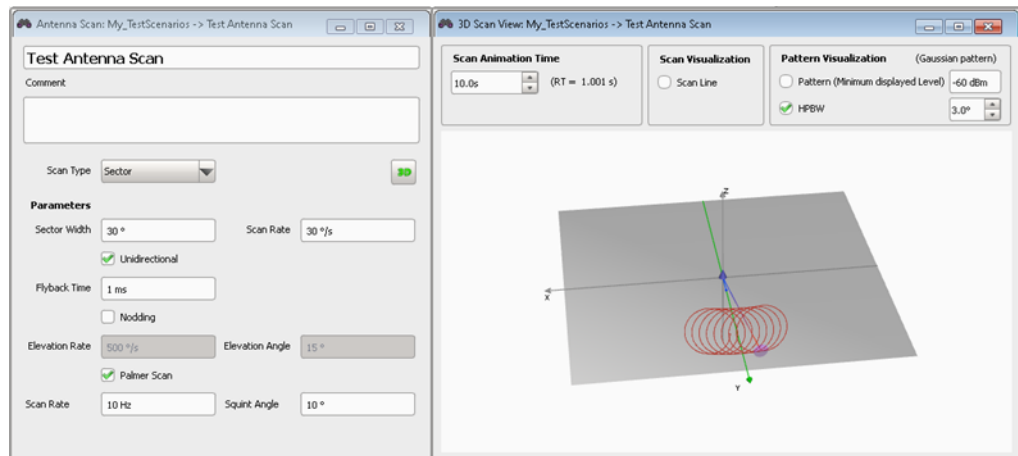


Figure 10-7: Sector scan with enabled Palmer scan

"Sector Width, deg"

Sets the width of the scanned sector.

"Scan Rate, deg/s"

Defines how fast the antenna is turning.

"Unidirectional" The antenna uses a unidirectional scan mode, i.e. the antenna turns in one direction only and "jumps" back within the selected "Flyback Time".

"Nodding"

If enabled, superimposes a horizontal nodding on the circular scan. The height of the nods and their rise time are defined with the parameters "Elevation Angle" and "Elevation Rate", see [Figure 10-6](#).

"Palmer Scan"

If enabled, superimposes a conical scan on the circular scan, see [Figure 10-7](#).

Remote command:

[SCAN:SECTor:WIDTh](#) on page 353

[SCAN:SECTor:RATE](#) on page 350

[SCAN:SECTor:UNIDirection](#) on page 352

[SCAN:SECTor:FLYBack](#) on page 352

[SCAN:SECTor:NODDing](#) on page 354

[SCAN:SECTor:NELevation](#) on page 354

[SCAN:SECTor:NRATe](#) on page 354

[SCAN:SECTor:PALMer](#) on page 355

[SCAN:SECTor:PRATe](#) on page 355

[SCAN:SECTor:PSQuint](#) on page 355

Raster Scan Settings

The raster antenna scans a sector with defined "Sector Width" but performs several scans with different elevations.

See [Figure 10-10](#).

- "Raster Width, deg"
Sets the width of the scanned sector.
Scanned is an angle of \pm "Raster Width"/2 degrees, starting from the Y axis.
- "Scan Rate, deg/s"
Defines how fast the antenna is turning.
- "Bar Count, Bar Width"
Sets the number of bars and the distance between two consecutive scanned bars (sectors).
- "Retrace Time"
Defines how fast the antenna returns to its initial orientation after completing the last bar.
- "Direction"
Switches between a horizontal and a vertical scanning direction.
- "Unidirectional"
The antenna uses a unidirectional scan mode. The scan is not a continuous but the antenna turns in one direction only (counterclockwise, CCW) and "jumps" to the next bar within the selected "Flyback Time".
- "Palmer Scan"
If enabled, superimposes a conical scan on the circular scan, see [Figure 10-7](#).

Remote command:

[SCAN:RASTer:WIDTh](#) on page 353

[SCAN:RASTer:RATE](#) on page 350

[SCAN:RASTer:BARWidth](#) on page 351

[SCAN:RASTer:BARS](#) on page 351

[SCAN:RASTer:RETRace](#) on page 352

[SCAN:RASTer:DIRection](#) on page 353

[SCAN:RASTer:UNIDirection](#) on page 352

[SCAN:RASTer:FLYBack](#) on page 352

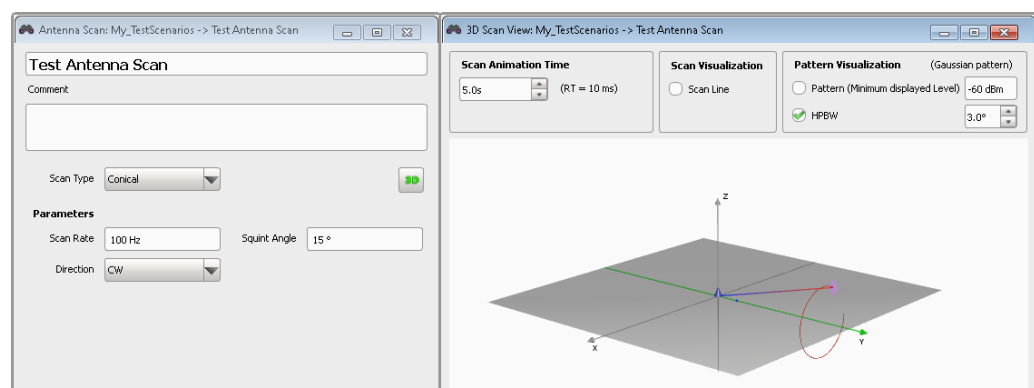
[SCAN:RASTer:PALMer](#) on page 355

[SCAN:RASTer:PRATe](#) on page 355

[SCAN:RASTer:PSQuint](#) on page 355

Conical Scan Settings

A conical antenna scans a conus with defined "Squint Angle" (radius); the radar beam is rotated around the y-axis (i.e. boresight).



- "Scan Rate, Hz"
Defines how fast the antenna is turning.

"Squint Angle, deg"

Sets the radius of the scanned circle.

"Direction"

Sets the direction, clockwise ("CW") or counter clockwise ("CCW"), the antenna is turning in.

Remote command:

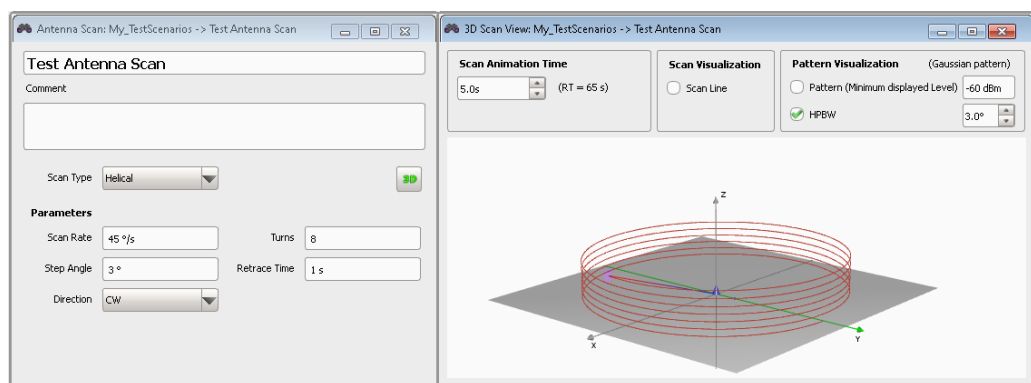
[SCAN:CONical:RATE](#) on page 350

[SCAN:CONical:ROTation](#) on page 349

[SCAN:CONical:SQUint](#) on page 351

Helical Scan Settings

A helical antenna scan turns in a circle (scans 360°) and performs several scans with different elevations.



"Scan Rate, deg/s"

Defines how fast the antenna is turning.

"Step Angle, deg"

Sets the step width with that the antenna azimuth changes.

"Turns"

Sets the number of 360° turns the antenna performs.

"Retrace Time"

Defines how fast the antenna returns to its initial orientation after completing the last turn.

"Direction"

Sets the direction, clockwise ("CW") or counter clockwise ("CCW"), the antenna is turning in.

Remote command:

[SCAN:HELical:RPM](#) on page 349

[SCAN:HELical:ELEVation:STEP](#) on page 350

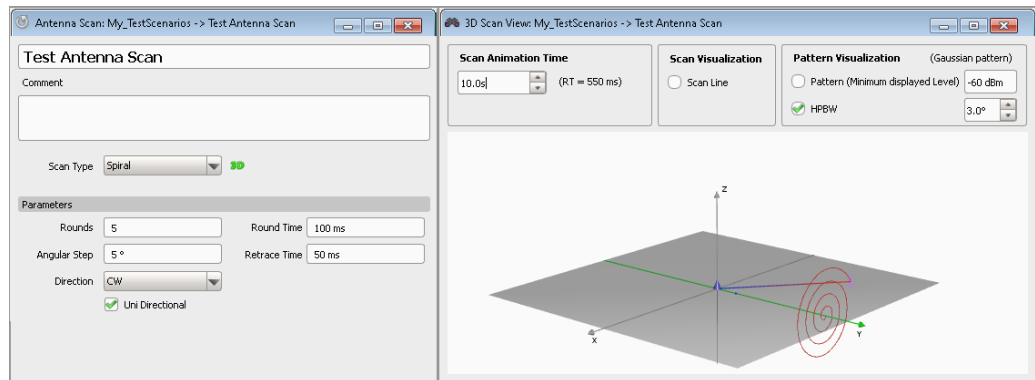
[SCAN:HELical:TURNs](#) on page 350

[SCAN:HELical:RETRace](#) on page 352

[SCAN:HELical:ROTation](#) on page 349

Spiral Scan Settings

A spiral antenna scan turns in a circle (scans 360°) and performs several scans with different radii.



- "Rounds" Sets the number of rounds the antenna performs.
- "Angular Step" Sets the distance in degrees with that the scan radius is increased.
- "Round Time" Defines how fast the antenna is turning.
- "Retrace Time" Define how fast the antenna returns to its initial orientation after completing the last turn.
- "Direction" Sets the direction, clockwise ("CW") or counter clockwise ("CCW"), the antenna is turning in.
- "Uni Directional" Sets the scan behavior when the scan reaches the last round:
- "Uni Directional = On" - the scan is not continuous. It turns for the specified number of rounds and then takes the retrace time to return to the origin of the spiral scan.
 - "Uni Directional = Off" - the scan is continuous; it turns forth and backwards. When the specified number of rounds is reached, the scan direction turns backwards back to the origin of the spiral scan.

Remote command:

[SCAN:SPIRAL:ROUNds](#) on page 353

[SCAN:SPIRAL:STEP](#) on page 353

[SCAN:SPIRAL:RTIME](#) on page 353

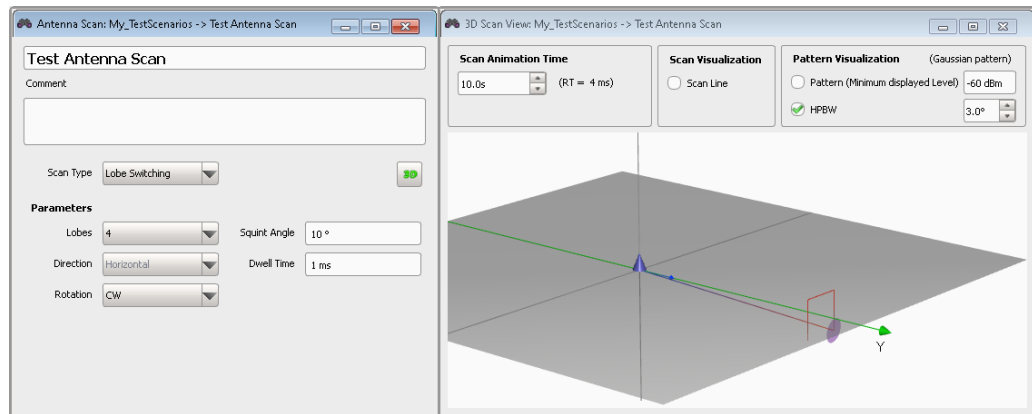
[SCAN:SPIRAL:RETRace](#) on page 352

[SCAN:SPIRAL:ROtation](#) on page 349

[SCAN:SPIRAL:UNIDirection](#) on page 352

Lobe Switching Scan Settings

A lobe switching antenna uses two or four slightly separated antenna elements.



- "Lobes" Set the number of lobes.
- "Direction" Sets the switching direction, i.e. horizontal or vertical.
- "Squint Angle" Sets the angle that the beam axis is offset from the tracking axis (i.e. the y-axis) of the antenna.
- "Dwell Time" Sets the antenna motion, i.e. how fast the switches between the lobes.
- "Direction" Sets the direction, clockwise ("CW") or counter clockwise ("CCW"), the antenna is turning in.

Remote command:

- [SCAN:LSW:LOBes](#) on page 351
- [SCAN:LSW:DIRection](#) on page 350
- [SCAN:LSW:SQUint](#) on page 351
- [SCAN:LSW:DWELL](#) on page 351
- [SCAN:LSW:ROTation](#) on page 349

10.4 How to Create a Library with Antenna Patterns and Scans

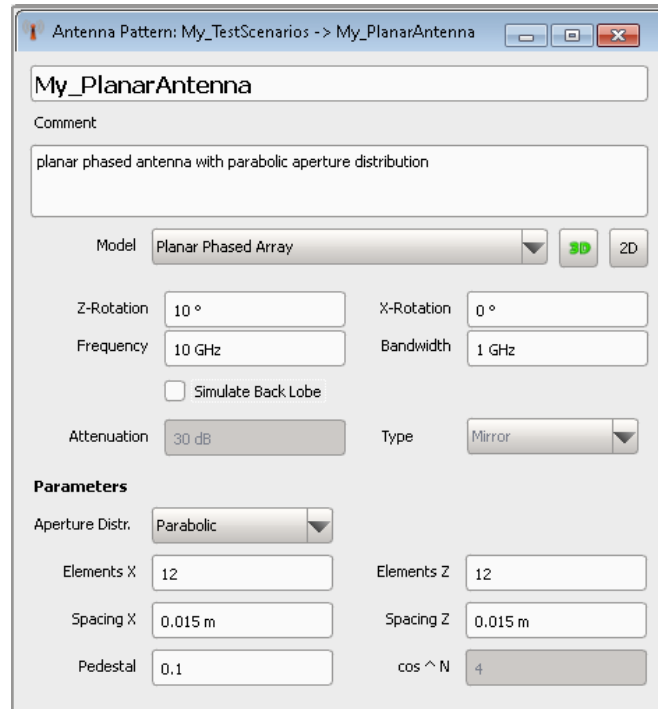
See:

- ["To create an antenna pattern"](#) on page 186
- ["To create an antenna scan"](#) on page 189
- ["To create a custom antenna pattern based on the required antenna characteristics"](#) on page 190
- ["To import a user-defined antenna pattern from file in one of the predefined file formats"](#) on page 191
- ["To import an antenna pattern file in custom file format"](#) on page 193
- ["To observe the effect of polarization"](#) on page 193

To create an antenna pattern

1. Select "Repository Tree > Antenna Pattern > New" .

2. Select "Antenna Model", e.g. "Planar Phased Array".



3. Adjust further settings, like "Rotation", "Frequency" or "Aperture Distribution".
4. Select "2D" to display the antenna pattern in a graph in polar or with Cartesian coordinates.

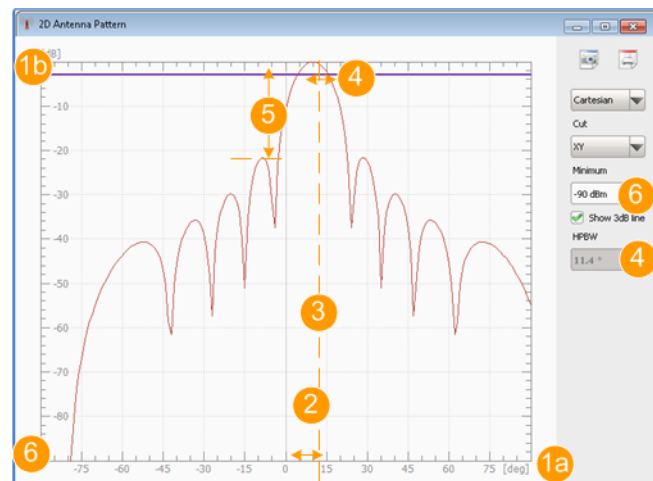


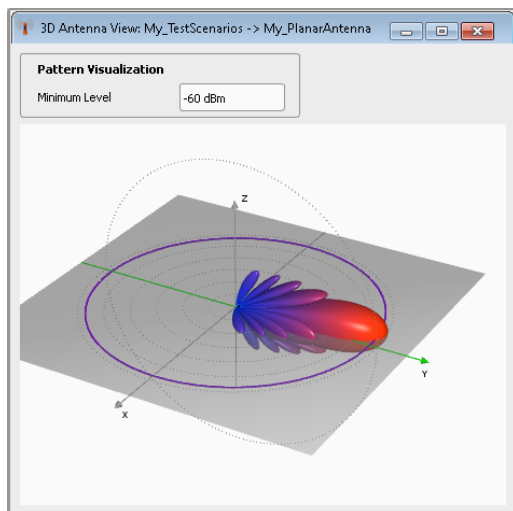
Figure 10-8: 2D antenna pattern: understanding the displayed information

- 1a, 1b = Antenna pattern diagram, as a function of angle (deg) and antenna gain (dB)
- 2 = "Z-Rotation = 10°"
- 3 = Beam axis
- 4 = HPBW, i.e. the beam width at half power (3 dB)
- 5 = Side lobe level
- 6 = Antenna gain minimum (-90 dB)

- Change the number of used elements, e.g. "Elements X = 20" and compare the 2D diagram.

More antenna elements result in a smaller beamwidth and side lobes with lower level.

- Select "Antenna Pattern > 3D" to display the 3D antenna pattern.



The 3D view is interactive. Use the mouse cursor to turn it around any of the axes and the mouse wheel to zoom in the view.

- Select "Antenna Pattern > Simulate Back Lobe > On" and "Attenuation = 0 dB".
- In the 2D view, select "Polar" coordinates.

The display confirms the antenna pattern with a back lobe, where the back lobe is simulated as the mirrored pattern of the main lobe ("Type > Mirror"). See [Figure 10-9](#).

- Select "Antenna Pattern > Back Lobe Attenuation = 40 dB".

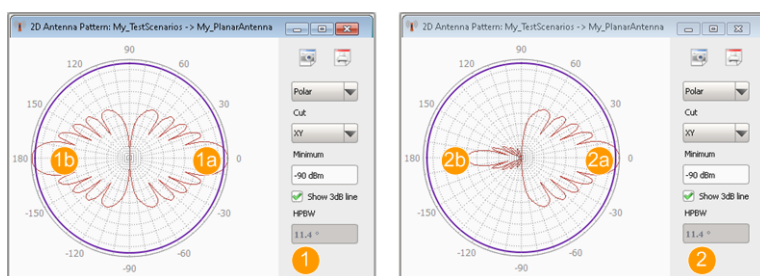


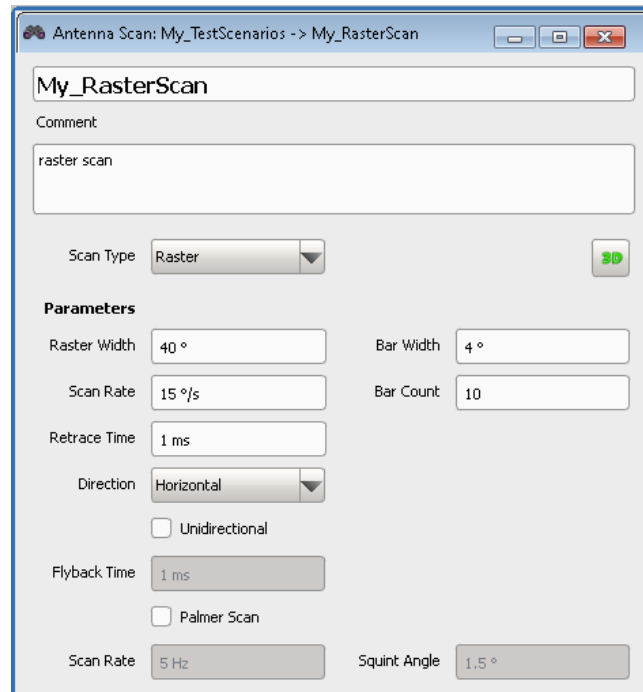
Figure 10-9: Simulate Back Lobe: understanding the displayed information

- 1 = "Attenuation = 0 dB"
- 2 = "Attenuation = 40 dB"
- 1a, 2a = Main lobe
- 1b = Unattenuated back lobe, i.e. the mirrored pattern of the main lobe
- 2b = Attenuated back lobe

For description of the provided settings, see [Chapter 10.2, "Antenna Pattern Settings"](#), on page 164.

To create an antenna scan

1. Select "Repository Tree > Antenna Scans > New".
2. Select "Scan Type", e.g. "Raster Scan".



3. Adjust further settings, like "Raster Width" and "Bar Width".
4. Select "3D" to display the 3D antenna pattern.
5. Use the mouse cursor to turn it around any of the axes and the mouse wheel to zoom in the view.

The antenna scan confirms the settings.

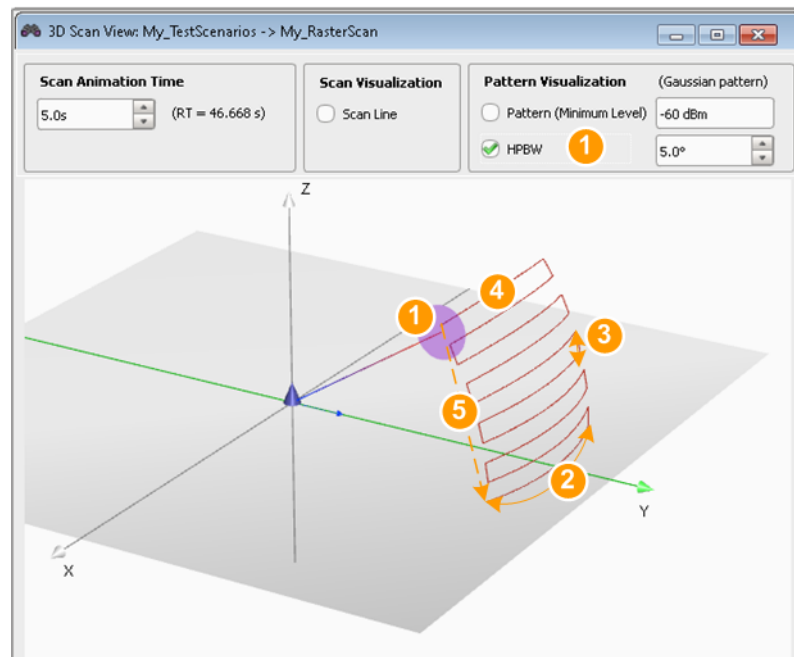


Figure 10-10: Horizontal raster scan: understanding the displayed information

- 1 = Antenna pattern visualized as "HPBW = 5°"
- 2 = "Raster Width"
- 3 = "Bar Width = 4°"
- 4 = 10th scan as defined with "Bar Count"
- 5 = "Retrace Time"

For description of the provided settings, see [Chapter 10.3, "Antenna Scans Settings"](#), on page 179.

To create a custom antenna pattern based on the required antenna characteristics

1. Create an "Antenna Pattern".
See ["To create an antenna pattern"](#) on page 186
2. Select "Antenna Model > Custom".
3. Adjust the settings, e.g. set the "HPBW XY = 10°", "HPBW YZ = 3°", etc.
4. Observe the antenna pattern on the 2D and 3D views.

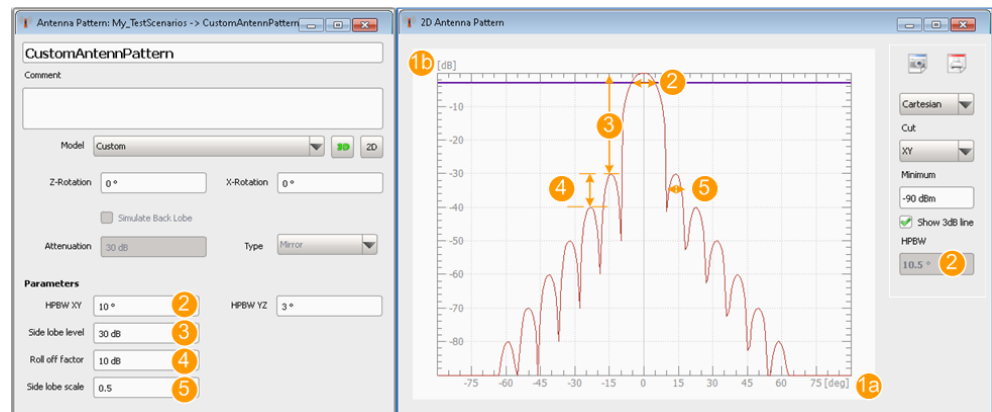


Figure 10-11: Custom antenna pattern: understanding the displayed information

- 1a, 1b = Antenna pattern diagram, as a function of angle (deg) and antenna gain (dB)
 2 = "HPBW XY = 10°", i.e. the main beam width at half power (3 dB)
 3 = Side lobe level, i.e. the attenuation of the first pair of side lobes
 4 = Roll off factor or step size, used to calculate the attenuation of the subsequent side lobes
 5 = Scale factor to calculate the HPBW of the side lobes, i.e. how much smaller or wider the side lobes are compared to the main lobe

To import a user-defined antenna pattern from file in one of the predefined file formats

Per default, you can import custom antenna pattern in the formats listed in [Chapter A. 2, "Antenna Pattern File Formats"](#), on page 485.

1. Create an "Antenna Pattern".
2. Select "Antenna Model > Import from File".
3. Select "Import Filter > internal".
4. Select "Load", navigate to the folder with your custom antenna patterns and load a *.ffe, a *.tsv, a *.ffd, a *.csv, or a *.ant_pat file.
5. Open the 2D view to visualize the antenna pattern.

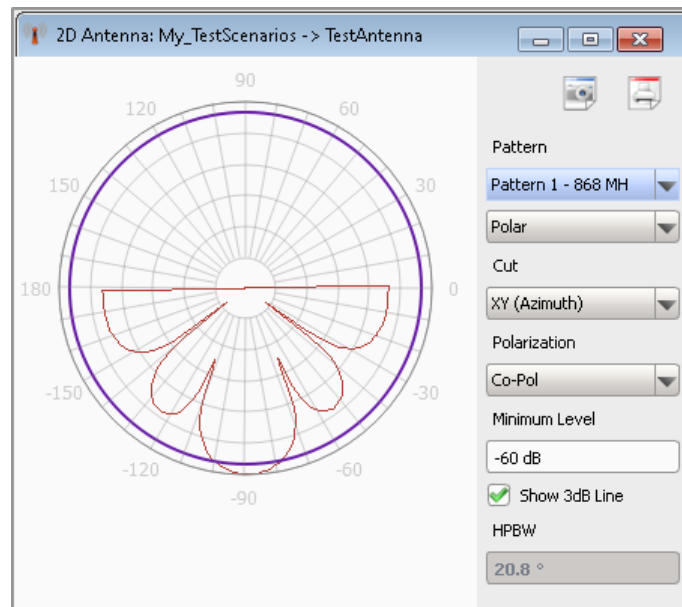
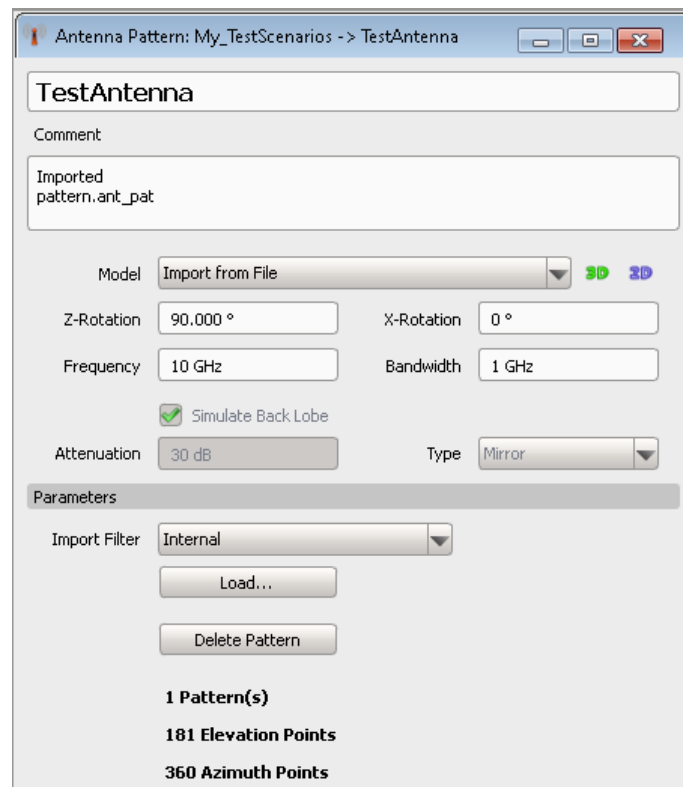


Figure 10-12: 2D view: Example of a user-defined antenna pattern

In this example, the antenna pattern is directed to -90° .

6. Select "Antenna Pattern > Z-Rotation = 90° ".



- On the 2D and 3D views, observe the effect of this parameter.

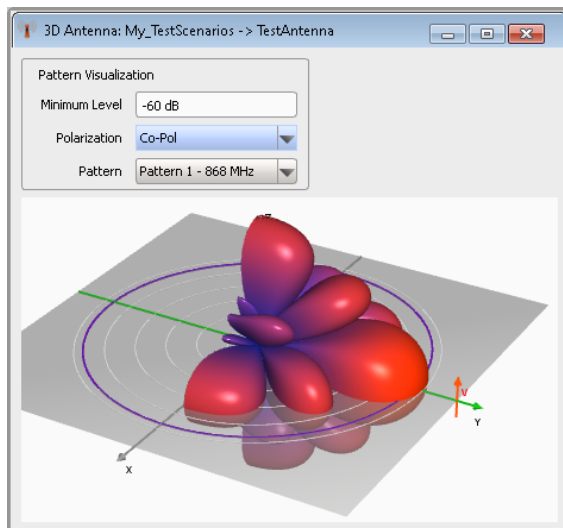


Figure 10-13: 3D view: Example of a user-defined antenna pattern

The antenna pattern is directed on the y-axis.

To import an antenna pattern file in custom file format

Importing antenna patterns in custom file format require that you create a suitable import filter first. That is, that you define and import the file format as a plugin.

- Import the plugin in the repository, for example `My_AntPatternFileFormat`. See ["To import a plugin"](#) on page 297
- Create an antenna pattern and select "Model > Import from File".
- Select "Import Filter > My_AntPatternFileFormat".
- Select "Load", navigate to the folder with your custom antenna pattern (*.txt file) and select it.
- Open the 2D and 3D view to visualize the antenna pattern.

If the file contains more than one antenna patterns, select the pattern to be displayed.

To observe the effect of polarization

- Create an antenna with a Gaussian beam pattern and a vertical polarization.
- Open the 3D display and observe the pattern for "Polarization > Co-Pol" and "Minimum Level = -90 dB".
- Select "Polarization > Cross-Pol".

The automatically selected cross-polarized antenna is an antenna with *horizontal (H)* polarization.

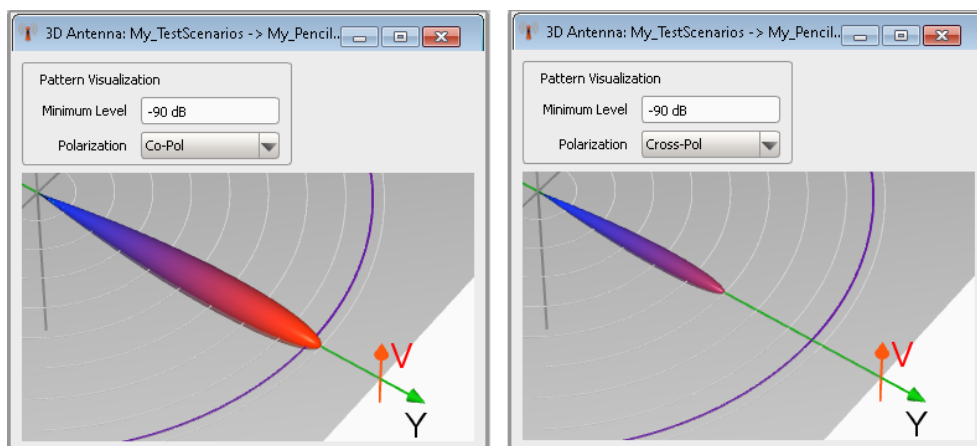


Figure 10-14: Effect of cross-polarization

The left figure shows the Gaussian antenna pattern. The right figure shows the resulting antenna pattern. This is a pattern with the same shape but with maximum power that is 30 dB lower than the co-polarized case. That is, the applied *polarization loss is -30 dB*.

4. Create a 2D scenario with one emitter and a receiver.
5. Assign the Gaussian beam pattern to the emitter and the receiver.
To simplify the explanation, this example does not use an antenna scan.
Note: Antenna polarization is not considered, if the receiver uses an isotropic omnidirectional antenna.
6. Align the receiver and the emitter: for the emitter, set "Altitude > Point to Receiver > On" and "Roll = 0°".
7. Open the "3D View with Receiver" and observe the maximum normalized power level at the receiver.

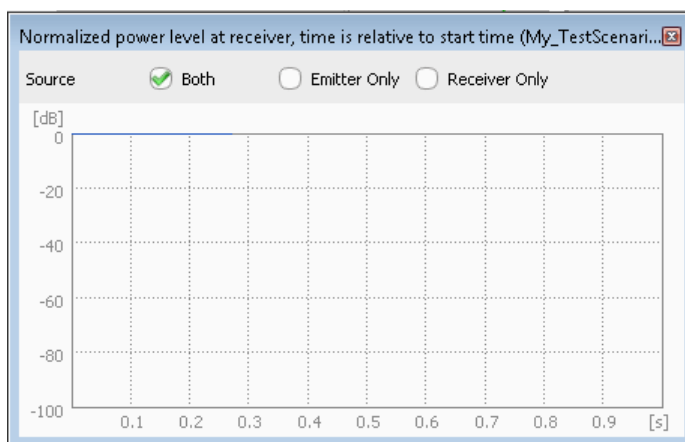
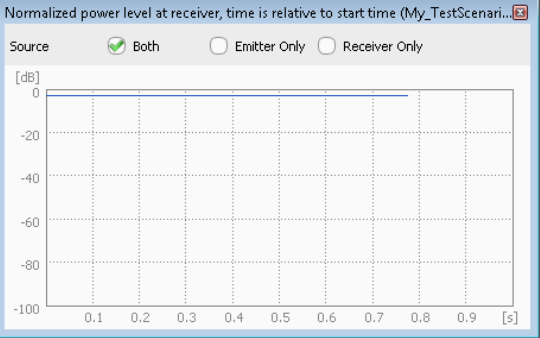
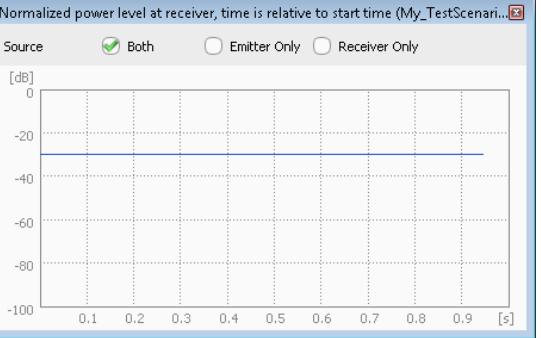


Figure 10-15: Normalized power level at the receiver if the antennas are co-polarized (Roll = 0°)

The maximum power level is 0 dB.

8. Change the antenna roll and compare the maximum normalized power level at the receiver.

Normalized power level at the receiver for "Roll = 45°"	Normalized power level at the receiver for "Roll = 90°"
 <p>Normalized power level at receiver, time is relative to start time (My_TestScenari...)</p> <p>Source <input checked="" type="checkbox"/> Both <input type="checkbox"/> Emitter Only <input type="checkbox"/> Receiver Only</p>	 <p>Normalized power level at receiver, time is relative to start time (My_TestScenari...)</p> <p>Source <input checked="" type="checkbox"/> Both <input type="checkbox"/> Emitter Only <input type="checkbox"/> Receiver Only</p>
<p>The maximum power level is 3 dB less than the co-polarized case.</p>	<p>The antennas are cross-polarized. The maximum power level is 30 dB less than the co-polarized case.</p>

See also:

- [Table 10-2](#)
- [Chapter 15.5, "How to Create Scenarios with Emitters, Interferers and a Receiver"](#), on page 258

11 Emulating Emitters

In the context of this software, a simple emitter describes a pulse sequence whereas a complex emitter emulates a radar system. The emitters are characterized with parameters like operating mode, beam, waveform, antenna pattern, and antenna scan.

The available emitter settings depend on the complexity of the simulated system:

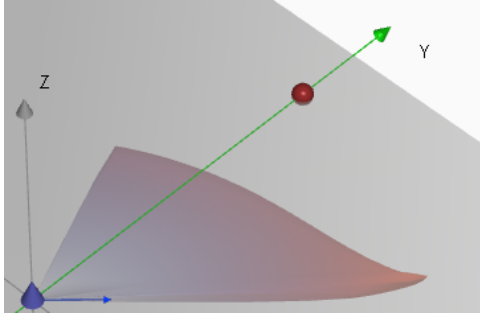
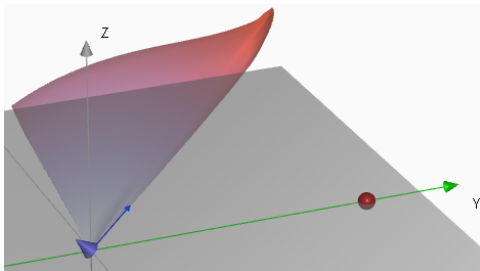
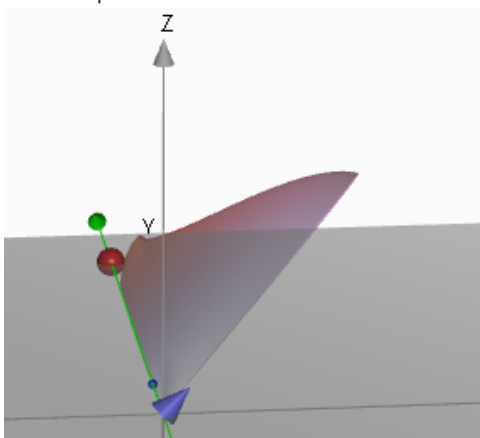
- Simple emitters are described only by a sequence of pulses. Antenna beam patterns and antenna scans are not required
- Complex emitters are described by the combination of an operating mode, antenna pattern, antenna scan and a sequence, as well as attitude information.

Emitter attitude

The attitude is a description of the emitter and interferer orientation. Emitters and interferers can be rotated in 3 dimensions: yaw, pitch and roll.

The examples in [Table 11-1](#) show an emitter with a cosecant antenna pattern in a single emitter scenario. In direction finding scenarios and scenarios with multiple emitters, the effect is the same.

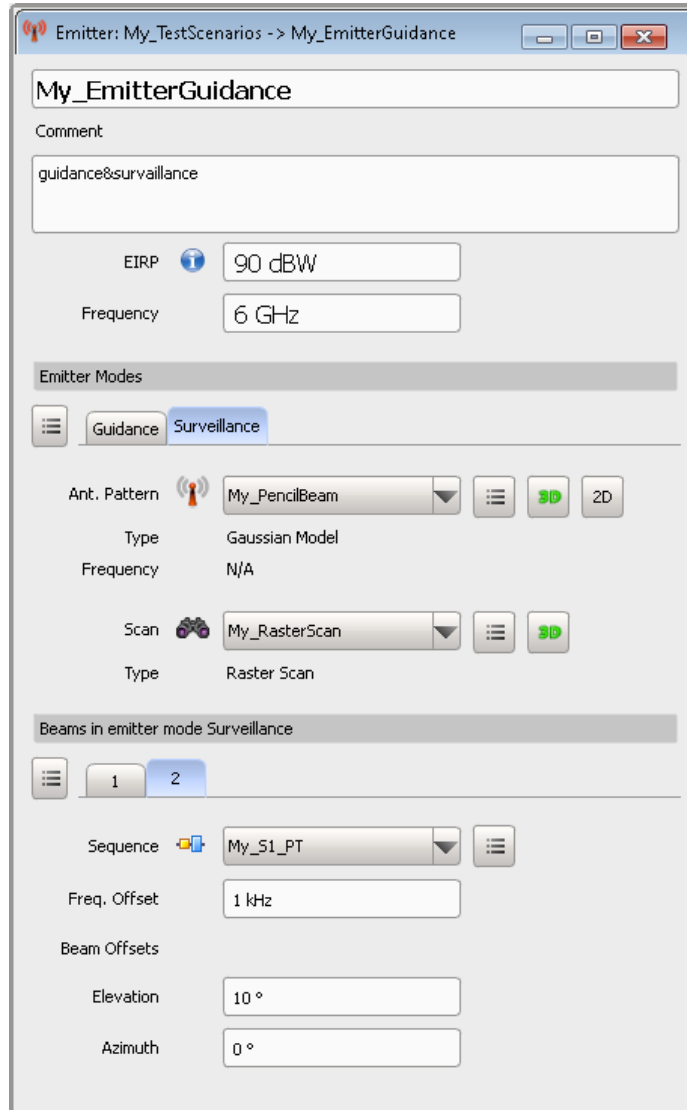
Table 11-1: Impact of the yaw, pitch and roll angles on the emitter attitude

Attitude	Rotation around axis	Rotation
Yaw	Z-axis	<p>In the x/y plane, to the left/right</p>  <p>Yaw = 45° (Pitch = Roll = 0°)</p>
Pitch	X-axis	<p>In the y/z plane, up/down</p>  <p>Pitch = 45° (Yaw = Roll = 0°)</p>
Roll	Y-axis	<p>In the x/z plane</p>  <p>Roll = 45° (Yaw = Pitch = 0°)</p>

11.1 Emitter Settings

Access:

- ▶ Select "Repository Tree > Emitter > New".



An emitter is described as a combination of an operating mode, an antenna pattern, an antenna scan, and a sequence. Complex emitters support several beams per operating mode.

Settings:

Emitter Name.....	199
Comment.....	199
EIRP.....	199

Frequency.....	199
Emitter Modes.....	199
Emitter Beams Definition.....	200

Emitter Name

Enter the emitter's name.

Remote command:

`EMITter:NAME` on page 330

`EMITter:CREate` on page 329

`EMITter:SElect` on page 330

`EMITter:CATalog?` on page 329

`EMITter:REMove` on page 331

Comment

Enter a short description.

Remote command:

`EMITter:COMMeNt` on page 331

EIRP

Sets the equivalent isotropically radiated power (EIRP) of the emitter.

This parameter is used, if the emitter is configured in a "Scenario Type > Localized Emitter".

Example:

For emitter with:

- Transmit power = 1 MW, i.e. 60 dBW
- Antenna Gain = 20 dBi

Set "EIRP = 80 dBW".

Remote command:

`EMITter:EIRP` on page 366

Frequency

Sets the emitter's operating frequency.

If the selected antenna is calculated for a particular frequency, set the emitter's "Frequency" to a value within the suggested frequency range, see "Emitter Modes" on page 199.

Remote command:

`EMITter:FREQuency` on page 367

Emitter Modes

Defines one or more modes the emitter works in.

Use the standard functions in the context menu to create or rename modes.

Each emitter mode is defined by the combination of an antenna pattern, antenna scan and at least one beam with assigned pulse sequence.

Use the 3D view to visualize the selected antenna scan and pattern, see:

- ["2D and 3D diagrams"](#) on page 168
- ["3D Scan View"](#) on page 180

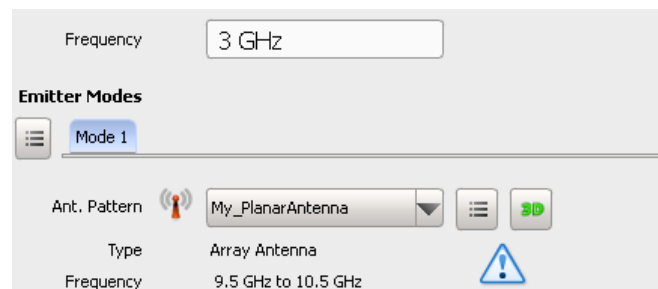
"Antenna Pattern"

Selects an existing antenna pattern or accesses a dialog for creating a new one.

See ["To create an antenna pattern"](#) on page 186.

The antenna pattern and type are displayed.

A warning symbol indicates that the selected emitter's [Frequency](#) is not sufficient for the current antenna.



Set the emitter's [Frequency](#) to a value within the suggested frequency range.

"Antenna Scan"

Selects an existing antenna scan or accesses a dialog for creating a new one.

See ["To create an antenna scan"](#) on page 189

Remote command:

[EMITter:MODE:NAME](#) on page 330

[EMITter:MODE:ADD](#) on page 332

[EMITter:MODE:COUNT?](#) on page 333

[EMITter:MODE:SElect](#) on page 333

[EMITter:MODE:CLEar](#) on page 335

[EMITter:MODE:DELeTe](#) on page 334

[EMITter:MODE:ANTenna](#) on page 367

[EMITter:MODE:ANTenna:CLEar](#) on page 335

[EMITter:MODE:SCAN](#) on page 368

[EMITter:MODE:SCAN:CLEar](#) on page 335

Emitter Beams Definition

In each operating mode, the emitter may use one or more beams (e.g. with different pulse sequences), while using the same antenna and scan type. Emitters use multiple beams for exact position detection, especially when the distance to the objects and beamwidth increase.

Use the standard functions in the context menu to create or rename beams.

Assign a pulse sequence to each beam.

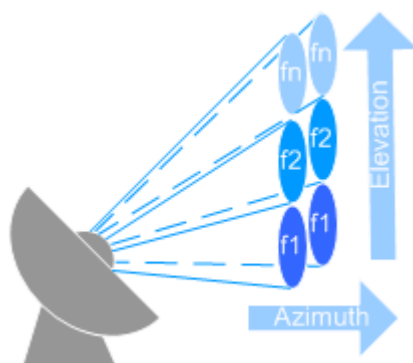


Figure 11-1: Stacked beams (simplified representation)

- "Activate" Activates a beam.
- "Sequence" Selects an existing sequence or accesses a dialog for creating a new one.
See [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124
- "Frequency Offset" Offsets the beam frequency by the selected value, see [Figure 11-1](#).
- "Beam Position Offset" Offsets the beam in terms of an "Elevation" and an "Azimuth", see [Figure 11-1](#).
To *offset the complete emitter's antenna*, use one of the "Elevation" parameters, depending on your current scenario:
- "Single Emitter" scenario: [Elevation](#)
(see ["To configure an emitter in a scenario with static receiver"](#) on page 206)
 - "Emitters (Collection)" scenario: [Elevation](#)
(see ["To configure complex scenarios with several emitters"](#) on page 208)
 - Localized Emitters scenario: [Elevation](#)
(see ["To enable static emitters and place them on the 2D map"](#) on page 261)

Remote command:

[EMITter:MODE:BEAM:ADD](#) on page 332

[EMITter:MODE:BEAM:COUNT?](#) on page 333

[EMITter:MODE:BEAM:SELEct](#) on page 333

[EMITter:MODE:BEAM:STATe](#) on page 368

[EMITter:MODE:BEAM:SEQuence](#) on page 368

[EMITter:MODE:BEAM:OFFSet:AZIMuth](#) on page 367

[EMITter:MODE:BEAM:OFFSet:ELEVation](#) on page 367

[EMITter:MODE:BEAM:OFFSet:FREQuency](#) on page 367

[EMITter:MODE:BEAM:CLEar](#) on page 335

[EMITter:MODE:BEAM:DELEte](#) on page 334

11.2 3D Emitter Preview Settings

Access:

1. Open a "Single Emitter" or an "Emitter (Collection)" scenario.
2. In the block diagram, select "3D".

The dialog display:

- A 3D view of the emitter antenna with its pattern and scan
- A live plot of the normalized signal power level at the receiver

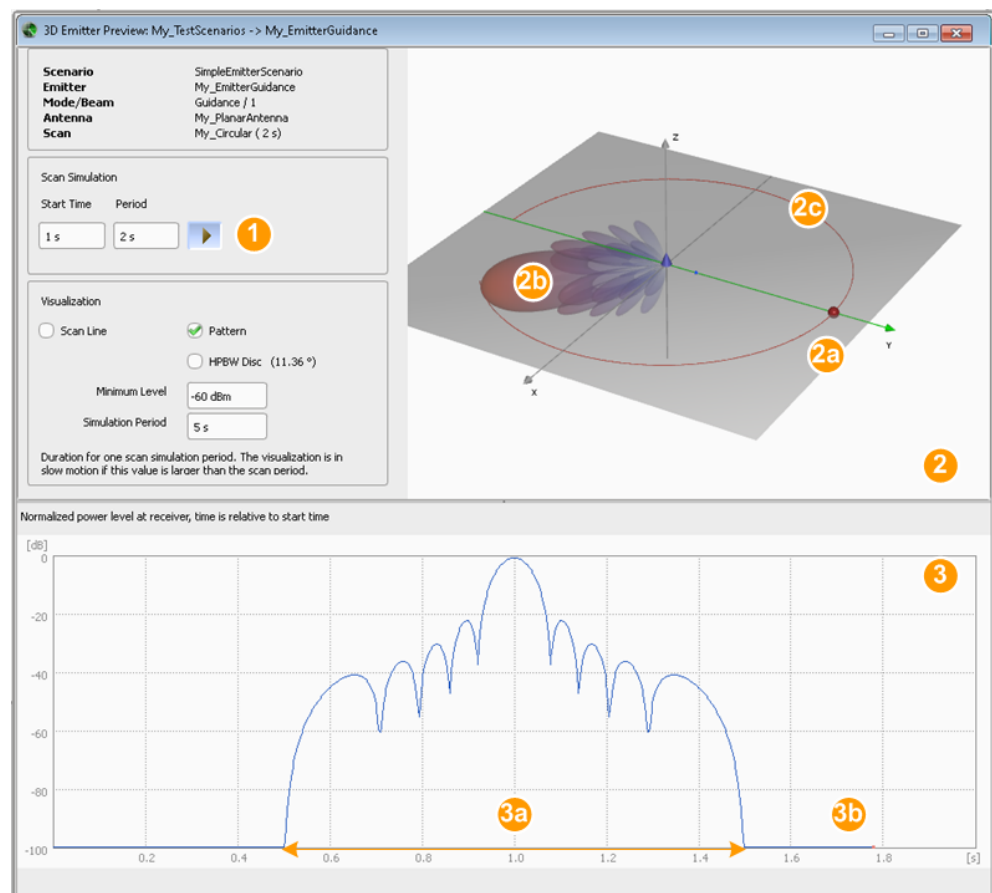


Figure 11-2: 3D Emitter Preview: Understanding the displayed information

- 1 = Selection of preview time span (selected is one antenna turn)
- 2 = Interactive 3D view
- 2a = Receiver with an isotropic antenna pattern, no antenna scan and at a fixed location
- 2b = Antenna pattern diagram of the emitter; antenna pattern without back lobes
- 2c = Circular antenna scan
- 3 = Live plot of the normalized signal power level at the receiver
- 3a = Antenna pattern (main and side lobes), compare with the pattern on [Figure 10-8](#)
- 3b = Antenna pattern without back lobes

Available settings:

Scan Simulation	203
Visualization	203

Scan Simulation

Sets the time span visualized on the live plot as a start time and simulation period.

The "Period", i.e. the total duration the antenna scan simulation, determines the max. value on the x-axis. The visualization is relative to the selected "Start Time".

Use the "Pause" icon to stop the visualization.

Visualization

Defines the way the scan is represented:

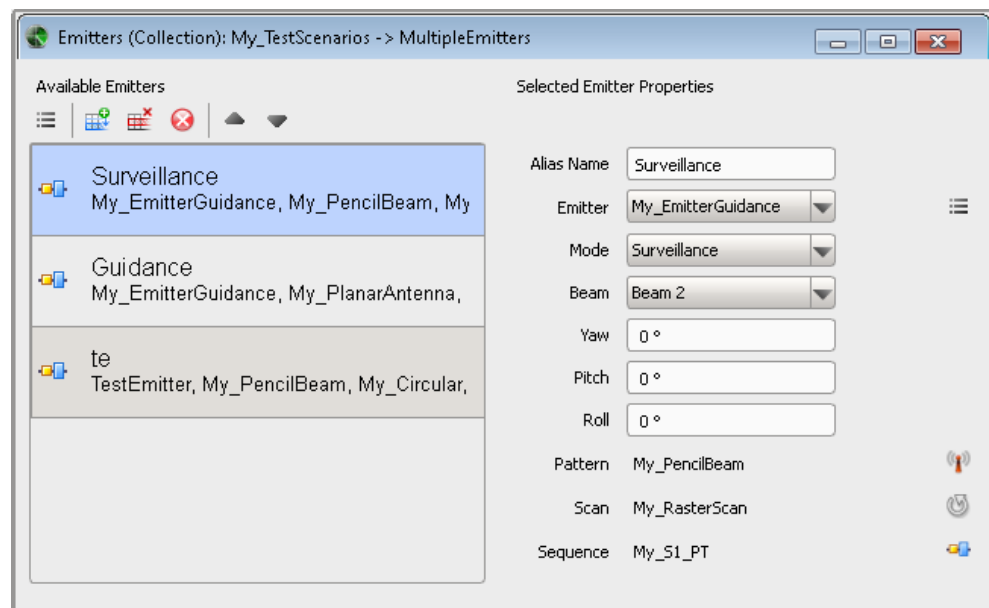
"Scan Line"	The scan is visualized by a line; the antenna pattern is disregarded.
"Pattern"	The visualization corresponds to the current selected antenna pattern.
"HPBW"	The antenna is represented by its HPBW . See also "To visualize the signal received by a static receiver" on page 207.
"Minimum displayed level"	Defines the minimum displayed side lobes level.
"Simulation Period"	Sets the time it takes the animation to complete a scan.

11.3 Lists with Multiple Emitters

Access:

1. Open an "Emitter (Collection)" scenario.
2. In the block diagram, select "Emitters > Edit".

The "Emitter (Collection)" scenario comprises several emitters. They are described in a list form.



Available settings:

List of Available Emitters	204
Properties of the selected Emitter	204

List of Available Emitters

Displays a list of emitters.

Use the standard functions in the context menu and the icons to add, reorder, or remove items.

See also "[To configure complex scenarios with several emitters](#)" on page 208.

Remote command:

[SCENario:CEMit:ADD](#) on page 332

[SCENario:CEMit:SElect](#) on page 333

[SCENario:CEMit:DElete](#) on page 335

[SCENario:CEMit:CLEar](#) on page 335

Properties of the selected Emitter

Each emitter is described with the following parameters:

"Alias Name" Enters an alias name.

"Emitter Name"
Selects an emitter/interferer.

"Mode" Selects the mode the emitter is working in.

"Beam" Sets the number of the currently used beam.

"Pitch, Yaw, Roll"
Offsets the antenna in terms of a "Pitch", a "Yaw" and a "Roll", see "[Emitter attitude](#)" on page 196.

"Antenna Pattern, Scan, Sequence"

Displays the current selected antenna pattern, antenna scan, and sequence.

To change any of them, select the corresponding icon.

Remote command:

`SCENario:CEMit:ALIAS` on page 433

`SCENario:CEMit:EMITter` on page 435

`SCENario:CEMit:EMITter:MODE` on page 435

`SCENario:CEMit:EMITter:MODE:BEAM` on page 436

`SCENario:CEMit:DIRection:PItCh` on page 434

`SCENario:CEMit:DIRection:YAW` on page 434

`SCENario:CEMit:DIRection:ROLL` on page 435

11.4 How to Create and Configure Emitters

See:

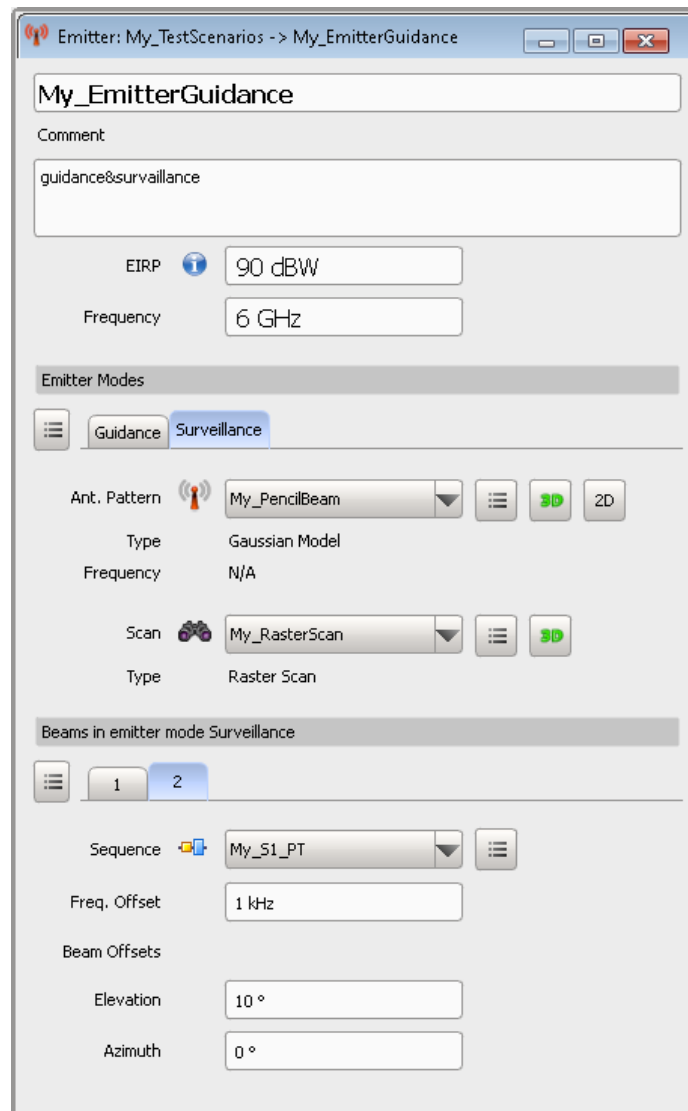
- ["To create and configure a new emitter"](#) on page 205
- ["To configure an emitter in a scenario with static receiver"](#) on page 206
- ["To visualize the signal received by a static receiver"](#) on page 207
- ["To configure complex scenarios with several emitters"](#) on page 208
- ["To enable static emitters and place them on the 2D map"](#) on page 261

To create and configure a new emitter

You can clone or copy an existing emitter or create a new one:

1. Select "Repository Tree > Emitter > New".
2. Enter a name and a comment.
3. Select "Mode 1", open the context menu, select "Rename" and enter a name for the emitter mode, e.g. "Guidance".
4. Select "Ant. Pattern" and select an antenna pattern from the antenna library or create a new one.
5. Select "Scan Type" and select an antenna scan from the antenna library or create a new one.
6. To configure the "Beam":
 - a) Select "Activate > On"
 - b) Select "Sequence" and select a sequence from the library.

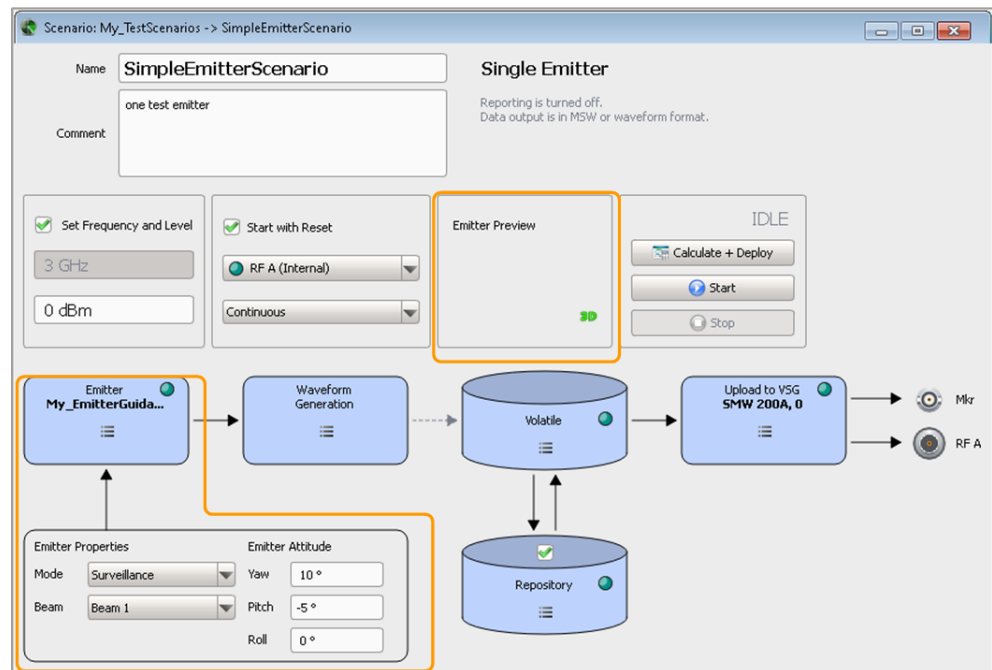
7. If required, enable a second beam.



For description of the provided settings, see [Chapter 11.1, "Emitter Settings"](#), on page 198.

To configure an emitter in a scenario with static receiver

1. Create a new "Single Emitter" scenario or open an existing one.



2. In the block diagram, select "Emitters", open the context menu and select an emitter from the list of available emitters or create a new one.
3. Configure the emitter settings, like for example select the current "Emitter Mode" and "Emitter Beam".

For description of the provided settings, see [Chapter 11.3, "Lists with Multiple Emitters"](#), on page 203.

To visualize the signal received by a static receiver

1. Open a scenario with single emitter, e.g. "SimpleEmitterScenario".
2. In the block diagram, select "Current Emitter > 3D".
The dialog displays the signal as "seen" by a static receiver. Per default, the receiver is located in a way, that the emitter scan is oriented with 0 deg "Pitch" and "Yaw" at it.
3. To "move" the receiver on the x-axis and on the z-axis, enable "Scenario > Pitch" and "Scenario > Yaw". The position on the y-axis (the distance to the emitter) is fixed.

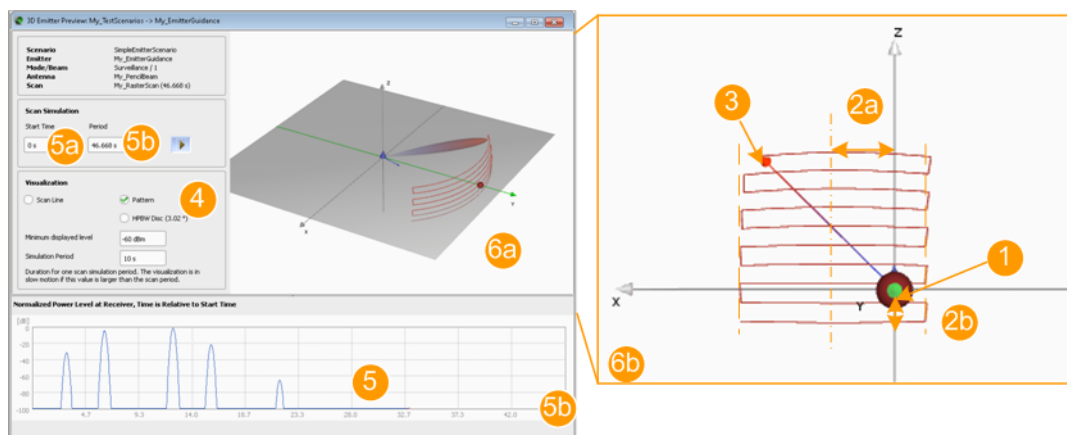


Figure 11-3: Signal at the receiver (XZ view): understanding the displayed information

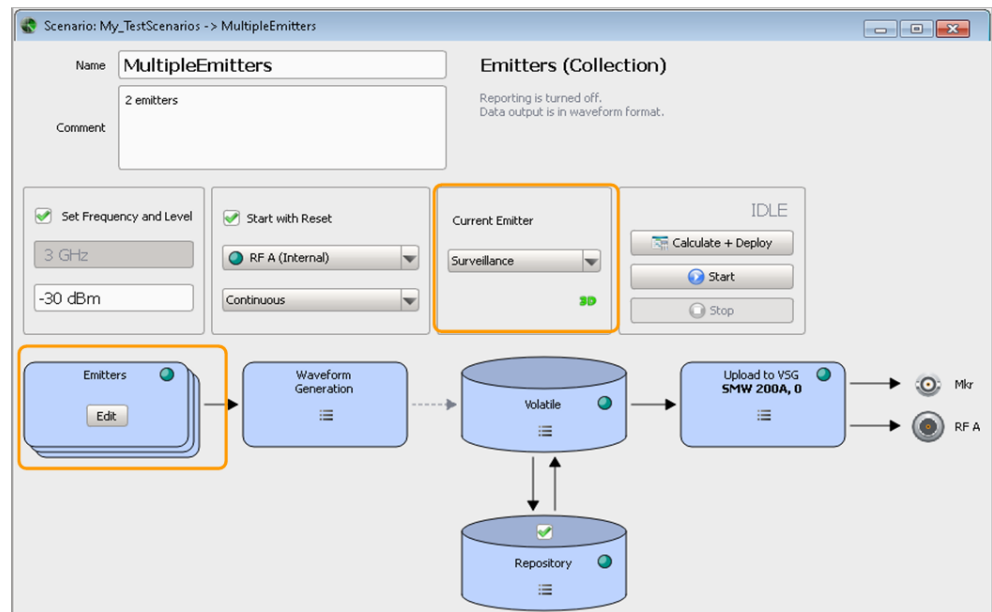
- 1 = Receiver's position, determined by the selected "Yaw" and "Pitch"; the position on the y-axis (the distance to the emitter) is fixed
 - 2a = "Yaw = 10 deg" turns the antenna horizontally, i.e. the receiver is not at the central scan axis but rotated in clockwise (CW) direction on the x-axis
 - 2b = "Pitch = -5 deg" turns the antenna vertically, i.e. "moves" the receiver on the z-axis
 - 3 = Current position of the scanning beam
 - 4 = Scan and pattern visualization; 6a uses "Scan Line = Pattern"
 - 5 = Amplitude changes in the received signal over time
 - 5a, 5b = Start time and simulation period (max value on the x-axis)
 - 6a, 6b = 6b is the XZ representation of the 6a 3D view; 6b uses "Scan Line = On"
4. Use the mouse to change the orientation of the interactive 3D diagram and to zoom on it.
For description of the provided settings, see [Chapter 11.2, "3D Emitter Preview Settings"](#), on page 202.

To configure complex scenarios with several emitters

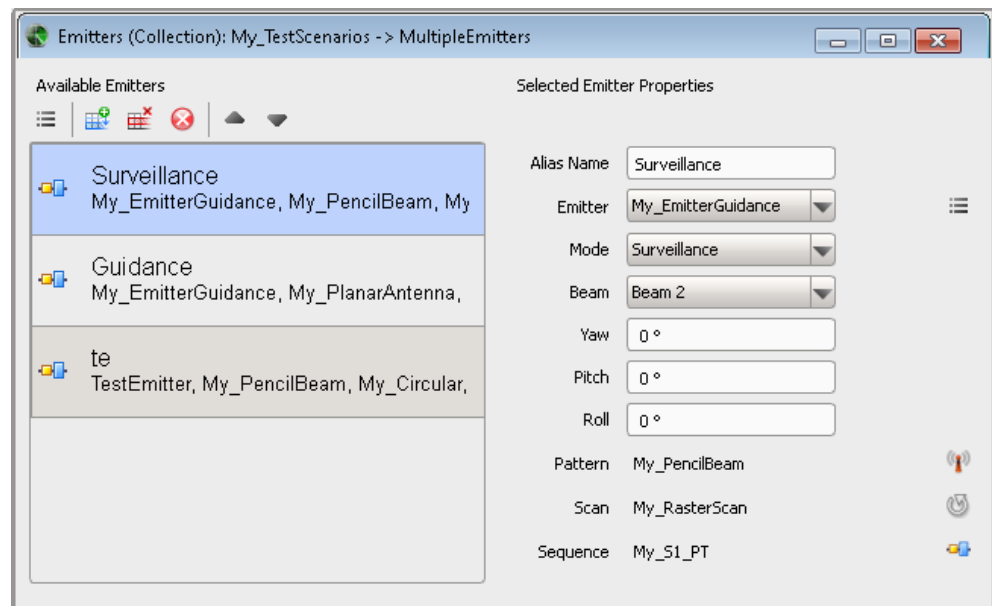
If your test situation requires to test the receiver's ability to detect the signal from different static emitters, you can use "Emitter (Collection)" scenario. In this dedicated scenario, you can configure several emitters and switch between them sequentially. You can also configure the receiver and change its position in the scanning beam of the emitters.

The following example creates a complex scenario with several emitters and a receiver to explain the principle.

1. Create an "Emitter (Collection)" scenario or open an existing one.



2. In the block diagram, select "Emitters > Edit".
The "Multiple Emitter" dialog opens.
3. Use the standard "New" function and insert the first emitter in the list.
You can select from the list of available emitters or create a new one.
4. Configure the emitter settings, like for example enter an alias name.



5. In the "Scenario" dialog, select the emitter that is transmitting, e.g. "Surveillance".
6. Use the 3D view to visualize the signal received by a static receiver.

See [Figure 11-3](#).

For description of the provided settings, see:

- [Chapter 11.3, "Lists with Multiple Emitters"](#), on page 203
- [Chapter 11.2, "3D Emitter Preview Settings"](#), on page 202

12 Emulating Receivers

The available receiver settings depend on the complexity of the simulated system:

- Receivers in the "Single Emitter" and "Emitters (Collection)" scenario types. These receivers have *an omnidirectional antenna with no antenna scan*. These scenarios only simulate the effect of emitter antennas and scans. Attenuation due to distance and free space propagation is not considered.
- Receivers in a 2D multi-emitters-receiver scenario have *one receive antenna*. They are described by an antenna beam pattern, an antenna scan, and an attitude information. The distance between the emitters and the receiver is configurable. This receiver complexity is used in the "Localized Emitters" scenario.
- Direction finding (DF) receivers can have *up to 10 antenna elements*. Each antenna element is described by the combination of an antenna pattern, antenna position and attitude. The receiver itself is also described with its position and attitude information. This receiver complexity is used in the "Direction Finding" scenario.

Table 12-1: Overview of the receiver types and provided settings

Scenario type	"Single Emitter" "Emitters (Collection)"	"Localized Emitters"	"Direction Finding"
Separate repository elements	-	-	Yes
Related settings	-	Chapter 12.1, "Receiver Properties" , on page 213	Chapter 12.2, "Receiver Settings" , on page 214
Scenario	Emitter-receiver	2D multi-emitter-receiver	2D multi-antenna receiver
2D map (gaming area)	-	Yes	
Emitter-receiver distance	Fixed	Configurable (Emitters placed on the 2D map)	
Receiver attitude	-	Configurable (Pitch, yaw, roll, and height)	
Number of antenna (elements)	1	1	≤ 10
Antenna pattern	Isotropic	Configurable	
Antenna scan	None (Steady)	Configurable	-
Antenna attitude	-	-	Configurable (Pitch, yaw, and roll per antenna element)

Received signal amplitude

One of the typical receiver characteristics is the Minimum Detectable Signal (MDS). The MDS is a measure of the receiver sensitivity and describes the minimum received signal amplitude S_{\min} that the receiver is able to detect. A typical receiver achieves an S_{\min} within the range of -80 dBm and -110 dBm.

In any type of emitter-receiver scenarios, the R&S Pulse Sequencer calculates the received signal and displays the time variation of the received normalized power.

The principle is illustrated on Figure 12-1.

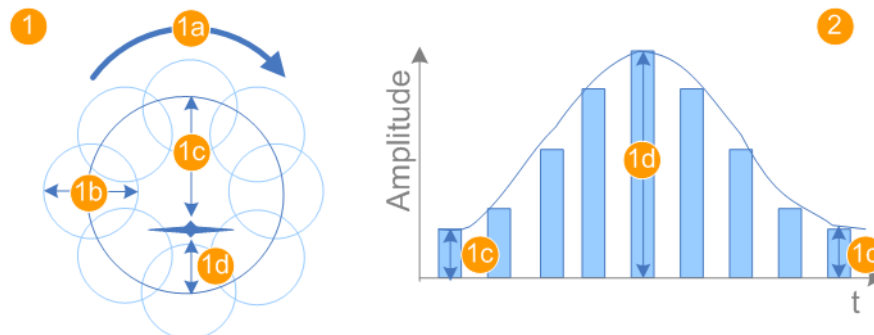


Figure 12-1: Simplified representation of a conical scan and the received signal

- 1 = Conical scan (antenna beams foresight)
- 1a = Direction of the rotating scan
- 1b = Antenna beam represented as its HPBW
- 1c, 1d = Target to beam distance, determines the amplitude of the received signal
- 2 = Variation of the amplitude of the received signal

The signal received by a static receiver has the shape of a sine wave, which amplitude is proportional to the distance between the target and the beam axis. Figure 12-2 shows how this simple case of an emitter-receiver scenario is visualized in the software.

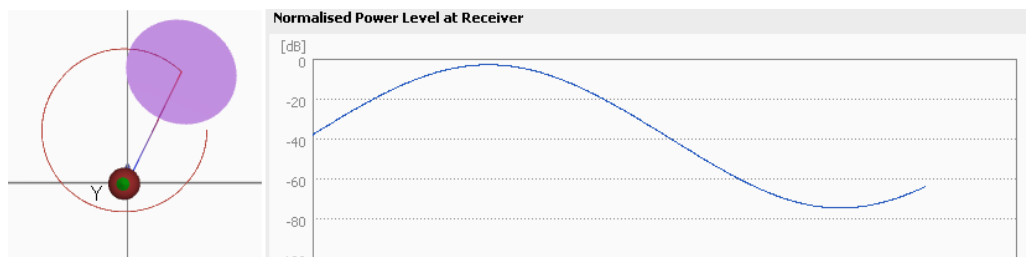


Figure 12-2: Normalized power level at receiver (simple emitter-receiver scenario)

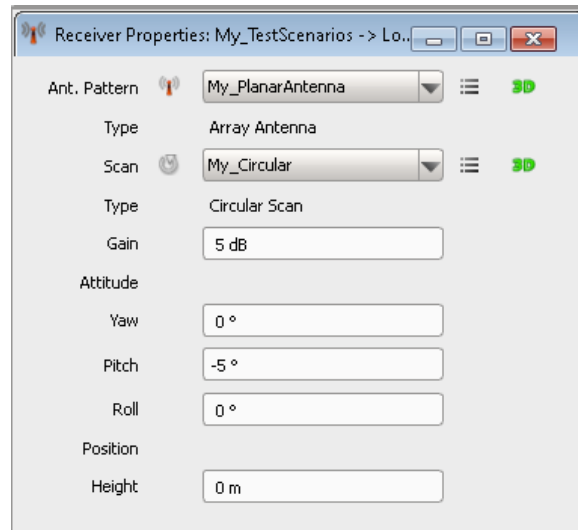
See:

- "To visualize the signal received by a static receiver" on page 207
- "To configure the receiver settings" on page 262

12.1 Receiver Properties

Access:

- ▶ In the 2D map of a "Localized Emitters" scenario, select "Receiver > Properties".



Settings:

Antenna Pattern and Scan	213
Gain	213
Attitude, Position	213

Antenna Pattern and Scan

Selects an antenna pattern and antenna scan type.

See also "[To configure the receiver settings](#)" on page 262.

Remote command:

[SCENario:LOCalized:RECeiver:ANTenna](#) on page 438

[SCENario:LOCalized:RECeiver:SCAN](#) on page 438

Gain

Sets the antenna [Gain](#).

Remote command:

[SCENario:LOCalized:RECeiver:GAIN](#) on page 439

Attitude, Position

Defines the orientation of the antenna as "Pitch", "Yaw", "Roll" and "Height".

See [Table 11-1](#).

Remote command:

[SCENario:LOCalized:RECeiver:DIRection:PItCh](#) on page 435

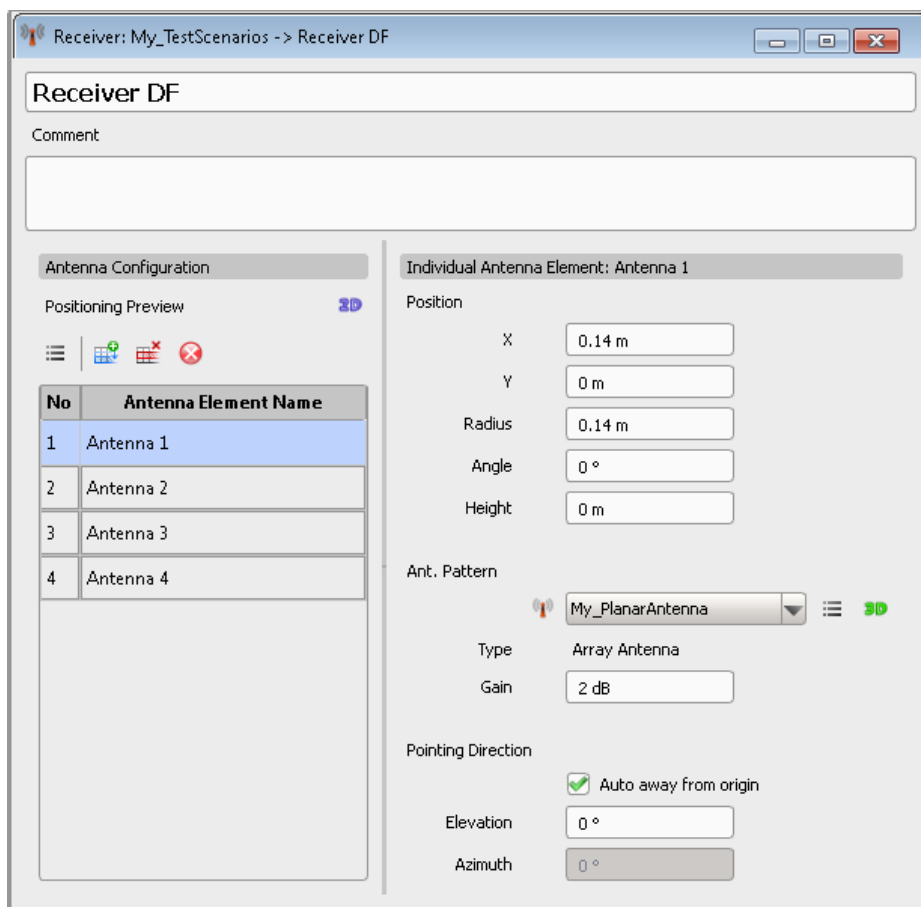
[SCENario:LOCalized:RECeiver:DIRection:YAW](#) on page 434

SCENario:LOCalized:RECeiver:DIRection:ROLL on page 435
 SCENario:LOCalized:RECeiver:HEIGHt on page 438

12.2 Receiver Settings

Access:

- ▶ Select "Repository Tree > Receiver > New".



Direction finding (DF) receivers can have up to 10 antenna elements, each described with an antenna pattern and position.

Settings:

Receiver Name.....	214
Comment.....	215
Antenna Configuration.....	215
Individual Antenna Element: Antenna x.....	216

Receiver Name

Enter the receiver's name.

Remote command:

[REceiver:NAME](#) on page 330

[REceiver:CREate](#) on page 329

[REceiver:SElect](#) on page 330

[REceiver:CATalog?](#) on page 329

[REceiver:REMove](#) on page 331

Comment

Enter a short description.

Remote command:

[REceiver:COMment](#) on page 331

Antenna Configuration

DF receiver can have up to 10 antenna elements.

"Positioning Preview 2D"

Displays the distribution of the antenna elements in a 2-dimensional chart (XY plane).

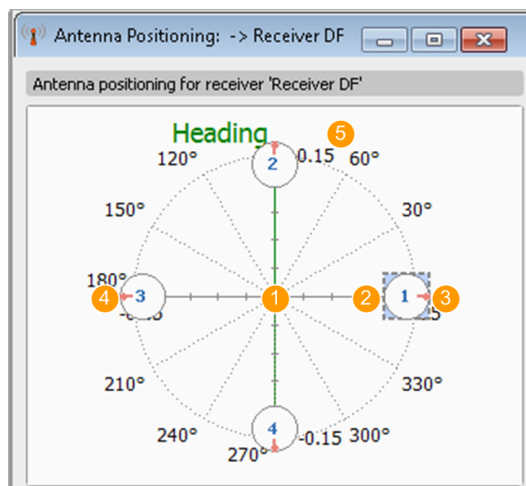


Figure 12-3: Antenna Positioning: understanding the displayed information

1 = Receiver origin

2 = Antenna element consequent number, as listed in the "List of antenna elements" table

3 = Antenna element orientation

4 = Selected antenna element

5 = Antenna element position in terms of radius and angle, relative to the receiver origin

"List of antenna elements"

Use the standard functions in the context menu and the icons to add, reorder, or remove antenna elements.

New elements are named automatically but you can change the alias name.

Remote command:

[REceiver:ANTenna:ADD](#) on page 332

[REceiver:ANTenna:SElect](#) on page 333

[REceiver:ANTenna:ALias](#) on page 416

[REceiver:ANTenna:DElete](#) on page 334

[REceiver:ANTenna:CLEar](#) on page 335

Individual Antenna Element: Antenna x

Describes the selected antenna element.

Open the "Positioning Preview 2D" view to display the configuration.

"Position" Antenna elements are spaced relative to the receiver origin.

Their 3D positions are defined as:

- "X" and "Y" values relative to the receiver origin.
- "Angle" (azimuth plane) and "Radius" (distance to the receiver origin on the XY plane), where "Angle = 0°" is on the X-axis.
- "Height" offset on the Z-axis relative to the receiver origin.

Example:

"X = 0 m" and "Y = -0.14 m" corresponds to "Radius = 0.14 m" and "Angle = 270 °".

Remote command:

[REceiver:ANTenna:POSition:X](#) on page 416

[REceiver:ANTenna:POSition:Y](#) on page 416

[REceiver:ANTenna:POSition:RADIus](#) on page 416

[REceiver:ANTenna:POSition:ANGLe](#) on page 416

[REceiver:ANTenna:POSition:HEIGHt](#) on page 417

"Antenna Pattern, Type"

Selects an existing antenna pattern or accesses a dialog for creating a new one.

The antenna pattern and type are displayed.

See also "[To create an antenna pattern](#)" on page 186.

"Gain" Sets the antenna Gain.

Remote command:

[REceiver:ANTenna:GAIN](#) on page 417

"Pointing Direction"

Turns the antenna beam axis in the elevation and the azimuth.

"Auto away form origin" sets the azimuth automatically, so that the beam axis is radial to the receiver origin.

Remote command:

[REceiver:ANTenna:DIRection:AWAY](#) on page 417

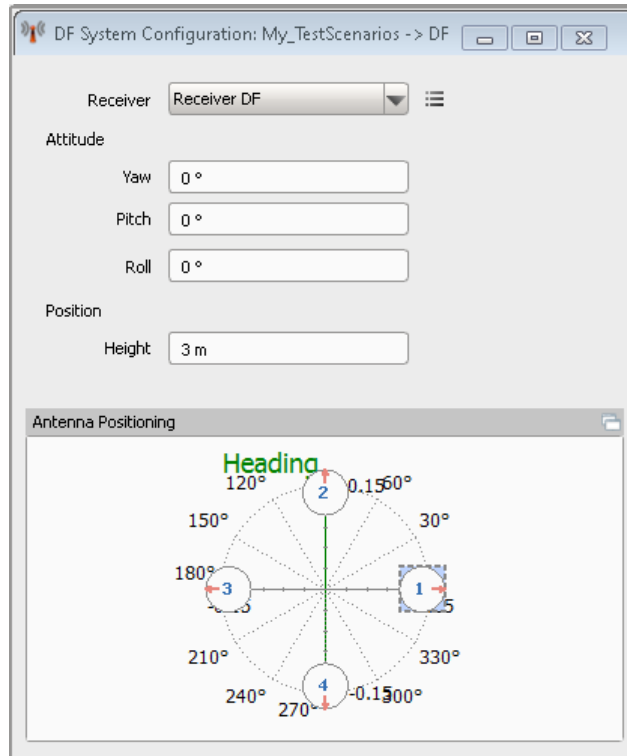
[REceiver:ANTenna:DIRection:AZIMuth](#) on page 417

[REceiver:ANTenna:DIRection:ELEVation](#) on page 418

12.3 DF System Configuration Settings

Access:

1. Open a "Direction Finding" scenario.
2. In the block diagram, select "Map > 2D".
3. On the 2D map, select "Receiver > Properties".



Settings:

Properties of the selected Receiver	217
Antenna Positioning	218

Properties of the selected Receiver

Each emitter is described with the following parameters:

"Receiver" Selects an existing receiver or accesses a dialog for creating a new one.

Remote command:

[SCENario:DIRection:RECeiver](#) on page 440

"Attitude > Pitch, Yaw, Roll"

Sets the receiver attitude in terms of "Pitch", "Yaw" and "Roll".
See [Table 11-1](#).

Remote command:

[SCENario:DIRection:RECeiver:DIRection:PItCh](#) on page 434

[SCENario:DIRection:RECeiver:DIRection:YAW](#) on page 434

[SCENario:DIRection:RECeiver:DIRection:ROLL](#) on page 435

"Position > Height"

Sets the receiver position.


Remote command:

[SCENario:DIRection:RECeiver:HEIGHt](#) on page 438

Antenna Positioning

Displays the distribution of the antenna elements on a 2D map (XY plane).

The graph resembles the one in the "Receiver" dialog, see [Figure 12-3](#).

Use the split window icon  to detaches the diagram and open it in a separate window.

13 Working with Waveforms and Generating Interfering Signals

In addition to creating pulse sequences, you can also create sequences composed of waveforms.

Waveforms can be created by the R&S Pulse Sequencer or with an external software. If the data format is in one of the supported import formats, you can import any waveform into the waveform library. Regardless whether they are created or imported, waveforms can be used as interferers in complex 2D scenarios.

Supported waveform types

- Internally created:
 - Multitone and **CW** waveforms
 - White Gaussian noise signals with configurable bandwidth
 - The signal of a group of background emitters
- Externally created:
 - Waveforms in the Rohde & Schwarz proprietary file format (*.wav). Such files are generated with the signal generation software R&S WinIQSIM2 or with the real-time options of the Rohde & Schwarz signal generators.
 - Files with custom I/Q data created with other Rohde & Schwarz instruments or third party software. For example MATLAB (*.mat), R&S®PR100 (*.riq), Rohde & Schwarz signal and spectrum analyzers (*.iq.tar), or the Rohde & Schwarz proprietary modified audio file format (*.wav).
 - Files with custom I/Q data in ASCII format (*.txt) or in binary format (*.bin). ASCII files may describe the I and Q vectors in one file or in separated *_i.txt and *_q.txt files. If file pairs are stored in the same directory, the second file is recognized automatically by the [Import Wizard](#).

See also [Table A-1](#), on page 480.

Waveforms as interfering signals

Test scenarios often require interfering signals, for example to simulate more realistic conditions or to stress the DUT (receiver).

In the R&S Pulse Sequencer, you can use waveforms as interfering signals. The following is an overview of the different kind of waveforms, their properties, and application fields. You can combine different kind of waveforms in the same scenario:

- Unmodulated (**CW**) interfering signals
Waveform types with suitable characteristics are the *multitone and CW waveforms*. See "[About the multitone signals](#)" on page 220.
- Pulse interfering signals
Waveform types with suitable characteristics are the *background emitters and the imported waveforms containing pulsed signal*.

A background emitter creates the signal of up to 255 emitters. When included in a "Localized Emitters" scenario, background emitters act as a source of a noise-like pulsed background signal.

See ["About the background emitters"](#) on page 221.

- Modulated interfering signal

Waveforms *containing radio communication signals* can be imported in the software.

These types of waveforms are useful if the test case requires interfering radio communication signals or any kind of modulated signals broadcasted at a particular frequency.

See ["Settings for importing externally created R&S waveforms or files with custom I/Q data"](#) on page 227.

- Noise

The waveform type *white Gaussian noise* creates this kind of signal.

See ["White Noise settings"](#) on page 227.

In the "Localized Emitters" scenarios, the interferers are treated like emitters, i.e. they are described with a waveform, an antenna pattern and antenna scan, static location with its X, Y and Z coordinates, and orientation defined with an "Azimuth" and an "Elevation". Per default, interferers broadcast power equally in all directions.

About the multitone signals

Multitone waveforms may consist of up to 1000 sine waves (tones).

The spacing and the number of tones are user-definable. The tones are arranged symmetrically around the RF carrier (see [Figure 13-1](#)).

The start phase of each tone (CW) signal is a random value. This approach typically results in signals with lower PAPR compared to multitone signals where all CW start with the same phase.

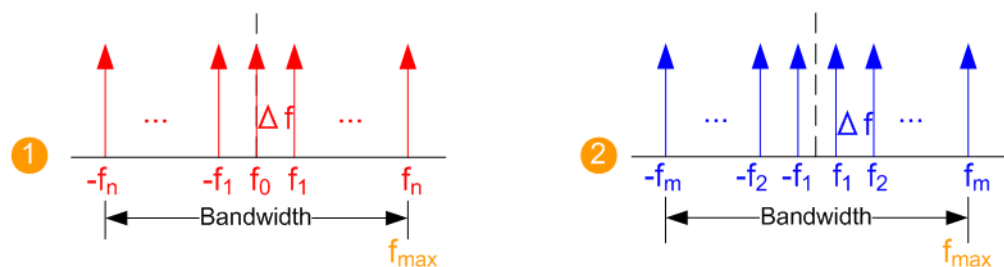


Figure 13-1: Multitone waveforms: arrangement of the tones and calculation of the total bandwidth

- 1 = Odd number of tones
- 2 = Even number of tones
- Δf = "Spacing"
- Bandwidth = ("Tones" - 1) * Δf
- f_{\max} = Bandwidth/2 = ("Tones" - 1) * $\Delta f/2$

The oversampling is set automatically and the clock rate is calculated as follows:

$$\text{Clock Rate} = 1.25 * \text{Bandwidth} = 1.25 * 2 * f_{\max}$$

Example:

The following applies for a multitone waveform with "Tones = 64" and "Spacing = 100 kHz":

- Bandwidth = (64 - 1)*100 KHz = 6.3 MHz
- Clock Rate = 7.875 MHz

For description of the related settings, see "CW and Multitone waveform settings" on page 224.

About the background emitters

A background emitter creates the signal of up to 255 emitters.

The created waveform is the superimposed (overlay) signal of up to 255 pulse sequences, one sequence per emitter. Each pulse sequence is composed of a pulse repeated with random PRIs. The pulse widths are randomly selected values that are constant within a sequence but vary on a per emitter basis.

Each emitter uses a randomly selected frequency within the selected bandwidth.

See Figure 13-2 for illustration of the waveform creation principle.

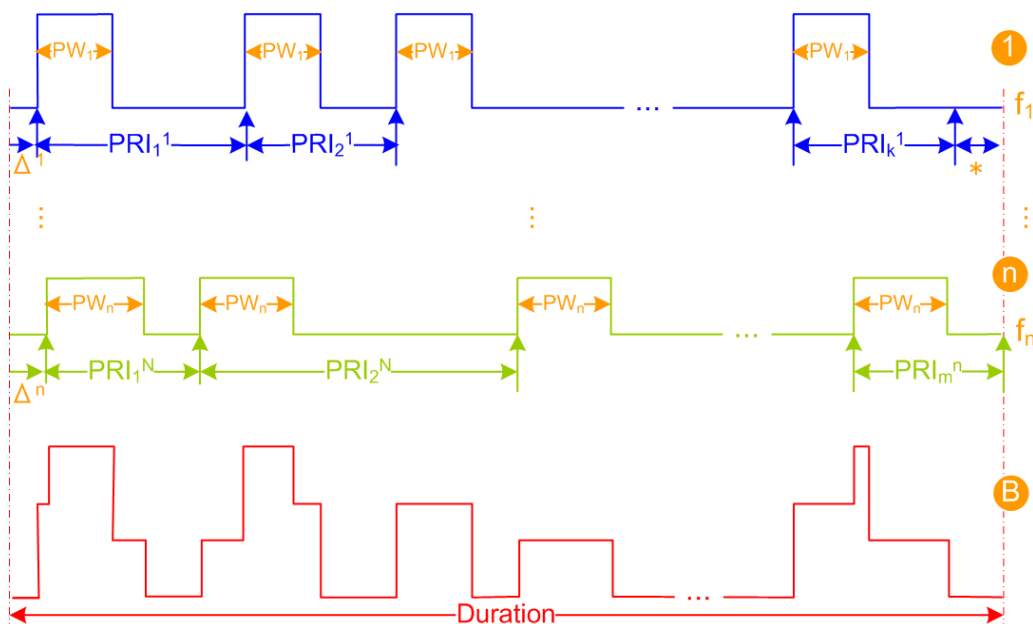


Figure 13-2: Background emitters: simplified representation of waveform structure

- n = Number of emitters ("Count")
- 1, n = Pulse sequences of the 1st and the nth emitters
- f₁ to f_n = n random frequency values, within the selected "Bandwidth"
- Δ¹, Δⁿ = n random delays
- PW₁, PW_n = n random pulse width values, constant per sequence and different for each emitter
- PRI₁¹ .. PRI_kⁿ = Random PRIs, within the "PRI/PW Ratio" value range
- k, m, PRI = Number of pulses per sequence and PRIs, selected so that the level of the superimposed signal portions does not exceed 12 dB above the Ref. Level (see Figure 13-3)
- * = Additional pulse off time at the sequence end to avoid pulse truncation
- B = Resulting signal (Background emitter)

The random process uses the uniform distribution function. The selected values fulfill the user-defined value ranges and the following rules:

- Each pulse uses a "Rise/Fall Time = 1 Sample" and "Pulse Width ≥ 3 Samples"
- Each sequence starts with a randomly selected delay Δ , calculated as follows:
 $\Delta^n = \text{rand}() * (\text{PRI}_m^n - \text{PW}_n)$, where $\text{rand}() = 0 \dots 1$
- Pulses are not truncated.
 If the pulse width of the last pulse is longer than the remaining sequence time, the pulse is skipped and the pulse off time is extended until the sequence end.
- The unattenuated pulse top power level of the first emitter is set to the reference level.

All other emitters use reduced pulse top power level, where the maximum level difference between the strongest and the weakest emitter is set with the parameter "Maximum Level Range".

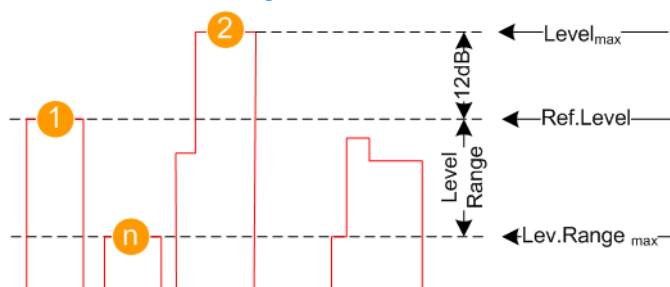


Figure 13-3: Background emitters signal leveling

- 1 = Pulse top power of the first emitter (i.e. the strongest emitter), set to the reference level (Ref. Level)
 - n = Pulse top power of the n^{th} emitter (i.e. the weakest emitter), set to the "Maximum Level Range" (Lev. Range_{max}) value
 - 2 = Superimposed pulses; the maximum level of the superimposed portion can be 12 dB above the reference level
- $\text{Level}_{\text{max}} = \text{Ref. Level} + 12 \text{ dB}$

Pulses are superimposed as long as the level of the superimposed signal portions does not exceed 12 dB above the reference level. If this limit is exceeded, a new PRI is selected.

- Pulse top power levels (in dB) are uniformly distributed values

The oversampling is performed automatically, where the used clock rate is calculated as follows:

Clock Rate = 1.25*"Bandwidth" and Clock Rate ≥ 1 MHz.

The minimum clock rate assures correct sampling of narrow bandwidth signals composed of many fast repeating pulses (small PRIs).

See:

- "Background Emitters settings" on page 225 for description of the related settings
- "To add a background signal" on page 260.

Waveform scenarios and sequences

Waveforms can be:

- Included in 2D scenarios ("Scenario Type > Localized Emitters") as interferer or as part of a sequence
- In all other scenario types, as part of a sequence

When include in sequences, waveforms are used in similar way as pulses, see [Chapter 8.2.1, "Sequence Description Settings"](#), on page 114.

Dedicated waveform sequences are also described in table form where the rows are processed sequentially. Each row is repeated once or as defined with the parameter "Repetition Count". Complex waveform sequences may contain waveforms and loops. Other control elements are not supported.

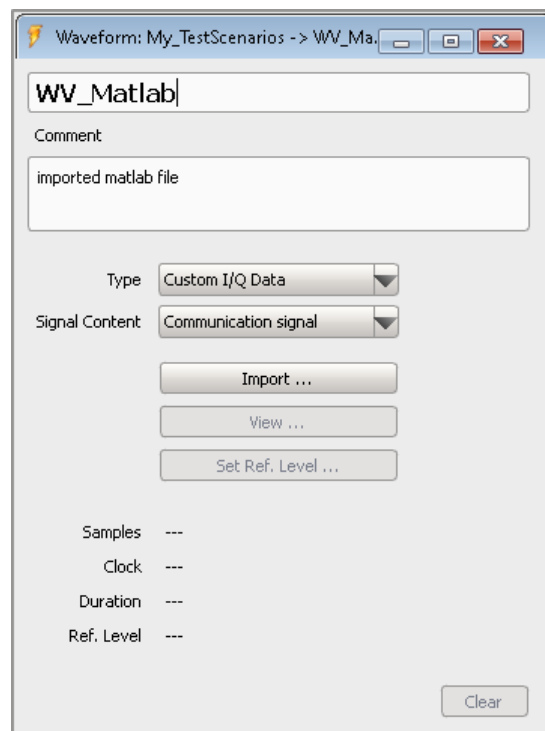
For step-by-step instructions, see:

- ["To create a waveform scenario"](#) on page 235
- ["To create a sequence of waveforms"](#) on page 235.

13.1 Waveform Settings

Access:

- ▶ Select "Repository Tree > Waveform > New".



Available settings:

- Waveform Name and Comment..... 224
- Type..... 224
- CW and Multitone waveform settings..... 224
- Background Emitters settings..... 225
- White Noise settings..... 227
- Settings for importing externally created R&S waveforms or files with custom I/Q data
..... 227
- Import Wizard..... 229
 - L Input..... 229
 - L ASCII and Binary Format..... 229
 - L MATLAB File Content..... 230
 - L RIQ File Content..... 230
 - L *.iq.tar and *.wav File Header..... 231
 - L Scaling, IQ vs Time, Constellation, Spectrum.....231

Waveform Name and Comment

Unambiguous name and an optional description.

Remote command:

- [WAVEform:NAME](#) on page 331
- [WAVEform:CREate](#) on page 329
- [WAVEform:SELEct](#) on page 330
- [WAVEform:CATalog?](#) on page 329
- [WAVEform:REMOve](#) on page 332
- [WAVEform:COMMeNt](#) on page 331

Type

Sets the type of the waveform to be created or imported.

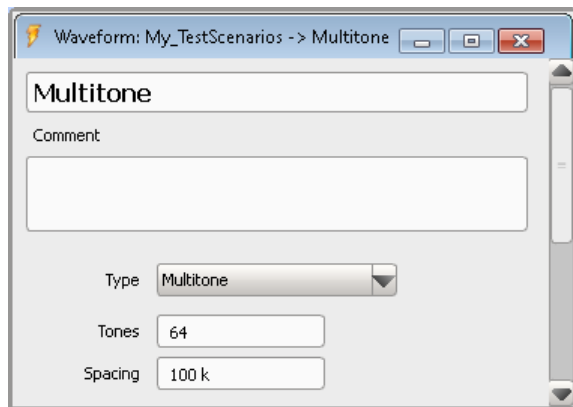
See also "[Supported waveform types](#)" on page 219.

Remote command:

- [WAVEform:TYPE](#) on page 458

CW and Multitone waveform settings

Unmodulated (CW) waveforms do not require further settings.



Multitone waveforms may consist of several unmodulated (CW) tones, where the tone spacing and the number of tones are user-definable.

See ["About the multitone signals"](#) on page 220 for details.

Provided are the following settings:

"Tones" Sets the number of tones composing the multitone signal.

"Spacing" Sets the spacing (Δf offset) between the tones.

Remote command:

[WAVeform:MT:COUNT](#) on page 459

[WAVeform:MT:SPACIng](#) on page 459

Background Emitters settings

A background emitter is a function that automatically creates the signal of up to 255 emitters.

Waveform: My_TestScenarios ->

BKG

Comment

Type: Background Emitters

Count: 64

Bandwidth: 10 MHz

Duration: 50 ms

Pulse Width Range

Minimum: 50 us

Maximum: 250 us

PRI / PW Ratio

Minimum: 10

Maximum: 20

Level Range

Maximum: 20

See ["About the background emitters"](#) on page 221 for details.

Provided are the following settings:

"Count" Sets the number of background emitters, that is the number of random selected frequencies within the defined "Bandwidth". Higher number of emitters leads to higher pulse density.

Remote command:

[WAVeform:BEMitter:COUNT](#) on page 459

"Bandwidth" Sets the maximum permissible bandwidth of the resulting signal. Emitters are randomly placed within the selected bandwidth.

Remote command:

[WAVeform:BEMitter:BWIDth](#) on page 460

"Duration" Sets the signal duration (play time) of the waveform.

Remote command:

[WAVeform:BEMitter:DURation](#) on page 460

"Pulse Width Range"

The emitter pulse widths are randomly selected values within the range PW_{min} to PW_{max} .

$$PW_{min} \leq PW^N \leq PW_{max}$$

Remote command:

[WAVeform:BEMitter:PW:MINimum](#) on page 460

[WAVeform:BEMitter:PW:MAXimum](#) on page 460

"PRI/PW Ratio"

The PRI values are selected so that the PRI/PW ratio is within the range $(PRI/PW)_{min}$ to $(PRI/PW)_{max}$.

$$PRI_m^N = k * PW^N, \text{ where:}$$

- $k = \text{rand}() * [(PRI/PW)_{max} - (PRI/PW)_{min}] + (PRI/PW)_{min}$ and
- $\text{rand}() = 0 \dots 1$

Remote command:

[WAVeform:BEMitter:PRI:RATio:MINimum](#) on page 460

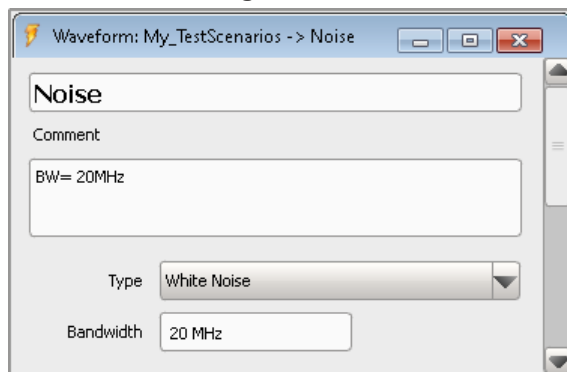
[WAVeform:BEMitter:PRI:RATio:MAXimum](#) on page 460

"Maximum Level Range"

Sets the maximum level difference between the strongest and the weakest emitter.

Remote command:

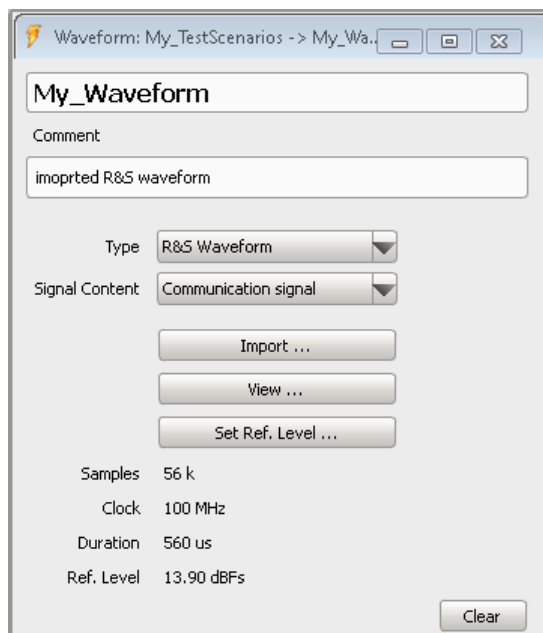
[WAVEform:BEmitter:LEVel:RANGe](#) on page 461

White Noise settings

"Bandwidth" Sets the bandwidth of the generated "White Noise" waveform.

Remote command:

[WAVEform:NOISe:BWIDth](#) on page 458

Settings for importing externally created R&S waveforms or files with custom I/Q data

This section describes the available settings.

For step-by-step instructions, see:

- ["To import a file with custom I/Q data"](#) on page 233
- ["To retrieve more information on an imported waveform"](#) on page 234

- ["To change the waveform reference level based on a user-definable signal section"](#) on page 270

"Signal Content"

Sets the waveform signal type and thus determines the filter type used during the resampling process.

The software applies different filters on pulsed signals and on the waveforms containing digital standard signals.

- Waveforms containing digital standard signals are intended for continuous playback. The end of the waveform fits the beginning to avoid spectral regrowth caused by the wrap-around.
- For pulsed signals, however, wrap-around and overshoot effects due to steep filters are undesired. Thus, low-order filters are applied to prevent from overshoots.

Remote command:

[WAVEform:SIGCont](#) on page 458

"Import" Accesses a file import wizard, see ["Import Wizard"](#) on page 229.

Remote command:

For loading waveform files

[WAVEform:WAVEform:LOAD](#) on page 458

"View, Set Ref. Level"

Access dialogs which visualize the waveform and displays more useful information on it, see:

- [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264
- [Chapter 16.2, "Waveform Reference Level Settings"](#), on page 267

Remote command:

[WAVEform:LEVel:REFErence](#) on page 458

"Samples, Clock, Duration, Ref. Level"

After data import, the "Waveform" dialog displays information on the waveform.

- "Samples": Indicates the number of waveform samples. A warning symbol indicates that the waveform contains fewer than 512 samples. The waveform length might violate the minimum required ARB file size.
- "Clock": Indicates the clock rate.
- "Duration": Indicates the waveform duration
- "Ref. Level": Shows the calculated reference level

Remote command:

[WAVEform:LEVel:REFErence](#) on page 458

"Clear" Removes the imported waveform or file with I/Q data.

Remote command:

[WAVEform:WAVEform:CLEAR](#) on page 459

[WAVEform:IQ:CLEAR](#) on page 459

Import Wizard

The data import wizard converts custom I/Q data into a Rohde & Schwarz waveform file.

The dialog and the provided settings differ depending on the selected file format but the differences are self-explanatory.

Input ← Import Wizard

The input settings depend on the custom I/Q data.

The example on [Figure 13-4](#) shows the dialog when importing custom I/Q data in ASCII format.

ASCII files may describe the I and Q vectors in one file or in separated *_i.txt and *_q.txt files. If file pairs are stored in the same directory, the second file is recognized automatically by the "Import Wizard".

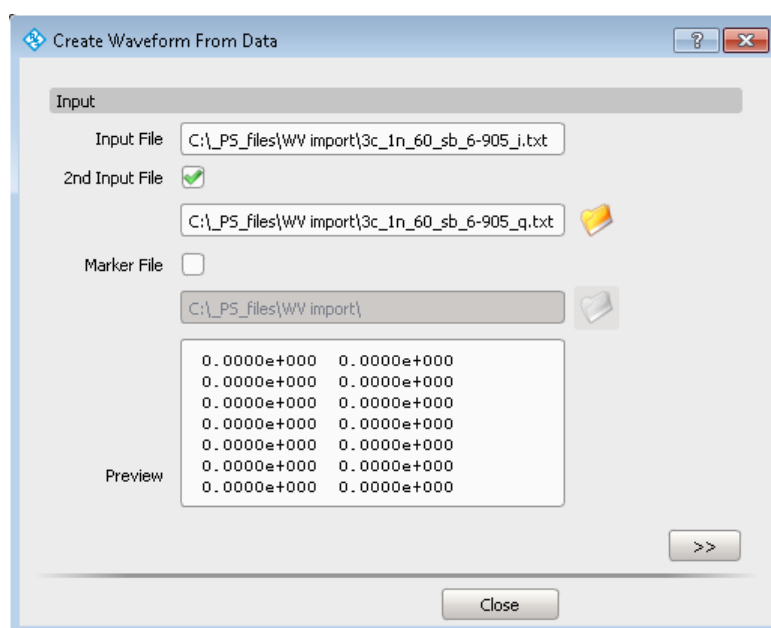


Figure 13-4: Import Wizard > Input (ASCII file)

ASCII and Binary Format ← Import Wizard

The following settings are available for importing of custom I/Q data in ASCII format (*.txt) and binary format (*.bin).

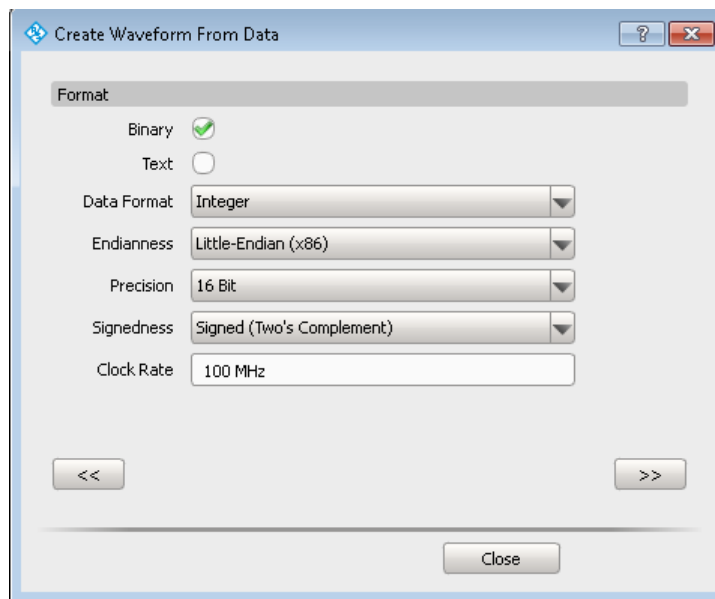


Figure 13-5: Import Wizard > Format (ASCII and binary files)

MATLAB File Content ← Import Wizard

Indicates the file with custom I/Q data and the data type.

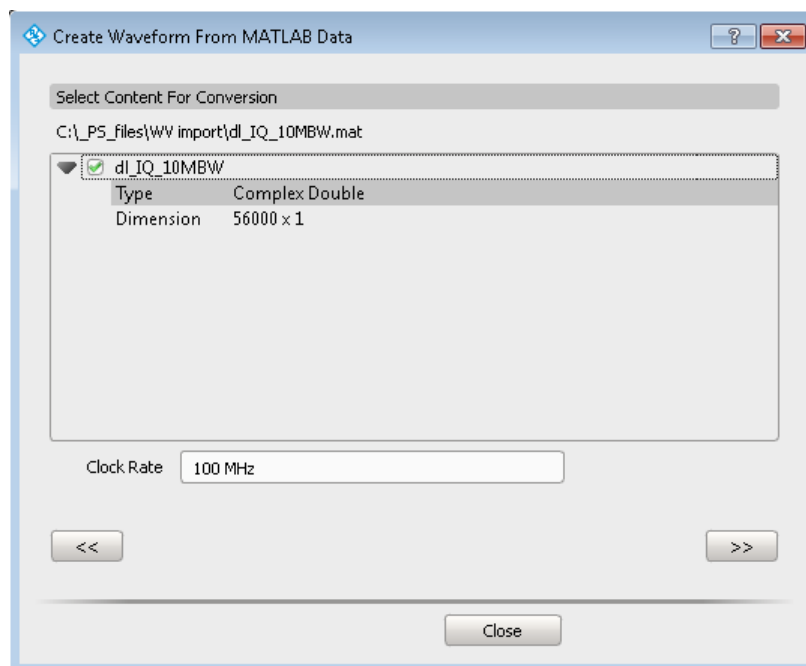


Figure 13-6: Import Wizard > Content (MATLAB file)

RIQ File Content ← Import Wizard

Displays the file content information.

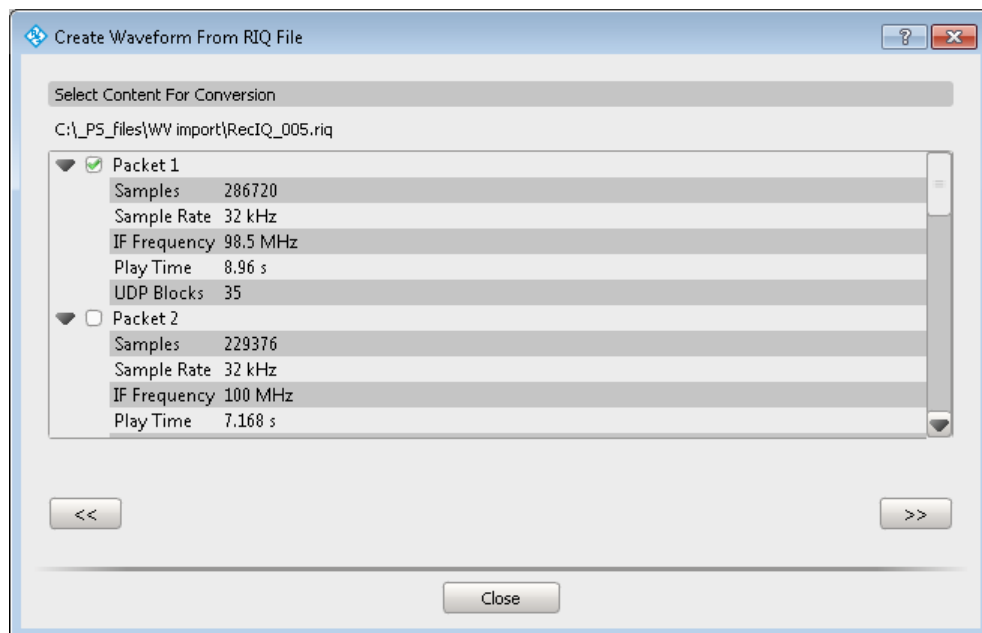


Figure 13-7: Import Wizard > Content (RIQ file)

***.iq.tar and *.wav File Header ← Import Wizard**

Displays the file header information retrieved from the file content.

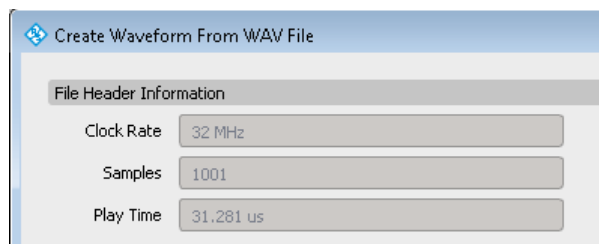


Figure 13-8: Import Wizard > Content (*.iq.tar and *.wav file)

Scaling, IQ vs Time, Constellation, Spectrum ← Import Wizard

The example on [Figure 13-9](#) shows the dialog when importing MATLAB files.

For overview of the supported file types, see "[Supported waveform types](#)" on page 219.

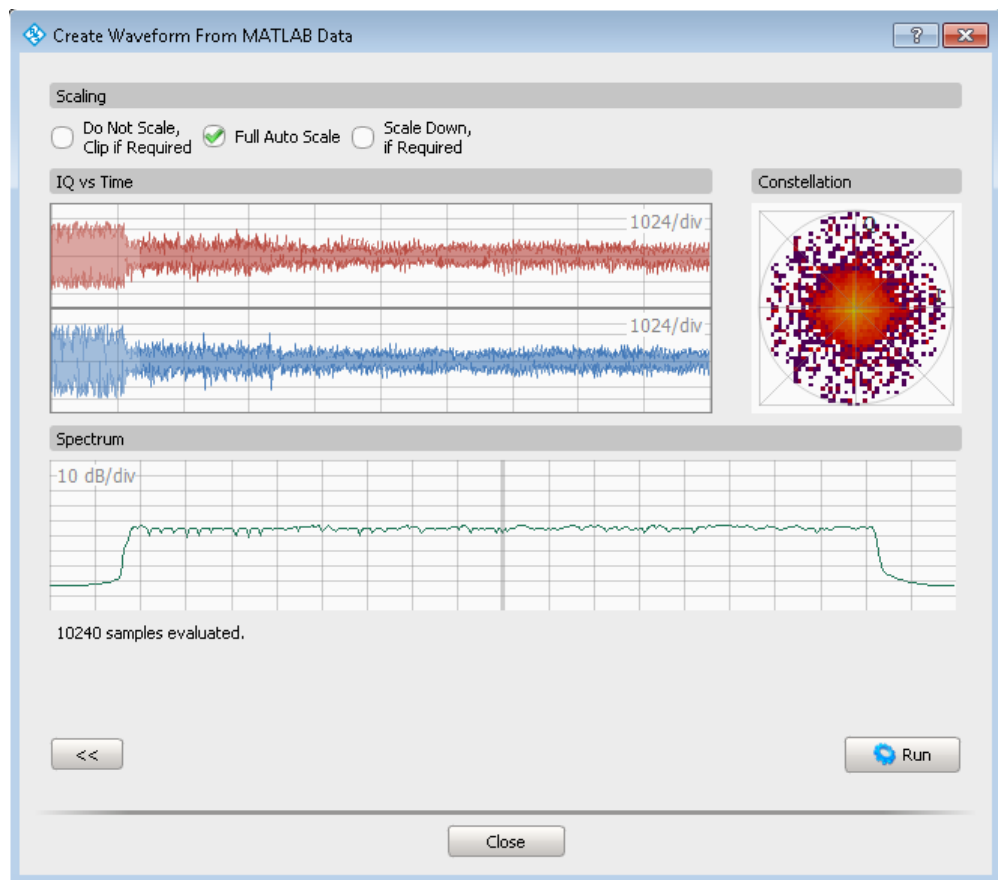


Figure 13-9: Import Wizard > Scaling (MATLAB file)

The dialog shows a preview of the first 10 000 samples in the file.

Per default, the "Full Auto Scale > On" is selected. The waveform is calculated so that it perfectly fits the 16-bit dynamic range of the instrument's ARB. The waveform is not clipped.

The selection "Do Not Scale, Clip if Required > On" leaves the signal amplitude unchanged. This setting could lead to low signal levels or clipping.

13.2 How to Create a Waveform Scenario and Work with Waveforms

See:

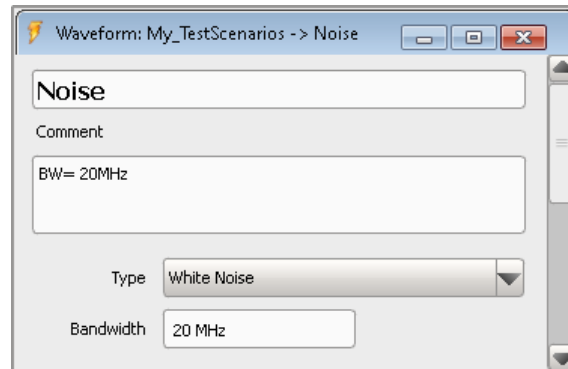
- ["To create a waveform scenario"](#) on page 235
- ["To create a sequence of waveforms"](#) on page 235
- For how to build a waveform library, see:
 - ["To create a waveform"](#) on page 233
 - ["To import a file with custom I/Q data"](#) on page 233
 - ["To retrieve more information on an imported waveform"](#) on page 234

To create a waveform

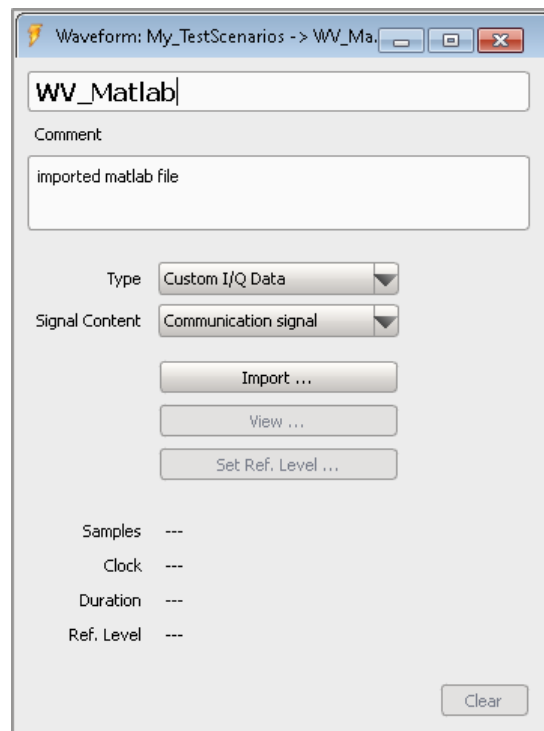
1. Select "Repository Tree > Waveform > New".

A new waveform with default settings is created and added to the project tree. New waveforms are named `Wav_<n>` where `n` is a number starting at one.

2. Add information to describe and identify the waveform, like a name and a comment.
3. Select "Type > White Noise" and set the "Bandwidth".

**To import a file with custom I/Q data**

1. Select "Repository Tree > Waveform > New".
2. Enter a name and a comment.
3. Select "Type > Custom I/Q Data".



4. Select "Import", navigate, and select the *.mat file and follow the instructions.

An "Import Wizard" opens. See the example on [Figure 13-9](#).

The wizard differs depending on the selected file format but the differences are self-explanatory.

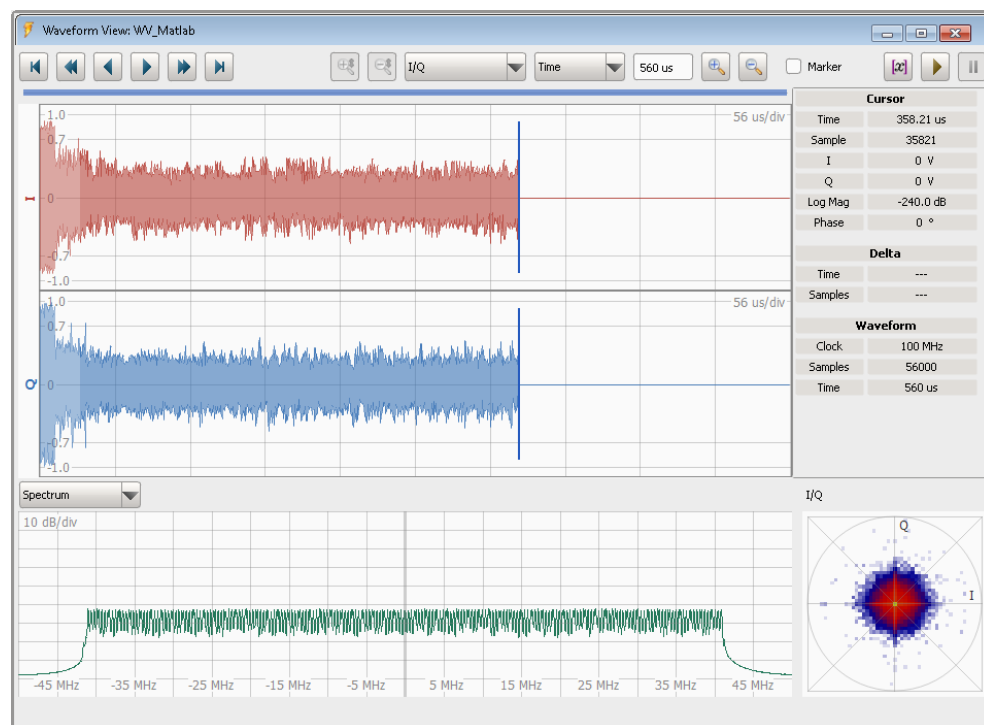
After the import has been completed, the "Waveform" dialog displays information on the waveform.

Samples	56 k
Clock	100 MHz
Duration	560 us
Ref. Level	14.47 dBfs

See also ["To retrieve more information on an imported waveform"](#) on page 234.

To retrieve more information on an imported waveform

1. Select "Repository Tree > Waveform", e.g. "WV_Matlab"
2. To display the I/Q diagram or the spectrum of the signal, in the "Waveform" dialog, select "View".



For description of the provided settings, see:

- [Chapter 16, "Visualizing and Analyzing Waveforms"](#), on page 264
- [Chapter 16.1, "Waveform and Data View Settings"](#), on page 264

To change the waveform reference level

- ▶ Follow the instructions in ["To change the waveform reference level based on a user-definable signal section"](#) on page 270.

For description of the provided settings, see:

- [Chapter 16.2, "Waveform Reference Level Settings"](#), on page 267

To emulate an interferer by using waveforms

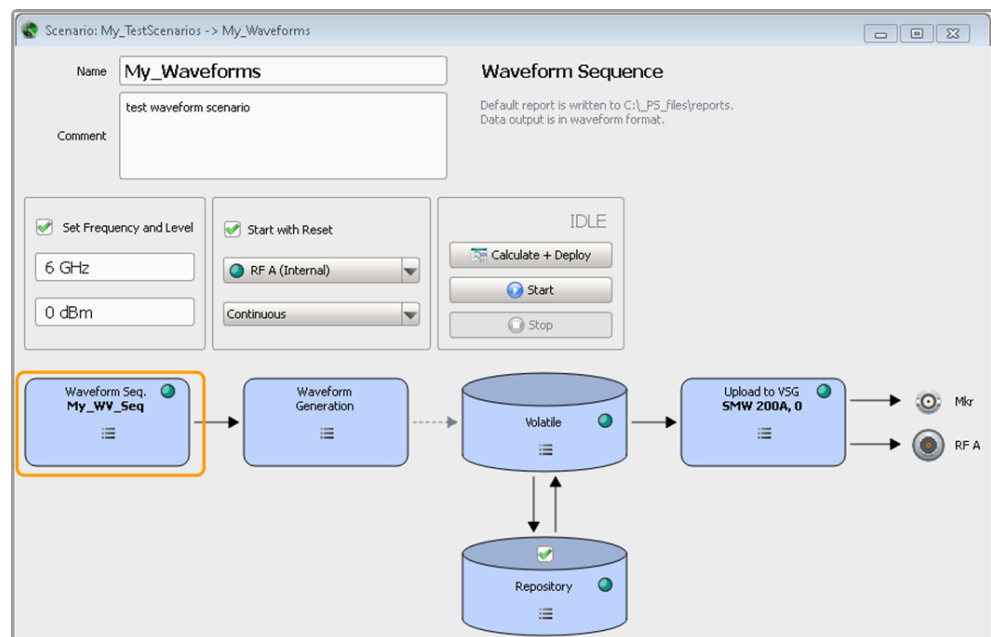
Waveforms can be used in sequences or, in "Localized Emitters" and "Direction Finding" scenarios, used to emulate interferes.

- ▶ See ["To add an existing waveform as an interferer to the scenario"](#) on page 260 for how to emulate interferers.

To create a waveform scenario

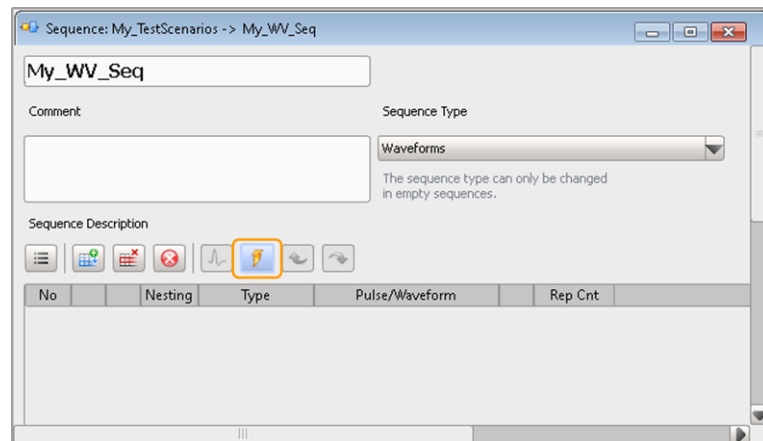
The "Waveform Sequence" scenario is a dedicated scenario type for combining waveforms.

- ▶ Create a scenario with "Scenario Type > Waveform Sequence".
See also ["To create a single pulse train scenario"](#) on page 69.



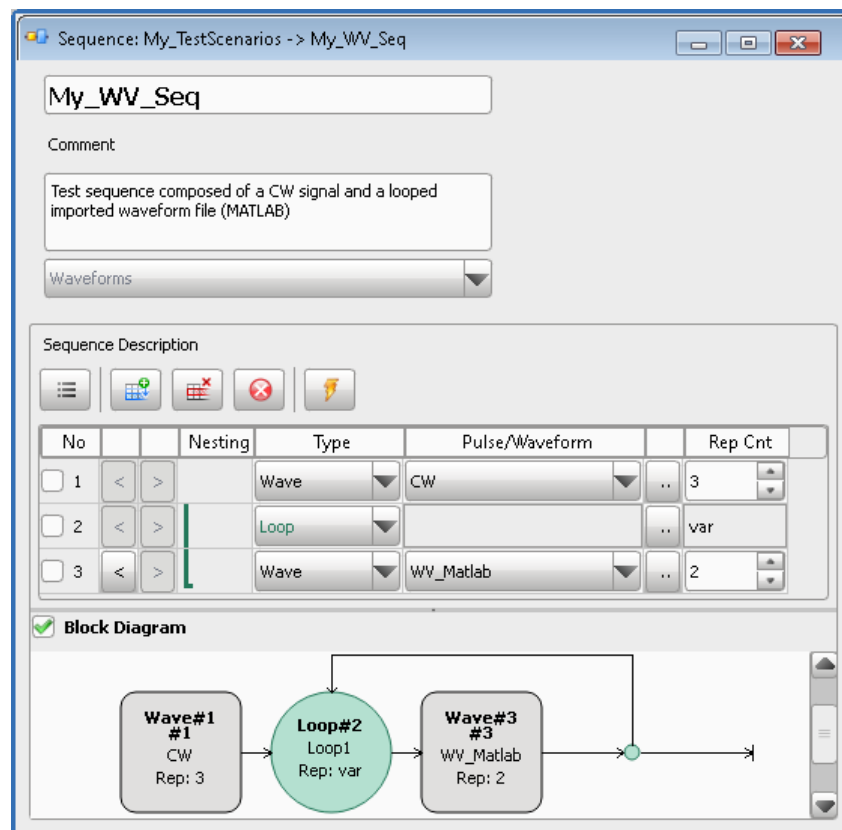
To create a sequence of waveforms

1. Perform one of the following:
 - a) In the "Waveform Scenario" dialog, select "Waveform Sequence > New".
 - b) Select "Repository Tree > Sequence > New" and select "Sequence Type > Waveforms".



2. To add waveforms to the sequence, perform one of the following:
 - select the "Add Waveform" icon create a waveform, see ["To create a waveform"](#) on page 233.
 - If there are waveforms in your repository, select the "Append New Item" icon and select "Type > Wave".

A waveform sequence contains one or more waveforms. Complex sequences may also contain loops.



See also ["To define and apply a loop"](#) on page 126.

14 Creating and Mapping Generator Profiles

Files generated by the R&S Pulse Sequencer can be transferred to and played by remotely connected signal generators.

The connection between the software and the signal generator is configured with the following steps:

- Create a *generator profile*
The generator profile is a logical placeholder for a physical instrument. It describes the minimum requirements that must be fulfilled so that the signal can be generated.
When a new empty repository is created, one default generator profile is created, too.
- Configure a *physical instrument*
Generator profiles are assigned to physical instruments.
- Assign a generator profile to the scenario.

14.1 Profile to Instrument Links Settings

Access:

- ▶ Perform one of the following:
 - select "Menu bar > Configure > Profile to Instrument Links"
 - in the block diagram of a scenario, select "Upload to VSG/Local ARB File > context menu > Generator Link".

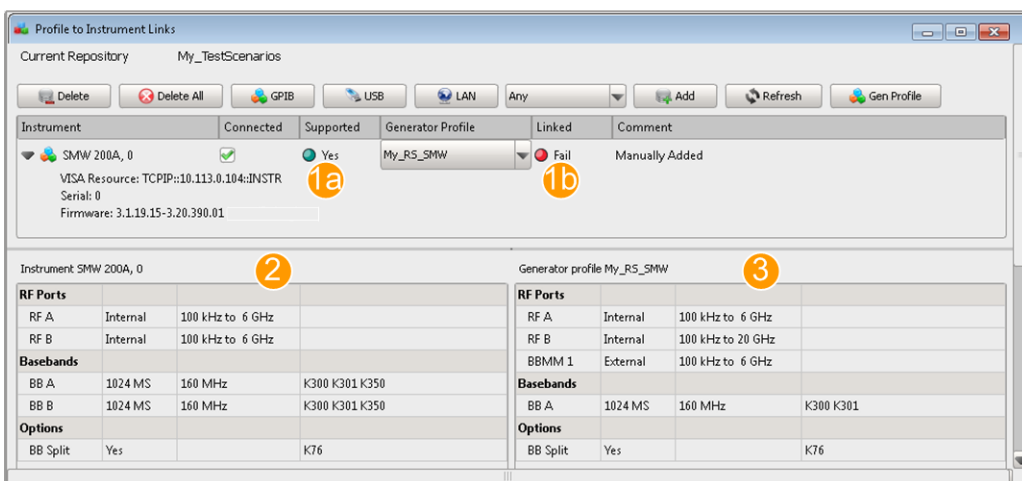


Figure 14-1: Profile to Instrument Links: understanding the displayed information

- 1a, 1b = Status messages; "Linked > Fail" indicates insufficient instrument's configuration for the current profile (e.g. compare the available and the required max. RF)
- 2 = Lists the retrieved instrument configuration
- 3 = Lists the profile capabilities, see [Chapter 14.3, "How to Create and Map Generator Profiles"](#), on page 244

Available settings:

Current Repository	238
Delete, Delete All	238
Scan GPIB/USB/LAN, Any	238
Add	238
Refresh	238
Gen Profile	238
List of connected instruments	239
Device Capabilities, Profile Capabilities	240

Current Repository

Indicates the current repository.

Delete, Delete All

Standard edit functions to handle the list of instruments.

Remote command:

[INSTRument:DELeTe](#) on page 334

[INSTRument:CLEar](#) on page 335

Scan GPIB/USB/LAN, Any

Scans the selected local interface for connected instruments. To limit the scan time, select the instrument's type, e.g. "SMW200A".

The GPIB scan is limited to the first GPIB card that is installed (GPIB0).

Add

Adds an instrument manually.

If an instrument with the selected IP address/computer name or complete VISA resource string exists, it is added to the list of instruments.

Remote command:

[INSTRument:ADD](#) on page 371

Refresh

Checks whether the listed instruments are connected or not and updates their connection status indication.

Gen Profile

Creates a matching generator profile for the selected connected generator.





See [Chapter 14.2, "Generator Profile Settings"](#), on page 240.

Remote command:

[INSTrument:CREate](#) on page 371

List of connected instruments

Displays information on all known and previously configured instruments, like the used remote control interface, firmware version and status information on the performed capabilities check.

Instrument	Connected	Supported	Generator Profile	Mapped	Comment
 SMW200A, 0 VISA Resource: TCPIP::10.113.0.104::INSTR Serial: Firmware: 3.1.19.8-3.20.324.27		 Yes	SMW200A, 0	 OK	Manually Added

- "Instrument" Automatically retrieved instrument information, including instrument name, used remote control interface, and firmware version. Instruments that are currently not connected (not online) are displayed in grey; their "Connected" status is disabled.
- "Connected" Network connection status indication. It shows whether the instrument is connected to the local network or not. Execute "Refresh" to update the connection status indication of all listed instruments. If an instrument, that is being used in one of the scenarios, is not connected:
- This scenario can be configured to store the calculated waveform in ARB files (see also "[Load ARB File](#)" on page 62)
 - In a scenario with localized emitters, in the "Assign Signal Sources to Generator Profiles" dialog, the instrument is indicated as mapped but its color is red. (see also "[Generator Profiles](#)" on page 277). The scenario cannot be started, the "Scenario > Start" button is disabled.
- "Generator Profile" Assignment of a generator profile to the instrument. The capabilities of the instrument are compared with the requirements of this profile. See "[To find connected instruments and assign the generator profiles to them](#)" on page 245.
- "Linked" Status information on the performed capabilities check.
- "Comment" Additional information. Double click on the "Comment" field to change the displayed information.

Remote command:

[INSTrument:LIST?](#) on page 370

[INSTrument:SElect](#) on page 334

[INSTrument:MAP](#) on page 372

[INSTrument:COUNT?](#) on page 333

[INSTrument:FIRMWare?](#) on page 371

[INSTrument:LINK:STATus?](#) on page 372

Device Capabilities, Profile Capabilities

A side-to-side overview of the provided and required capabilities.

Remote command:

[INSTrument:CAPAbilities?](#) on page 370

14.2 Generator Profile Settings

Access:

- ▶ Perform one of the following:
 - Select "Repository Tree > Generator Profiles > New"
 - Open the automatically created default profile.

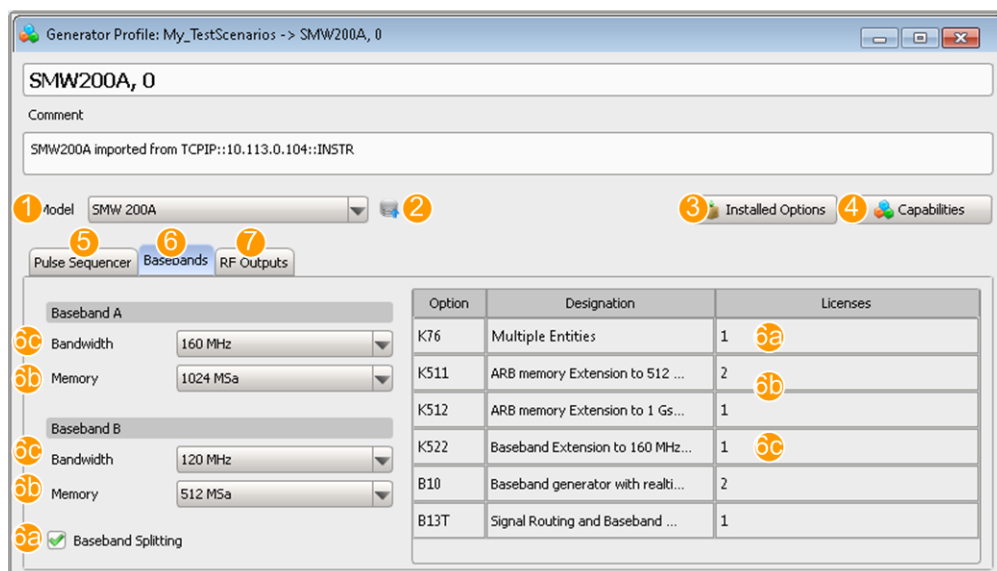


Figure 14-2: Generator Profiles: understanding the displayed information

- 1 = Generator type
- 2 = Retrieves the configuration from the selected connected instrument
- 3 = Displays a list of the required options
- 4 = Displays a list of the profile capabilities
- 5 = Installed options, required to play waveforms generated with the R&S Pulse Sequencer
- 6, 6a, 6b, 6c = Available baseband options and their features
- 7 = RF options and available RF outputs; in R&S SMW incl. connected external instruments like R&S SGT, R&S SGS, etc.

Available settings:

[Name](#).....241

[Comment](#).....241

[Model](#).....241

[Import configuration form a connected generator, List of available signal generators](#).....241

[Installed Options](#).....241

Capabilities.....	242
Pulse Sequencer.....	242
Basebands.....	243
RF Outputs.....	243

Name

Enter the name of the generator's profile.

Remote command:

[GENerator:NAME](#) on page 330

[GENerator:CREate](#) on page 329

[GENerator:CATalog?](#) on page 329

[GENerator:SElect](#) on page 330

[GENerator:REMove](#) on page 331

Comment

Enter a short description.

Remote command:

[GENerator:COMMeNt](#) on page 331

Model

Selects the type of the signal generator form the list of supported instruments.

The hardware characteristics of the selected generator influence the way the created waveform file is processed. In particular, the available ARB bandwidth limits the waveform length.

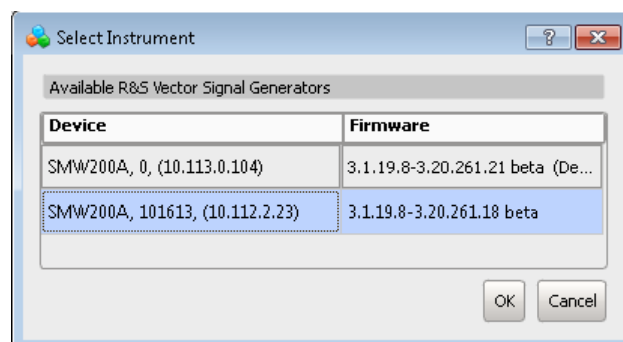
Remote command:

[GENerator:TYPE](#) on page 370

Import configuration form a connected generator, List of available signal generators

To retrieve the list of installed options form the connected generators, select the icon "Import configuration".

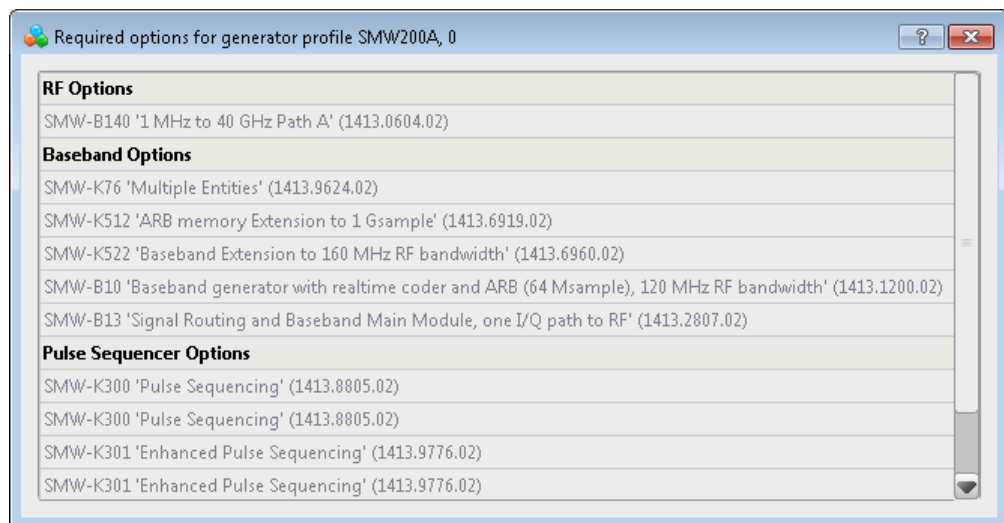
The "Select Instrument" dialog opens and lists all generators that are configured in the [Instrument Configuration](#) dialog.



To create the profile of a particular instrument, select it and confirm with "OK".

Installed Options

Displays an overview of the required options.

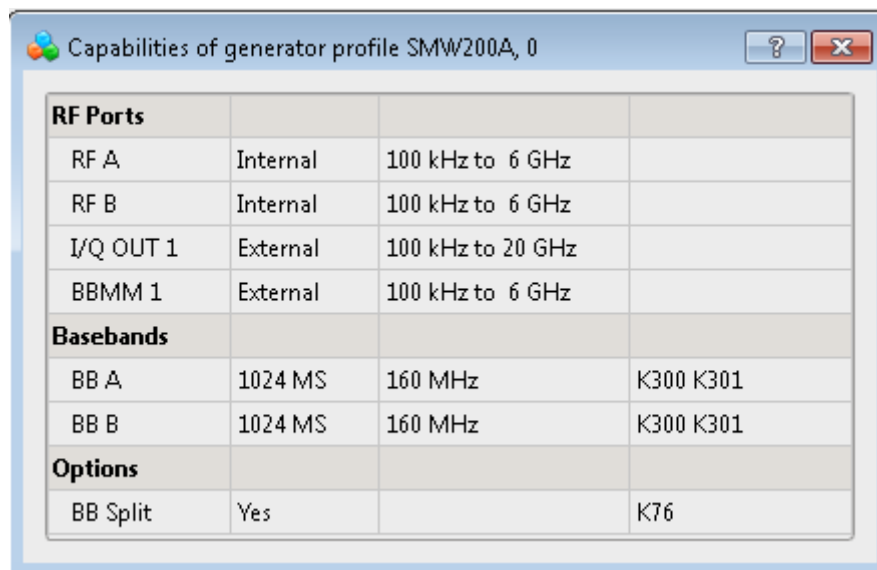


Remote command:

[GENerator:CAPabilities](#) on page 370

Capabilities

Displays an overview of the generator capabilities regarding supported scenario types, processing of waveforms and multi-segment files, and available RF paths.



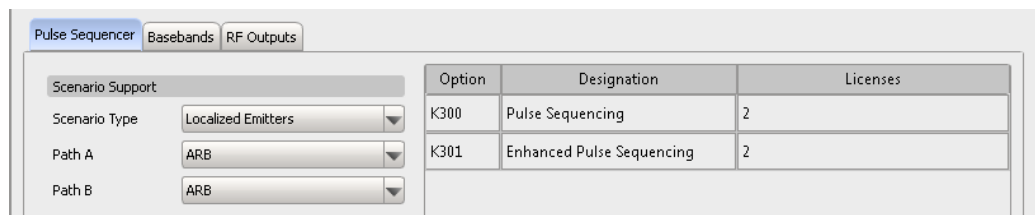
Remote command:

[GENerator:CAPabilities](#) on page 370

Pulse Sequencer

Indicates the options, that are required to play the selected scenario type.

The information in the [Capabilities](#) dialog is updated.



Basebands

Sets the baseband-related parameters and displays a list of the required baseband options, see [Figure 14-2](#).

"Bandwidth" Sets the maximum supported baseband bandwidth.

"Memory Extension"
Sets the total available ARB memory.

"Baseband Splitting"
Enables the R&S SMW to use more than 4 basebands.

Remote command:

See [GENerator:CAPabilities](#) on page 370

RF Outputs

Sets the maximum RF frequency per internal and external RF output and displays the required options.

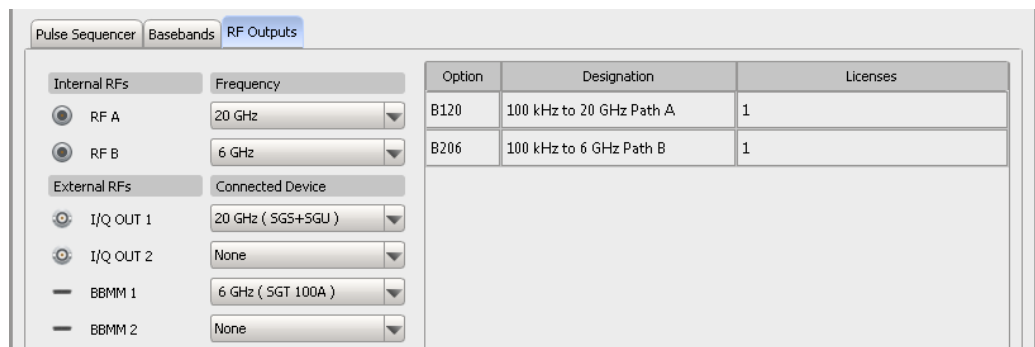


Figure 14-3: RF Outputs

"Output connector"
Click on the connector indicator to access the "Find Connector" dialog. The physical location of the current connector on the front/rear panel of the instrument is shown.

"Frequency, Connected Device"
Sets the maximum RF frequency, per RF output incl. connected external instruments.

Remote command:

See [GENerator:CAPabilities](#) on page 370

14.3 How to Create and Map Generator Profiles

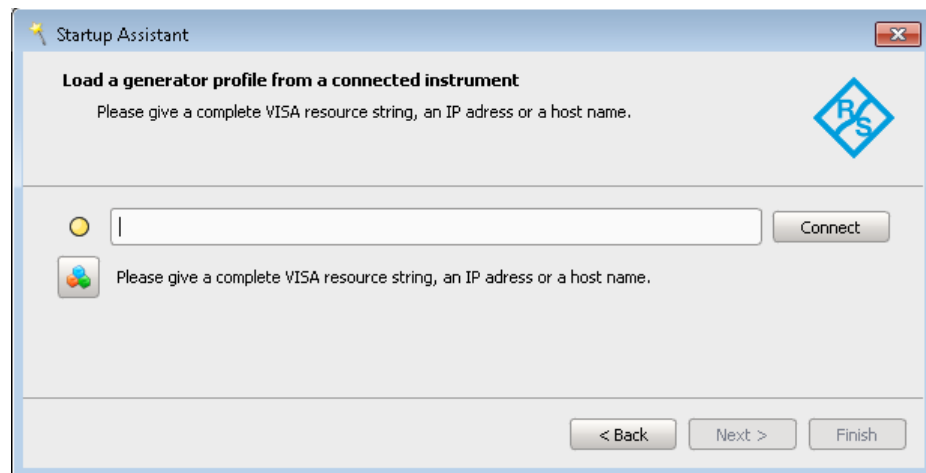
Configure the remotely connected instrument connections first, see:

- ["To find connected instruments and assign the generator profiles to them"](#) on page 245
- ["To retrieve the profile of the connected generator"](#) on page 246
- ["To assign a connected generator to a scenario"](#) on page 247
- ["To create a generator profile with the "Startup Assistant""](#) on page 244

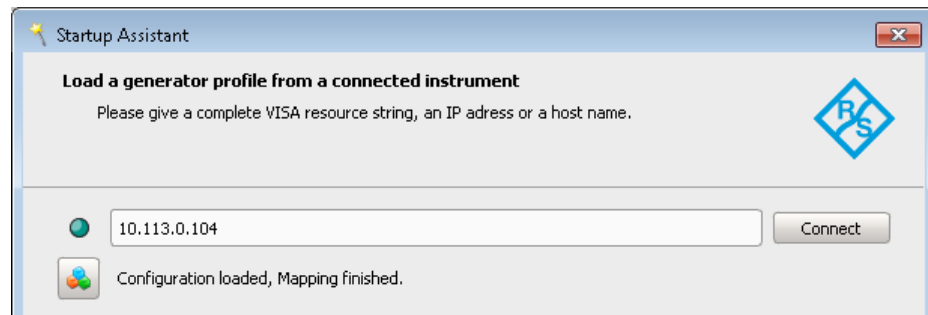
To create a generator profile with the "Startup Assistant"

You can use the "Startup Assistant" to create generator profiles from the connected instruments automatically.

1. In the menu bar, select "Help > Wizard".
2. Select "Create a new generator profile form a connected instrument" and confirm with "Next".



3. Enter the VISA resource string, the IP address or the host name of the instrument. For example, enter *10.113.0.104*. See also [Figure 14-1](#).
4. Select "Connect".

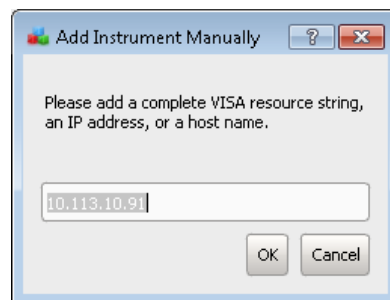


5. Observe the status LED and the displayed information.

A green LED confirms that the required generator profile is created and mapped to the instrument.

To find connected instruments and assign the generator profiles to them

1. Connect a signal generator and the PC the R&S Pulse Sequencer software is running on to the LAN, see [Figure 4-1](#).
2. To open the "Profile to Instrument Links" dialog, perform one of the following:
 - Select "Menu bar > Configure > Profile to Instrument Links".
 - In the "Scenario" dialog, select "Upload to VSG/Local ARB file > context menu > Generator Link".
3. To add an instrument, perform one of the following:
 - Select "Scan LAN".
 - Select "Add", and enter the IP address, the computer name, or the complete VISA resource string of the connected instrument and select "Ok".



See description R&S®SMW200A Getting Started.

The LAN is scanned for connected instruments. If suitable instruments are found, the dialog shows them, together with information on their capabilities.

4. Select "Profile" and select a generator profile from the list.

The instrument capabilities are compared to the profile requirements in regard of processing the current configured scenario and pulse sequence.
See [Figure 14-1](#)

Note: The connected instrument requires at least one option R&S SMW/SMBV/SGT-K300 to play ARB files generated with R&S Pulse Sequencer.
5. Observe the status indication.

- **Green LED/"OK"**: indicates a suitable instrument's configuration
- **Yellow LED/"N/A"**: indicates an unknown instrument or unsuccessful mapping
- **Red LED/"Fail"**: indicates that the requirements are not matched, e.g. missing option, insufficient amount of memory, etc.

For description of the provided settings, see [Chapter 14.2, "Generator Profile Settings"](#), on page 240.

To retrieve the profile of the connected generator

If a physical generator has been configured in the "Profile to Instrument Links" dialog, you can create its profile automatically.

Use one of the following:

1. In the "Profile to Instrument Links" dialog:
 - a) Select a connected generator.
 - b) Select "Gen Profile".
2. In the repository tree, create a "Generator Profiles" and:
 - a) Select "Generator" and select the generator type.
 - b) Select the icon "Import configuration from connected generator".
 - c) In the list of the "Available signal generators", select an instrument and confirm with "OK".

The created profile matches the capabilities of the connected generator.

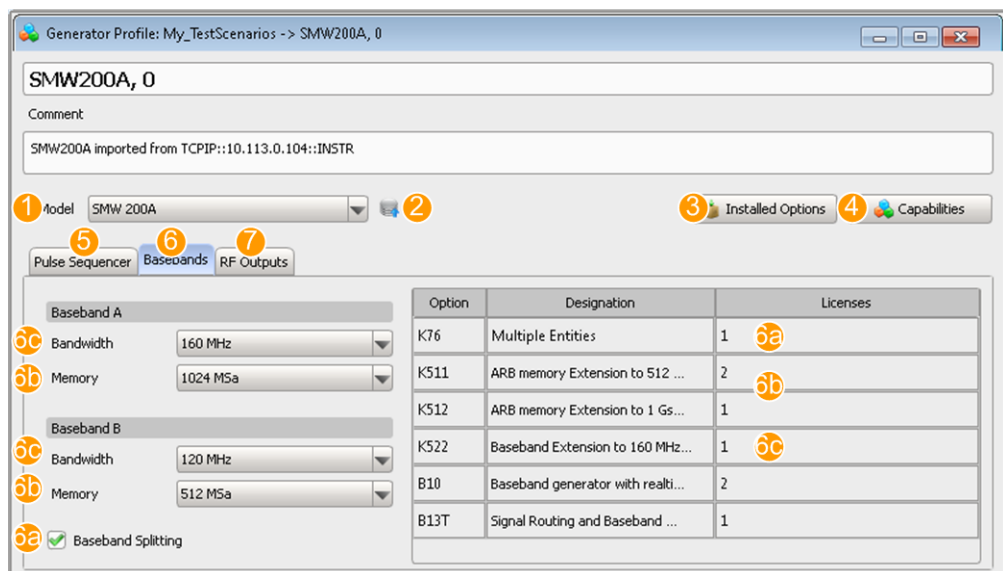


Figure 14-4: Generator Profiles: understanding the displayed information

- 1 = Generator type
- 2 = Retrieves the configuration from the selected connected instrument
- 3 = Displays a list of the required options
- 4 = Displays a list of the profile capabilities

- 5 = Installed options, required to play waveforms generated with the R&S Pulse Sequencer
6, 6a, 6b, 6c = Available baseband options and their features
7 = RF options and available RF outputs; in R&S SMW incl. connected external instruments like R&S SGT, R&S SGS, etc.

To assign a connected generator to a scenario

1. Open an existing scenario, e.g. "SimplePulseTrain".
2. Select "Upload to VSG > context menu" and select:
 - a) "Target > Generator"
 - b) "Select" and select a generator's profile, e.g. "R&S_SMW".The displayed information confirms the selected profile.
3. Observe the status indication.
 - **Green LED:** indicates that the selected profile is suitable for the current scenario;
If all other prerequisites are fulfilled, the "Start" function is active.
 - **Yellow LED:** indicates insufficient capabilities
 - **Red LED:** indicates that no profile is assigned

See [Chapter 17, "Playing the Generated Waveform Files"](#), on page 272.

15 Creating Complex 2D Scenarios with Receiver, Emitters and Interferers

The dedicated scenario types for complex 2D emitter-receiver scenarios are:

- "Localized Emitters"
- "Direction Finding"

Complex 2D emitter-receiver scenarios can include the following elements:

- Receiver
Simple or complex, depending on the scenario.
There is exact one receiver per gaming area (2D map).
The receiver is described by one or more antenna elements, static position and attitude information.
- Emitters
Simple or complex, depending on their definition.
Emitters are described by the combination of an operating mode, antenna pattern, antenna scan and a sequence, static position and attitude information.
- Interferers
Interferers are simulated by waveforms. If placed on a 2D map, interferers are treated like emitters.
They are described by a waveform, an antenna pattern, an antenna scan, static position and attitude information.
- Background emitters
Background emitters are simulated by waveforms. They are a source of a noise-like pulsed background signal, broadcasted with equal power in all directions.

15.1 2D Map Settings

Access:

- ▶ Perform one of the following:
 - In a "Scenario Type = Localized Emitters", select "Emitters > 2D".
 - In a "Scenario Type = Direction Finding", select "Map > 2D".

The name of the dialog changes depending on the scenario but the settings and the workflow are similar.

The 2D diagram displays the configuration on a map with North/East coordinates.

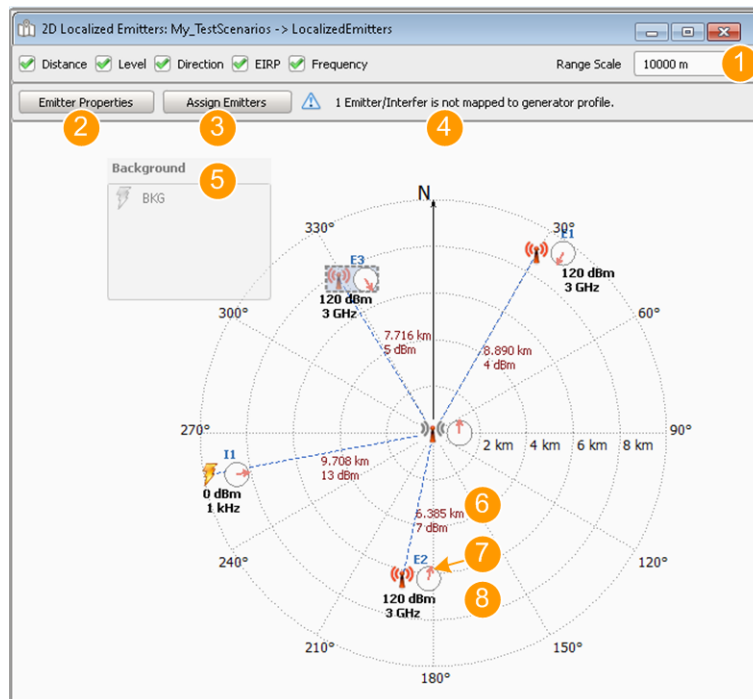


Figure 15-1: 2D view: understanding the displayed information

- 1 = Display options; the [Range Scale](#) for example defines the radius of the displayed 2D map
- 2 = Access the [Lists with Multiple Emitters and Interferers](#) Emitters/Interferers Properties dialog
- 3 = Access the [Assign Signal Source to Generator Profiles](#) dialog
- 4 = Mapping status indication
- 5 = Background signal and background emitters; see [Background Emitters settings](#)
- 6 = Distance to the receiver and [Strength of the received signal](#)
- 7 = "Point to Receiver = On" (antenna is automatically turned in the direction of the receiver)
- 8 = Emitter (or interferer) indicated with its alias name, EIRP, and transmission frequency

Settings:

Distance, Level, Direction, EIRP, Frequency	249
Range Scale	249
Emitter Properties	250
Assign Emitters	250
Assign Receiver Signals	250
Background	250
Strength of the received signal	250
Emitter/Interferer position on the 2D map and distance to the receiver	250

Distance, Level, Direction, EIRP, Frequency

To change the displayed information, enable or disable the corresponding parameter.

Range Scale

Sets the radius of the 2D map.

To change the units, select "Menu bar > Configure > Settings > General > Distance Units".

See "[To change the general program settings](#)" on page 37.

Emitter Properties

Accesses the "Emitter/Interferer Properties" dialog, see [Chapter 15.3, "Lists with Multiple Emitters and Interferers"](#), on page 253.

Assign Emitters

Accesses the "Assign Signal Source to Generator Profiles" dialog.

See [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275.

Assign Receiver Signals

Accesses the "Assign Receiver Signals to Generator Profiles" dialog.

See [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275.

Background

Displays the sequences or the background emitters that compose the background signal.

Remote command:

See [SCENario:LOCalized:SEquence](#) on page 439

See [SCENario:DIRection:SEquence](#) on page 439

Strength of the received signal

The displayed value is the reference value, used to calculate the power at the receiver antenna output. It is the virtual power level at the receiver output for two *perfectly aligned isotropic antennas* of emitter and receiver. Hence, this value is the maximum signal level that can be expected at the receiver antenna output.

In practice, the received signal is weaker than the displayed signal strength. The following parameters influence the signal strength:

- Distance between the emitter and the receiver, that is the free space attenuation.
- Alignment of the antennas and used antenna patterns
- Antenna gain

Emitter/Interferer position on the 2D map and distance to the receiver

The 2D map indicates the current distance, calculated from the current emitter/interferer position on the 2D map.

To change the position of the emitter/interferer, use one of the following possibilities:

- Drag the emitter/interferer to the new position on the 2D map
- Open its context menu, select "Properties > Position" and change the coordinates, the "Attitude" and the "Distance".

15.2 3D Scan Pair View Settings

Access:

1. Perform one of the following:
 - In a "Scenario Type = Localized Emitters", select "Emitters > 2D".

- In a "Scenario Type = Direction Finding", select "Map > 2D".
2. In the 2D map, open the context menu of an "Emitter" and select "3D view with Receiver".

The settings change depending on the scenario but the differences are self-explanatory.

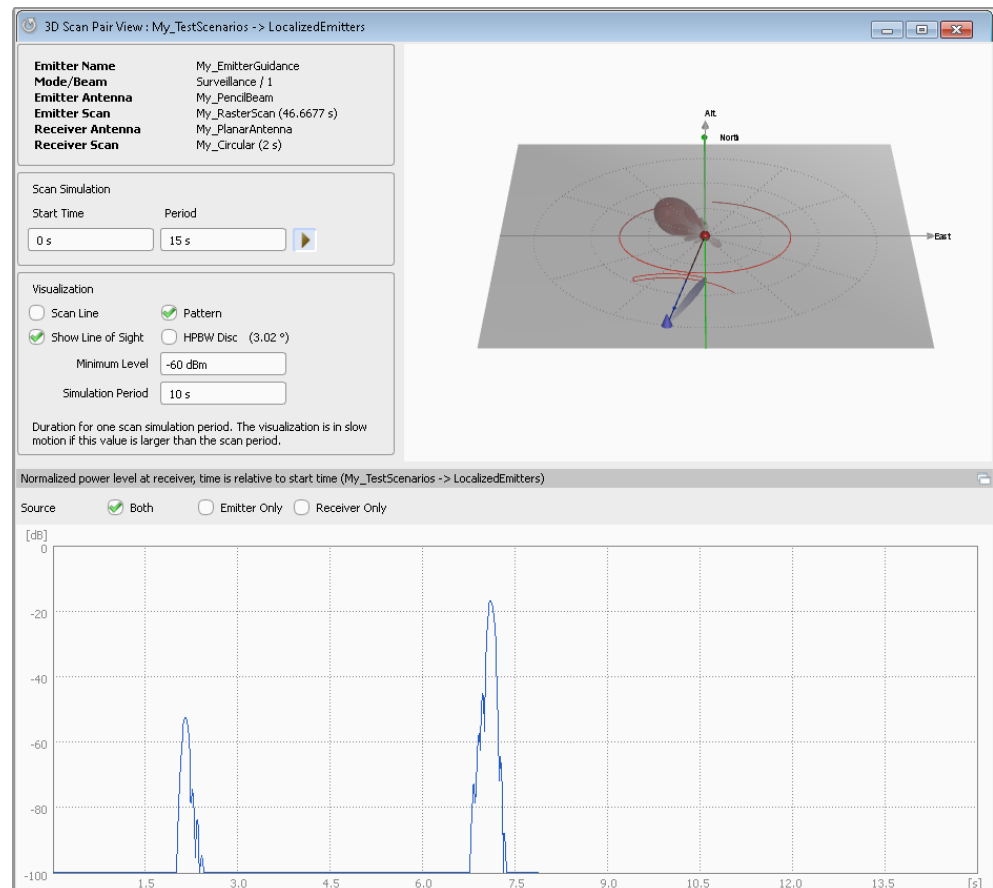



Figure 15-2: Example: "3D Scan Pair View" display in a "Localized Emitters" scenario

This dialog displays:

- A live plot of the normalized signal power level at the receiver.
- A 3D view of the receiver and emitter antennas with their patterns and scans.
- In "Direction Finding" scenario, the plot and the 3D view correspond to the selected receiver antenna element.



Use the split window icon  to detach the diagram and open it in a separate window.

Settings:

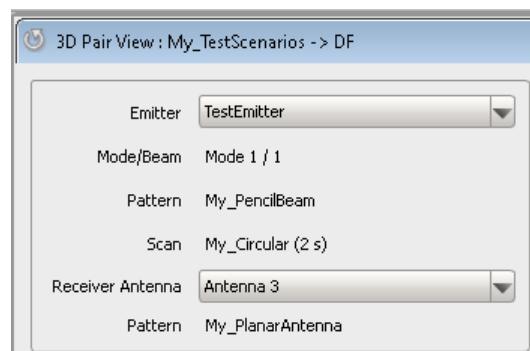
Live plot and 3D view configuration.....	252
Scan Simulation.....	252
Visualization.....	252
Both, Emitter only, Receiver only, Receiver Antenna.....	253

Live plot and 3D view configuration

Indicates the emitter and the receiver for that the 3D view and the live plot are displayed. Emitter and receiver are indicated with their major parameters:

- Emitter name, beam, antenna pattern, and antenna scan
- Receiver antenna pattern and scan

In "Direction Finding" scenario, you can switch between the receiver antenna elements.



The live plot and the 3D view are updated for each combination of emitter and antenna element.

Scan Simulation

Sets the time span visualized on the live plot as a start time and simulation period.

The "Period", i.e. the total duration the antenna scan simulation, determines the max. value on the x-axis. The visualization is relative to the selected "Start Time".

Use the "Pause" icon to stop the visualization.

Visualization

Defines the way the scan is represented:

- "Scan Line" The scan is visualized by a line; the antenna pattern is disregarded.
- "Pattern" The visualization corresponds to the current selected antenna pattern.
- "Show Line of Sight" Displays the line of sight (LOS) between the emitter and the receiver.
- "HPBW" The antenna is represented by its HPBW.
See also "[To visualize the signal on a 3D scan](#)" on page 262.
- "Minimum displayed level" Defines the minimum displayed side lobes level.
- "Simulation Period" Sets the time it takes the animation to complete a scan.

Both, Emitter only, Receiver only, Receiver Antenna

Defines the signal strength of which signal is visualized.

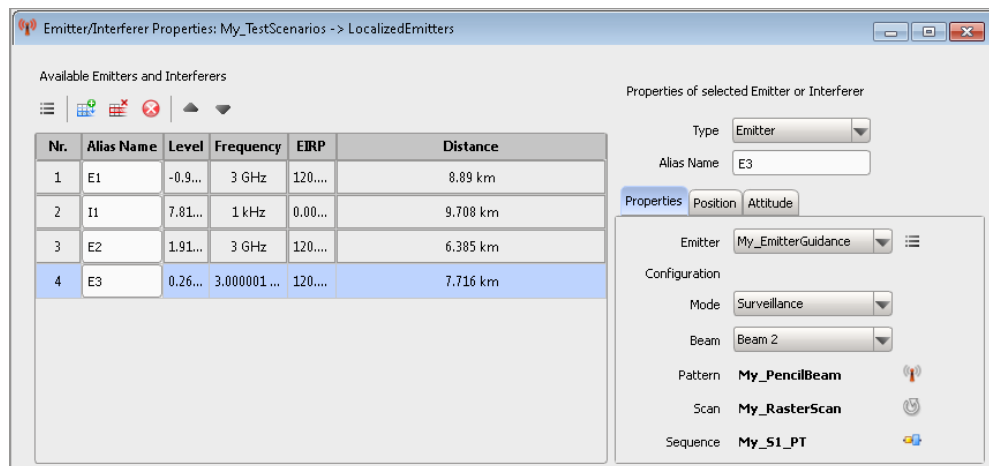
See also "To visualize the signal on a 3D scan" on page 262.

- "Both" Visualized is the resulting signal of both antennas and antenna scans.
- "Emitter only" Visualized is the signal of the emitter. It is assumed that the receiver has an omnidirectional antenna.
- "Receiver only" Visualized is the signal of the receiver. It is assumed that the emitter has an omnidirectional antenna.
- "Antenna" In "Direction Finding" scenario and if "Receiver Antenna > Show All Antennas" is selected, switches between the receiver antenna elements.

15.3 Lists with Multiple Emitters and Interferers

Access:

1. Perform one of the following:
 - In a "Scenario Type = Localized Emitters", select "Emitters > 2D".
 - In a "Scenario Type = Direction Finding", select "Map > 2D".
2. In the 2D map, select "Emitter Properties".



Available settings:

List of Available Emitters/Interferers..... 253

Properties of the selected Emitter/Interferer..... 254

Position..... 255

Attitude..... 257

List of Available Emitters/Interferers

Displays a list of emitters/interferers.

Use the standard functions in the context menu to add, reorder, or remove items.

See also ["To enable static emitters and place them on the 2D map"](#) on page 261.

Remote command:

[SCENario:LOCalized:ADD](#) on page 332

[SCENario:LOCalized:SElect](#) on page 333

[SCENario:LOCalized:CURRent](#) on page 434

[SCENario:LOCalized:DElete](#) on page 335

[SCENario:LOCalized:CLEar](#) on page 335

[SCENario:DIRection:ADD](#) on page 332

[SCENario:DIRection:SElect](#) on page 333

[SCENario:DIRection:CURRent](#) on page 434

[SCENario:DIRection:DElete](#) on page 335

[SCENario:DIRection:CLEar](#) on page 335

Properties of the selected Emitter/Interferer

Each emitter/interferer is described with the following parameters:

"Type" Defines whether an emitter or an interferer is configured.

Remote command:

[SCENario:LOCalized:TYPE](#) on page 439

[SCENario:DIRection:TYPE](#) on page 439

"Alias Name" Enters an alias name.

Remote command:

[SCENario:LOCalized:ALIAS](#) on page 433

[SCENario:DIRection:ALIAS](#) on page 433

"Emitter/Interferer Name"

Selects an emitter/interferer.

Remote command:

[SCENario:LOCalized:EMITter](#) on page 435

[SCENario:LOCalized:WAVEform](#) on page 439

[SCENario:DIRection:EMITter](#) on page 435

[SCENario:DIRection:WAVEform](#) on page 439

"EIRP" Sets the [EIRP](#) of the interferer.

Remote command:

[SCENario:LOCalized:WAVEform:EIRP](#) on page 440

[SCENario:DIRection:WAVEform:EIRP](#) on page 440

"Frequency" Sets the frequency of the interferer.

Remote command:

[SCENario:LOCalized:WAVEform:FREQuency](#) on page 440

[SCENario:DIRection:WAVEform:FREQuency](#) on page 440

"Mode" Selects the mode the emitter is working in.

Remote command:

[SCENario:LOCalized:EMITter:MODE](#) on page 435

[SCENario:DIRection:EMITter:MODE](#) on page 435

"Beam" Sets the number of current used beams.

Remote command:

[SCENario:LOCALized:EMITter:MODE:BEAM](#) on page 436

[SCENario:DIRection:EMITter:MODE:BEAM](#) on page 436

"Antenna Pattern, Scan, Sequence"

Displays the current selected antenna pattern, antenna scan, and sequence.

To change any of them, select the corresponding icon.

Remote command:

[SCENario:LOCALized:WAVEform:ANTenna](#) on page 438

[SCENario:LOCALized:WAVEform:SCAN](#) on page 438

[SCENario:DIRection:WAVEform:ANTenna](#) on page 438

[SCENario:DIRection:WAVEform:SCAN](#) on page 438

Position

The emitters and interferers on the 2D map are static elements with configurable coordinates.

To change the location of an element:

- Drag this element on the 2D map
- Use the "Position" parameters to define the location more precisely.

The emitter/interferer position can be defined as East/North coordinates and as angular positions. The definitions are interdependent.

Example:

- An emitter with the East/North coordinates "East = 0 m", "North = 7000 m" is placed on the North, i.e. the "Azimuth Angle = 0°".

Location parameters	Map view																					
<table border="1"> <thead> <tr> <th>Properties</th> <th>Position</th> <th>Attitude</th> </tr> </thead> <tbody> <tr> <td>East</td> <td>0 m</td> <td></td> </tr> <tr> <td>North</td> <td>7000 m</td> <td></td> </tr> <tr> <td>Height</td> <td>0 m</td> <td></td> </tr> <tr> <td>Distance</td> <td>7000 m</td> <td></td> </tr> <tr> <td>Azimuth Angle</td> <td>0 °</td> <td></td> </tr> <tr> <td>Elevation Angle</td> <td>0 °</td> <td></td> </tr> </tbody> </table>	Properties	Position	Attitude	East	0 m		North	7000 m		Height	0 m		Distance	7000 m		Azimuth Angle	0 °		Elevation Angle	0 °		
Properties	Position	Attitude																				
East	0 m																					
North	7000 m																					
Height	0 m																					
Distance	7000 m																					
Azimuth Angle	0 °																					
Elevation Angle	0 °																					

- If the "Azimuth Angle = 90°", the emitter is placed on the East axis, i.e. "North = 0 m" and "East = 7000 m".

Location parameters	Map view																					
<table border="1"> <thead> <tr> <th>Properties</th> <th>Position</th> <th>Attitude</th> </tr> </thead> <tbody> <tr> <td>East</td> <td>7000 m</td> <td></td> </tr> <tr> <td>North</td> <td>0 m</td> <td></td> </tr> <tr> <td>Height</td> <td>0 m</td> <td></td> </tr> <tr> <td>Distance</td> <td>7000 m</td> <td></td> </tr> <tr> <td>Azimuth Angle</td> <td>90 °</td> <td></td> </tr> <tr> <td>Elevation Angle</td> <td>0 °</td> <td></td> </tr> </tbody> </table>	Properties	Position	Attitude	East	7000 m		North	0 m		Height	0 m		Distance	7000 m		Azimuth Angle	90 °		Elevation Angle	0 °		
Properties	Position	Attitude																				
East	7000 m																					
North	0 m																					
Height	0 m																					
Distance	7000 m																					
Azimuth Angle	90 °																					
Elevation Angle	0 °																					

See also ["To enable static emitters and place them on the 2D map"](#) on page 261.

Remote command:

[SCENario:LOCALized:LOCation:EAST](#) on page 437

[SCENario:LOCALized:LOCation:NORTH](#) on page 437

[SCENario:LOCALized:LOCation:HEIGHT](#) on page 438

[SCENario:LOCALized:DISTANCE?](#) on page 436

[SCENario:LOCALized:LOCation:AZIMuth](#) on page 437

[SCENario:LOCALized:LOCation:ELEVation](#) on page 438

[SCENario:DIRection:LOCation:EAST](#) on page 437

[SCENario:DIRection:LOCation:NORTH](#) on page 437

[SCENario:DIRection:LOCation:HEIGHT](#) on page 438

[SCENario:DIRection:DISTANCE?](#) on page 436

[SCENario:DIRection:LOCation:AZIMuth](#) on page 437

[SCENario:DIRection:LOCation:ELEVation](#) on page 438

Attitude

Properties	Position	Attitude
		<input type="checkbox"/> Point to Receiver
		Yaw <input type="text" value="45 °"/>
		Pitch <input type="text" value="0 °"/>
		Roll <input type="text" value="0 °"/>

Defines the orientation of the antennas as "Pitch", "Yaw", and "Roll".

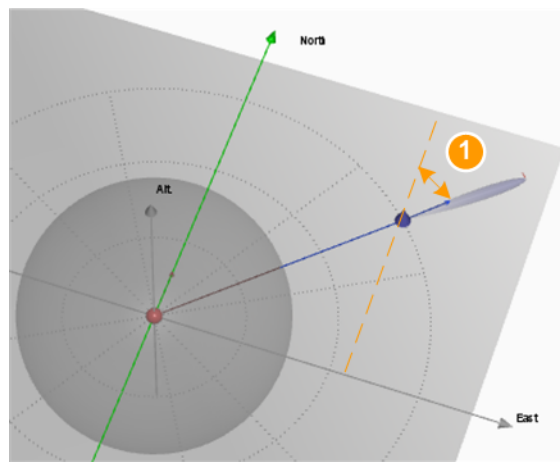


Figure 15-3: Yaw: understanding the displayed information

1 = Yaw is the angle between the antenna's boresight and North

If "Point to Receiver" is enabled, the antenna is automatically turned in the direction of the receiver.

See also "[Emitter attitude](#)" on page 196.

Remote command:

[SCENario:LOCalized:DIRection:TRACk](#) on page 436

[SCENario:LOCalized:DIRection:PITCh](#) on page 435

[SCENario:LOCalized:DIRection:YAW](#) on page 434

[SCENario:LOCalized:DIRection:ROLL](#) on page 435

[SCENario:DIRection:DIRection:TRACk](#) on page 436

[SCENario:DIRection:DIRection:PITCh](#) on page 434

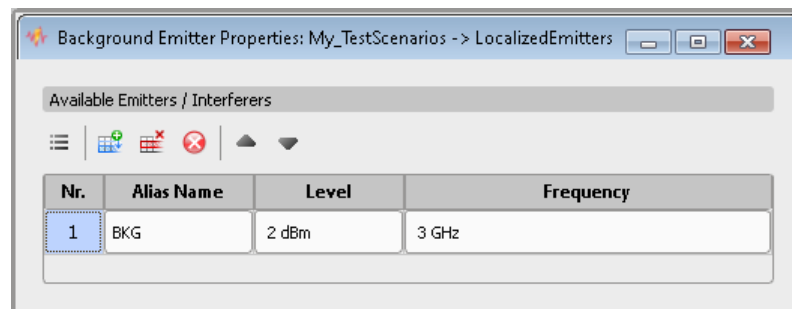
[SCENario:DIRection:DIRection:YAW](#) on page 434

[SCENario:DIRection:DIRection:ROLL](#) on page 435

15.4 Lists with Multiple Background Emitters

Access:

1. Perform one of the following:
 - In a "Scenario Type = Localized Emitters", select "Emitters > 2D".
 - In a "Scenario Type = Direction Finding", select "Map > 2D".
2. In the 2D map, select the "Background" block and select "Emitter Properties".



Settings:

[List of available background emitters and their properties](#).....258

List of available background emitters and their properties

Displays a list of the sequences or waveforms used to create the individual background signals.

Use the standard functions in the context menu to add, reorder, or remove items.

Background emitters are identified by their "Alias Name", "Level" and used "Frequency". Background emitters do not have antennas, specific position on the 2D map or direction.

See also the instruction "[To add a background signal](#)" on page 260.

Remote command:

See the commands listed in:

["List of Available Emitters/Interferers"](#) on page 253 and

["Properties of the selected Emitter/Interferer"](#) on page 254

15.5 How to Create Scenarios with Emitters, Interferers and a Receiver

If your test situation requires to test the receiver's ability to detect the sum signal from different static emitters and interferers, you can use "Localized Emitter" scenario. In this dedicated scenario, you configure the signal of one or more emitters and optional interferers that are received by a receiver with defined characteristics. You can also configure the receiver and change its position in the scanning beam of the emitters.

General workflow for creating complex 2D scenarios

To explain the principle, the following examples create a complex "Localized Emitter" scenario. The emitters and interferers that are located at a given distance to the receiver and placed on a 2D map. The receiver characteristics are also configured.

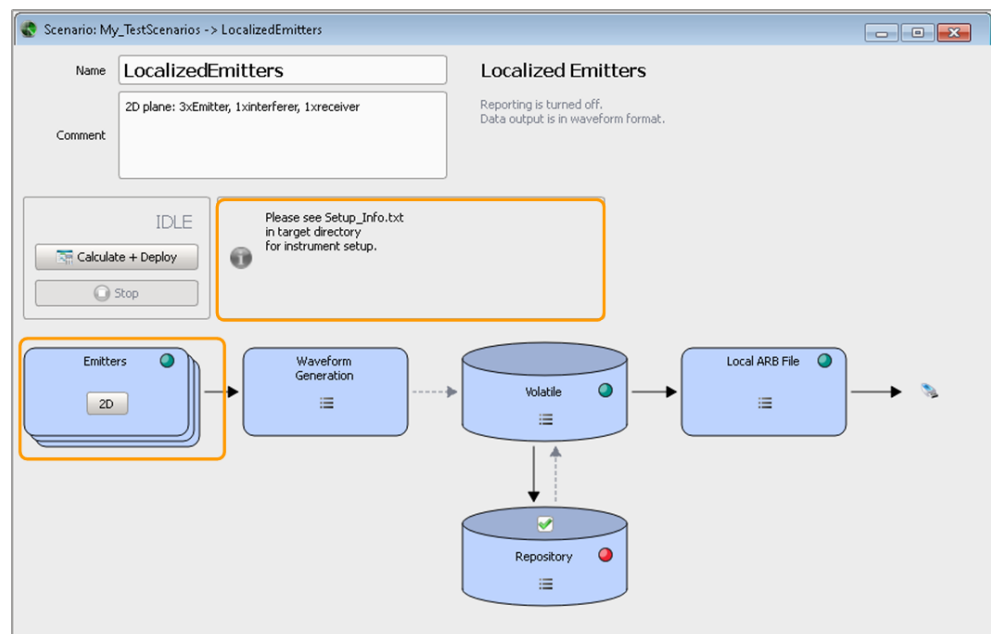


To create "Direction Finding" scenarios, follow the same workflow.

- "Open a suitable scenario" on page 259
- "To add an existing waveform as an interferer to the scenario" on page 260
- "To enable static emitters and place them on the 2D map" on page 261
- "To configure the receiver settings" on page 262
- "To visualize the signal on a 3D scan" on page 262
- "To create, load and play the waveforms of a 2D scenario automatically" on page 287

Open a suitable scenario

1. Create a scenario: "Scenario > Localized Emitters".



2. In the block diagram, select "Emitter > 2D".

The "2D Localized Emitters" dialog opens and displays a 2D view of the receiver and currently configured emitters and interferers, together with their main characteristics.

See [Figure 15-1](#).

3. Enable and configure emitters, interferers, and the receiver, see:
 - "To enable static emitters and place them on the 2D map" on page 261

- ["To add an existing waveform as an interferer to the scenario"](#) on page 260
 - ["To configure the receiver settings"](#) on page 262
4. Use the context menu to:
 - a) Display the 3D antenna patterns and scans
 - b) To add/remove an emitter or an interferer
 5. Use the mouse wheel to change the scale of the 2D diagram, i.e. to change the distance between the receiver and the emitters/interferers.

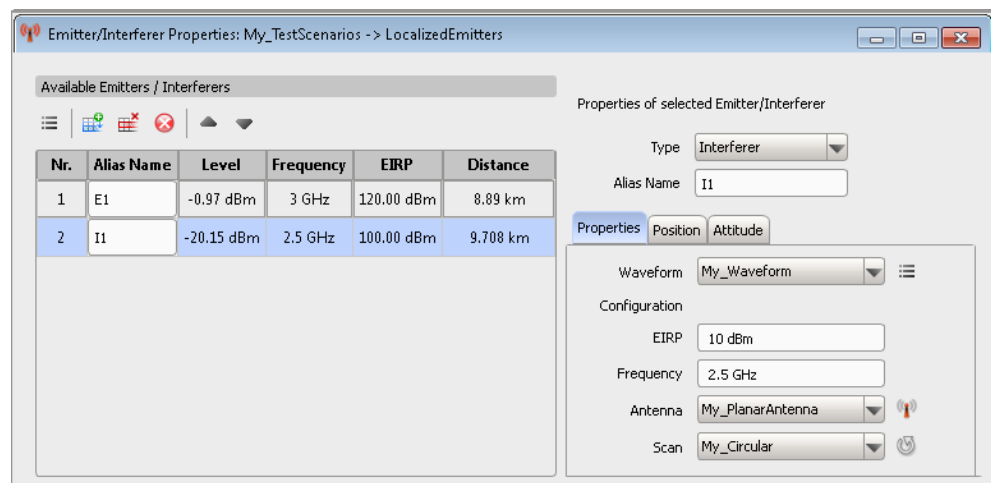
For description of the provided settings, see:

- [Chapter 15.1, "2D Map Settings"](#), on page 248
- [Chapter 15.2, "3D Scan Pair View Settings"](#), on page 250

To add an existing waveform as an interferer to the scenario

1. To add an interferer, select a waveform in the repository tree and drag&drop it into the 2D diagram.

The "Emitter/Interferer Properties" dialog opens and lists all emitter and interferers configured in the scenario.



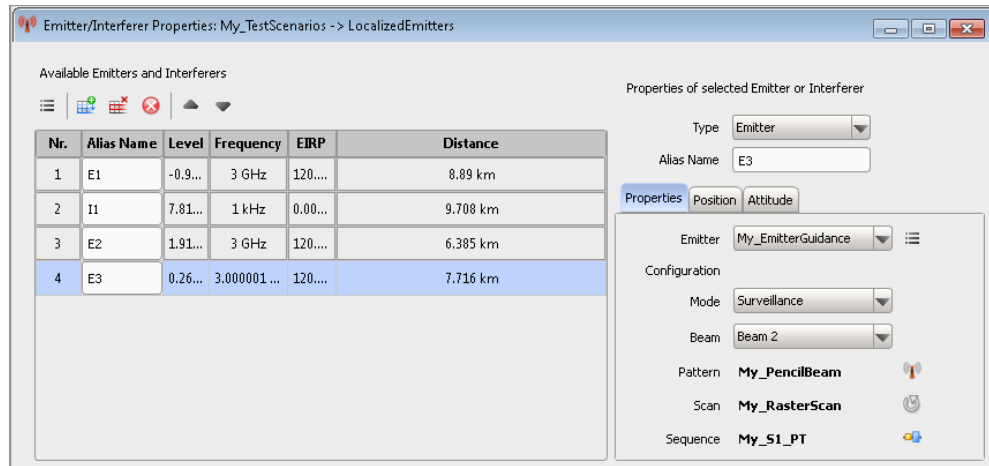
2. Configure the interferers's properties, for example:
 - a) Assign one of the existing antenna patterns or create a new one
 - b) Define the position more precisely
 - c) Change the attitude (orientation) of the antenna.

To add a background signal

1. In the repository tree, select a waveform or a sequence.
2. Drag the waveform (or the sequence) into the 2D diagram and drop it in the "Background" box.
3. Open the context menu and adjust the signal level and frequency.

To enable static emitters and place them on the 2D map

1. To add an emitter, drag&drop an emitter it into the 2D diagram.
2. Configure the emitter's properties.



The 2D diagram displays the configuration on a map with North/East coordinates.

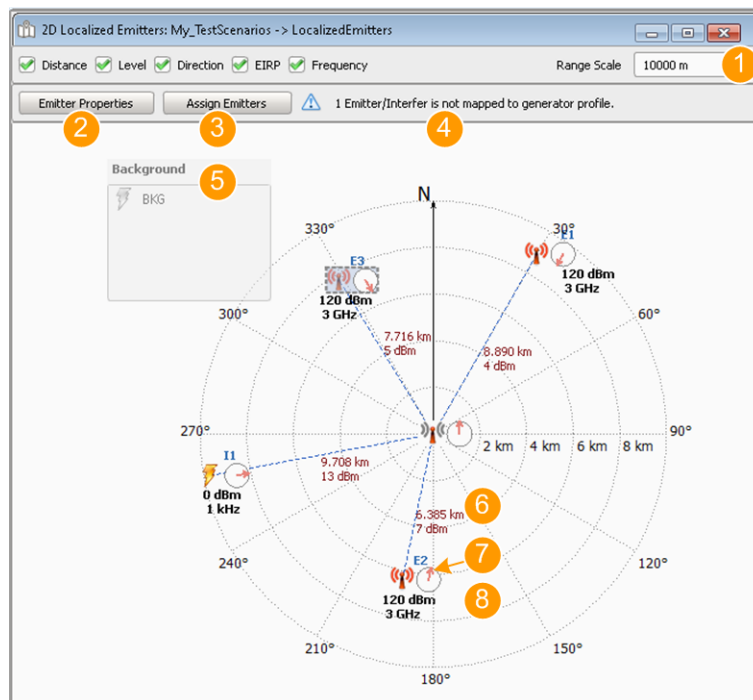


Figure 15-4: 2D view: understanding the displayed information

- 1 = Display options; the **Range Scale** for example defines the radius of the displayed 2D map
- 2 = Access the **Lists with Multiple Emitters and Interferers** **Emitters/Interferers Properties** dialog
- 3 = Access the **Assign Signal Source to Generator Profiles** dialog
- 4 = Mapping status indication
- 5 = Background signal and background emitters; see **Background Emitters settings**

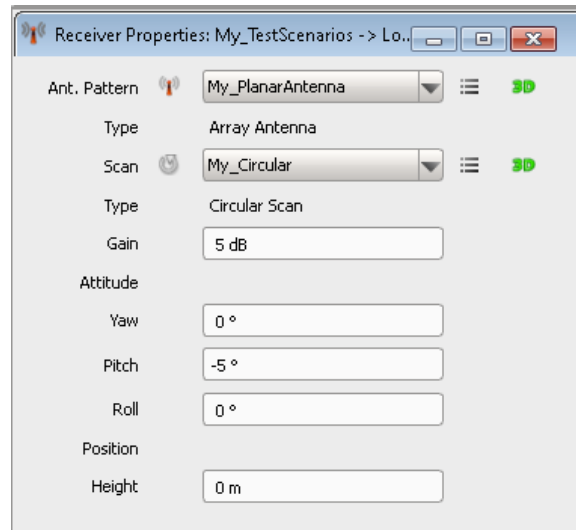
6 = Distance to the receiver and [Strength of the received signal](#)

7 = "Point to Receiver = On" (antenna is automatically turned in the direction of the receiver)

8 = Emitter (or interferer) indicated with its alias name, EIRP, and transmission frequency

To configure the receiver settings

- In the 2D diagram, open the context menu of the receiver and select "Properties".



For description of the provided settings, see [Chapter 12.1, "Receiver Properties"](#), on page 213.

To visualize the signal on a 3D scan

1. In the 2D diagram, open the context menu of an emitter/interferer and select "3D View with Receiver".
2. Select "Normalized power Level at Receiver > Both > On" to display the resulting signal.

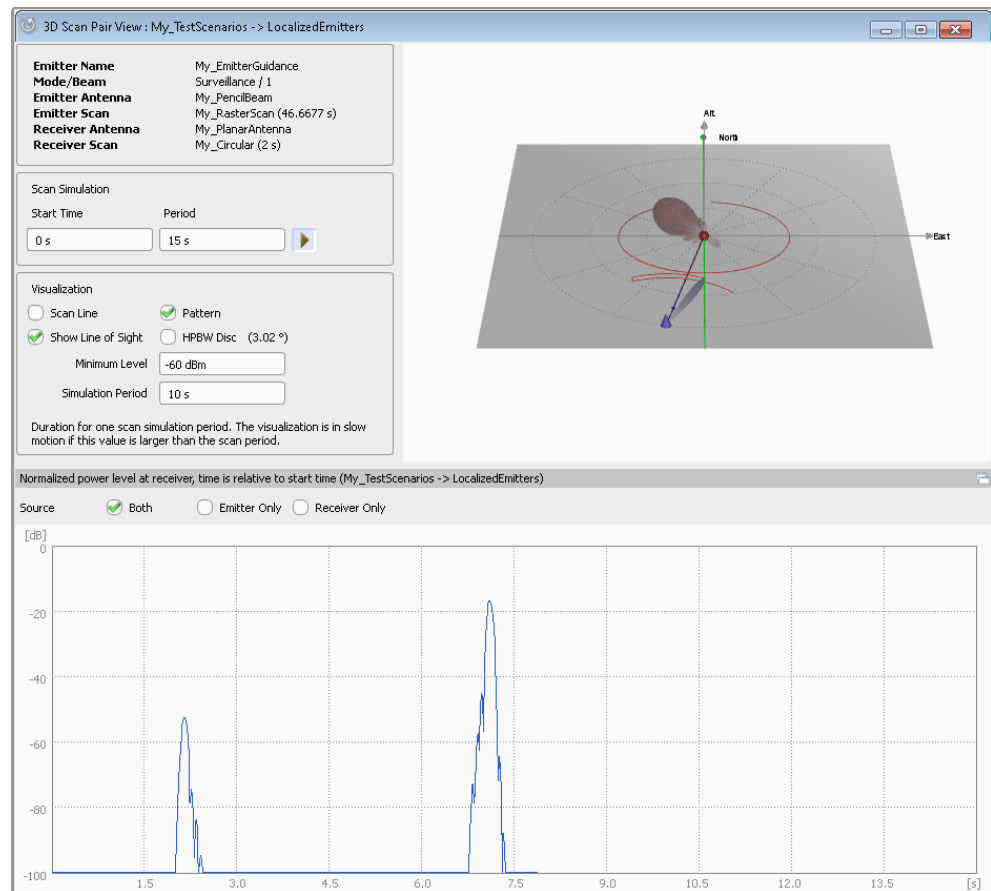


Figure 15-5: Example: "3D Scan Pair View" display in a "Localized Emitters" scenario

- To display only the signal at the receiver, enable "Emitter Only".
The calculation assumes a receiver with an omnidirectional antenna pattern.
- Compare the displayed signal with the signal received by a static receiver in a "Single Emitter" scenario with the same emitter "E2".
See [Figure 11-3](#).

For description of the provided settings, see [Chapter 15.2, "3D Scan Pair View Settings"](#), on page 250.

To map the generated signals to the signal generator

In "Scenario Type > Localized Emitters" and "Scenario Type > Direction Finding", you have to select the signal of which emitter/interferer or receive antenna is output on which signal generator RF path.

- ▶ Follow the instructions in ["To create, load and play the waveforms of a 2D scenario automatically"](#) on page 287.

16 Visualizing and Analyzing Waveforms

Waveform generated with the software or imported to the repository can be visualized with the built-in "Waveform/Data View" function. The R&S Pulse Sequencer reads a generated waveform, evaluates it and displays the I and Q data, the spectrum, and the constellation diagram of a signal section or of the entire waveform. The "Waveform/Data View" is only available if a waveform was created successfully.

Imported waveforms can also be evaluated to retrieve information on the reference signal level and to change it.

16.1 Waveform and Data View Settings

The waveform viewer reads the entire waveform data, evaluates it and visualizes the signal in three sections: The time domain view, the spectrum/spectrogram, and the I/Q constellation plot.

Access:

- In the "Scenario" dialog, select "Volatile > View".

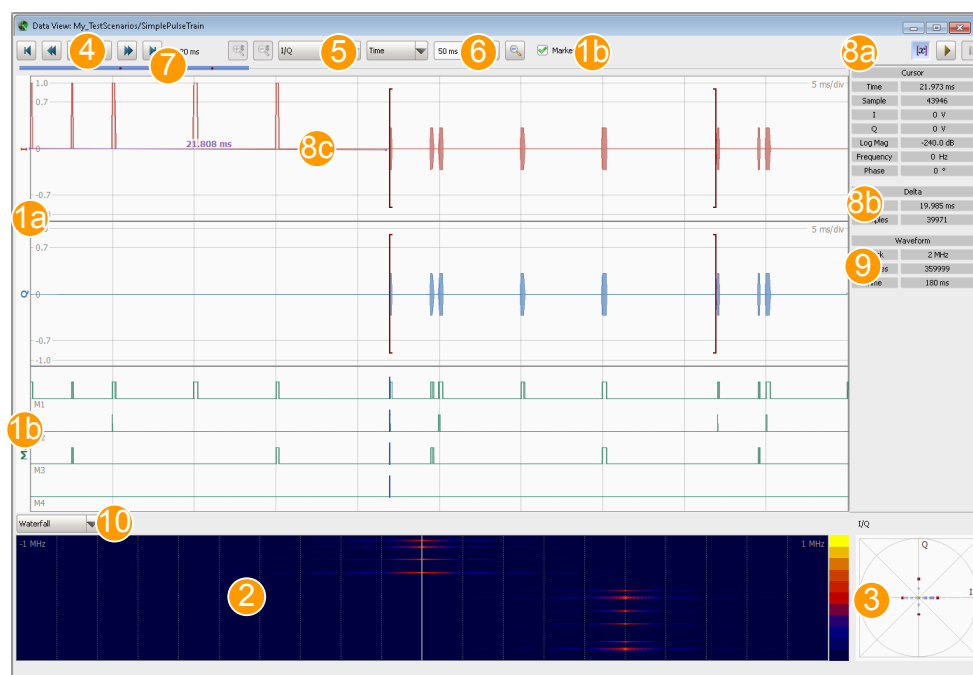


Figure 16-1: Waveform/Data View: Understanding the displayed information

- 1a = Time domain view, "View mode = I/Q Envelope" (screen points represent multiple signal samples)
- 1b = Enabled marker traces (M1 = all, M2 = first, M3 = last), see [Marker traces](#)
- 2 = Spectrogram in the default rainbow colors, see [Display Mode](#)
- 3 = I/Q constellation plot
- 4 = Navigation controls

- 5 = [View mode](#), e.g. I/Q data, frequency, phase, magnitude
- 6 = Zoom in/out and signal part as number of samples or a time duration, see [Duration/Samples](#)
- 7 = [Time line](#)
- 8a = Delta cursors, play and stop; see [Play/Stop](#)
- 8b, 8c = Retrieved delta information
- 9 = File information: clock rate, number of samples and duration
- 10 = [Display Mode](#) selection

Available settings:

[Navigation controls](#).....265

[View mode](#)..... 265

[Units](#)..... 265

[Zoom in/out](#)..... 266

[Marker traces](#)..... 266

[Duration/Samples](#).....266

[Time line](#).....266

[\[x\] Delta cursors, Delta markers](#).....266

[Play/Stop](#).....267

[Display Mode](#)..... 267

[I/Q constellation plot](#).....267

Navigation controls

Steps forward or backward to the beginning or end of the waveform.

View mode

Switches between the trace types.

The dialog size and the screen resolution limit the number of screen points in the time domain display. They also limit the resolution at which level and time information is represented. Particularly when a large signal section is analyzed, one single screen point can contain the information from many signal samples.

Depending on the zoom level, each screen point may represent multiple signal samples. If so, the area between the minimum and maximum value is filled with a semi-transparent trace color.

- "I/Q" I and Q trace on a linear scale in the range between -1.0 and +1.0
- "Magnitude dB/Watt/Volt" Signal amplitude on a 100 dB/1000 mW/100 mV scale
- "Phase" Phase trace on a linear scale in the range between -Pi and +Pi
- "Frequency" Frequency trace on a linear scale with adjustable scale
- "Average Power" Average value when multiple samples are represented by one screen point.
See also [Chapter 16.2, "Waveform Reference Level Settings"](#), on page 267.

Remote command:

`SCENario:VOLatile:VIEW:YMODe` on page 461

Units

Sets the units (time or samples) used on the x axis.

Remote command:

`SCENario:VOLatile:VIEW:XMODE` on page 461

Zoom in/out

Zooms in and out, i.e. changes the resolution on the x and y axis.

Remote command:

`SCENario:VOLatile:VIEW:ZOOM:POINT` on page 462

`SCENario:VOLatile:VIEW:ZOOM:RANGE` on page 462

Marker traces

In each of the view modes, enable "Marker > On" to display the up to four markers traces M1 to M4.

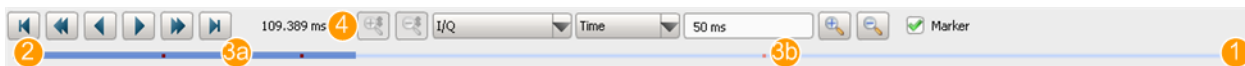
See also [Chapter 20, "Defining and Enabling Marker Signals"](#), on page 301.

Duration/Samples

Sets the number of displayed signal samples, e.g. to display only a part of the entire waveform.

The "Waveform Viewer" displays a maximum of 100000 Samples. To visualize a longer part of a waveform, you can reduce the clock rate used by the calculation, see [Clock, Duration](#).

Time line



- 1 = Time line
- 2 = Slider
- 3a, 3b = Delta cursor and mouse position (red dots)
- 4 = Time at mouse position

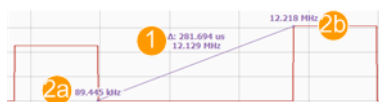
- The length of the timeline symbolizes the play time of the entire waveform.
- The blue slider indicates the area of the waveform that is visible in the viewport.
- The red dots indicate the time that corresponds to the cursors and mouse positions.
- The numerical value above the timeline indicates the time at the mouse position.
- A left mouse click moves the viewport center position to the selected time.
- Sliding the blue slider shifts the viewport, i.e. shifts the visible waveform area.

[x] Delta cursors, Delta markers

To retrieve delta information, use one of the following:

"Temporary delta information"

Drag the mouse cursor on the diagram.
The delta information is displayed temporary above the line.



- 1 = Delta information: upper value is the Δx [in the selected "Units"], lower value is the Δy [units depending on the selected "View mode"]
- 2a, 2b = Start and end y values [units depending on the selected "View mode"]

"[x] Delta cursors"

Select the "[x]" icon and Drag&Drop the cursors.

The delta cursors are indicated by brackets. The delta information is displayed in the "Delta" section of the dialog.

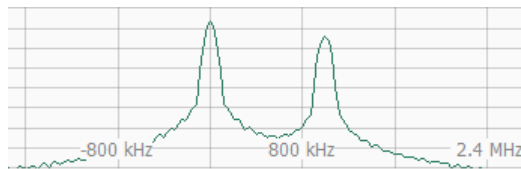
Play/Stop

Starts and stops the graphical waveform playback. The waveform section is shifted in time by one half of a display section every 100 ms.

Display Mode

Sets the display mode:

"Spectrum" The spectrum shows the **FFT** of the signal portion that is visible in the viewport. The FFT algorithm processes the waveform data in sections of 512 points.

**"Spectrogram"**

The spectrogram (waterfall diagram) shows the spectrum of a signal portion as a function of time. Each line in the spectrogram represents one FFT with 512 points.

The spectrogram uses a color scale to represent the dynamic range. You can change the default rainbow color scheme in the context menu of the diagram.

I/Q constellation plot

Depending on the number of samples visible in the viewport, displays the individual I/Q samples or turns to a density plot display.

16.2 Waveform Reference Level Settings

Access:

1. Select "Repository Tree > Waveform".

2. Select "Waveform > Set Ref. Level".

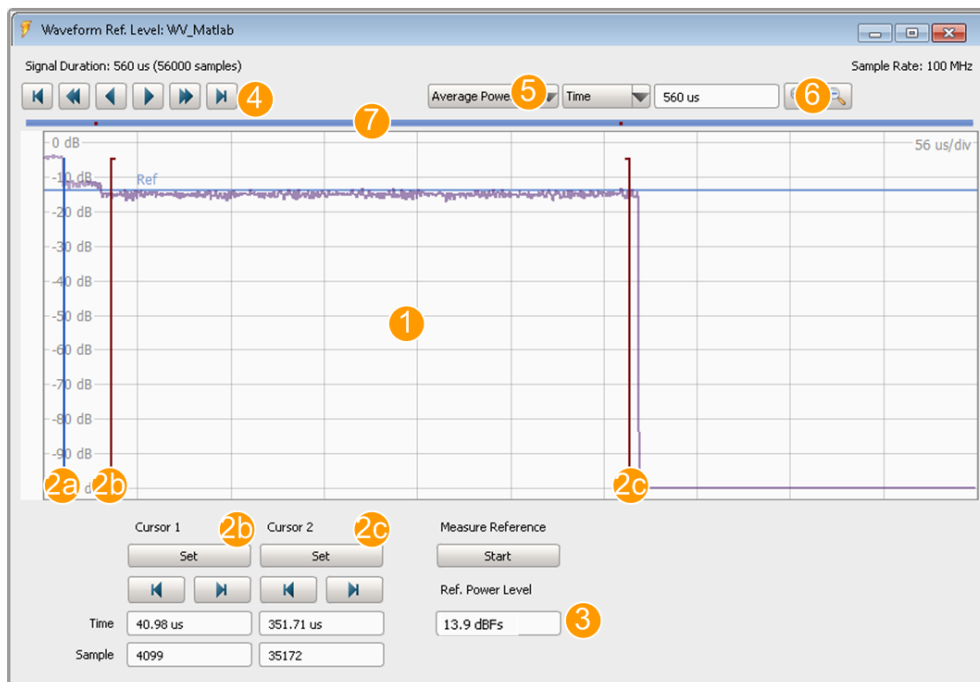


Figure 16-2: Waveform Reference Level: Understanding the displayed information

- 1 = Time domain view, View mode = Average Power
- 2a = Primary cursor
- 2b, 2c = Cursors
- 3 = Calculated reference level
- 4 = Navigation controls
- 5 = View mode, e.g. I/Q data, power, etc.
- 6 = Zoom in/out and signal part as number of samples or a time duration, see Duration/Samples
- 7 = Time line

The "Waveform Reference Level" reads the entire waveform data and provides a view of the I/Q data versus time. The provided navigation controls, zooming functions, and view modes are similar to the functions provided in the "Waveform View" dialog, see Figure 16-1.

The additional controls define the signal portion that is used for calculating of the reference signal level, see "To change the waveform reference level based on a user-definable signal section" on page 270.

Available settings:

Cursors.....268

Measure Ref. Level.....269

Cursors

Define the signal portion used for calculating of the reference signal level.

- The primary cursor is indicated by a single line. It marks the center point for all zoom operations.
- The other two cursors are indicated by brackets. They define the beginning and end point for the reference level evaluation period.

The "Set" buttons set the brackets to the current cursor position. Both brackets can also be set numerically by using the controls below the cursor navigation buttons.

Measure Ref. Level

If the imported file contains information on the reference level, this information is displayed as "Ref. Level". In files in the Rohde & Schwarz waveform file format (*.wv files), the reference level is present and correct, i.e. the corresponding tag is set.

If the tag is missing, the software evaluates all samples in the file to calculate the reference level. For some signals, such as pulses, the reference level calculated in this way is of little interest. In this case, you can select the waveform section to be evaluated, e.g. the pulse top power.

The reference level is recalculated for the particular waveform section (see the example in section ["To change the waveform reference level based on a user-definable signal section"](#) on page 270).

If a signal generator plays the waveform, the signal generator uses the reference level for leveling.

Remote command:

[WAVeform:LEVel:REFerence](#) on page 458

16.3 How to Analyze the Content of Waveform Files and Files with I/Q Data

See:

- ["To access the "Waveform/Data View" dialog"](#) on page 269
- ["To adjust the display and retrieve more information on the waveform"](#) on page 270
- ["To change the waveform reference level based on a user-definable signal section"](#) on page 270

To access the "Waveform/Data View" dialog

Perform one of the following:

1. Select "Repository Tree > Waveform" and:
 - a) Select an imported waveform, e.g. "WV_Matlab".
See ["To import a file with custom I/Q data"](#) on page 233.
 - b) Select "Waveform > View".
2. Select "Repository Tree > Scenario" and select for example "SimplePulseTrain". We assume, that the scenario is configured, a signal generator is selected and a sequence is assigned to this scenario.

See:

- [Chapter 6.3, "How to Select and Create a Test Scenario"](#), on page 68.

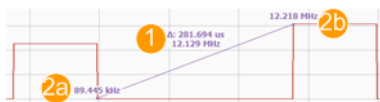
- [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124.
 - a) In the "Scenario" dialog, select "Start".
The software calculates the waveform. Green LEDs indicate that the processing is completed.
 - b) Select "Volatile > View".

See [Figure 16-1](#).

The "Waveform/Data View" displays the I and Q data, the spectrum, and the constellation diagram of the signal, as well as evaluated information from the waveform file, such as the clock rate, number of samples or the duration.

To adjust the display and retrieve more information on the waveform

1. Select the signal characteristic to be visualized, e.g. "View Mode > I/Q".
2. Set the cursor on a particular point to retrieve detailed information.
A single cursor is indicated by a blue line. It sets the center point for all zoom operations.
3. Use the mouse wheel or the "Zoom In/Out" functions to display a part of the waveform or the whole generate waveform.
4. To retrieve delta information:
 - a) Select the "[x]" icon and Drag&Drop the cursors.
The delta cursors are indicated by brackets. The delta information is displayed in the "Delta" section of the dialog.
 - b) Drag the mouse cursor on the diagram.
The delta information is displayed temporary above the line.



- 1 = Delta information: upper value is the Δx [in the selected "Units"], lower value is the Δy [units depending on the selected "View mode"]
 2a, 2b = Start and end y values [units depending on the selected "View mode"]

5. Select "Spectrogram" or "Spectrum".
6. To change the color scheme of the spectrogram, open the context menu and select e.g. "Green".
7. If you observe a part of the waveform, select the play icon to play back the waveform and observe the signal changes over time.

To change the waveform reference level based on a user-definable signal section

1. Select "Repository Tree > Waveform".
2. Select an imported waveform, e.g. "WV_Matlab".

The "Waveform" dialog displays information on the waveform. In this example, "Ref. Level = 14.47 dBFs"

Samples	56 k
Clock	100 MHz
Duration	560 us
Ref. Level	14.47 dBFs

3. Select "Waveform > Set Ref. Level".

The "Waveform Reference Level" dialog opens, see the example on [Figure 16-2](#). It displays the reference signal level of the complete signal. In this example, "Ref. Power Level = 14.47 dBFs".

4. Set the cursor on a particular point.

A single cursor is indicated by a blue line. It sets the center point for all zoom operations.

5. Use the mouse wheel or the "Zoom In/Out" functions to display a part of the waveform or the whole waveform.

6. To measure the reference signal of a particular signal part:

- a)
 - Set the cursor on the start point of the waveform interval to be measured or
 - select "Cursor 1 > Time" and enter the beginning of the evaluation period.
- b) Select "Cursor 1 > Set".
- c) Perform the same for the second cursor.

The two cursors are indicated by brackets. They indicate and define the beginning and end point for the reference level evaluation period.

7. Select "Measure Reference > Start".

Evaluated are all samples *in the defined period*, see the example on [Figure 16-2](#).

The calculated reference signal level is displayed. The software stores this value and uses it as a reference level.

In this example, the calculated value is "Ref. Power Level = 13.9 dBFs".

Tip: You can query the reference level with the command `WAVEform:LEVel:REFerence`.

8. Close the "Waveform Reference Level" dialog and observe the information in the "Waveform" dialog.

Samples	56 k
Clock	100 MHz
Duration	560 us
Ref. Level	13.90 dBFs

The dialog confirms the calculated value ("Ref. Level = 13.9 dBFs").

17 Playing the Generated Waveform Files

Complex pulse signals created with the R&S Pulse Sequencer can be played by:

- R&S SMW
- R&S SMBV

Depending on the installed option and the capabilities of the vector signal generator, files and waveforms are processed as an ARB file or in real time.

Waveform types, waveform handling, and impact of the instrument's hardware on the calculated waveform

The R&S Pulse Sequencer determines the kind of waveform the signal generator instrument is able to handle. Supported are the following waveform types:

- **"Pure" ARB file**
(R&S SMW/SMBV/SGT-K300/-K301)
The R&S Pulse Sequencer calculates a *long waveform without sequencing*; the calculation is performed offline and in advance.
- **Multi-Segment Waveform (MSW)**
(R&S SMW/SMBV/SGT-K300/-K301)
The R&S Pulse Sequencer *pre-processes the sequence and splits it up into sequential* elements. These elements are then played using loops.

The examples in the following sections explain the principle and use R&S SMW to play the waveforms:

- [How to Create, Transfer and Play the Waveforms of Simple Scenarios](#)
- [How to Play the Waveforms of Emitter-Based 2D Scenarios](#)

17.1 Waveform Generation Settings

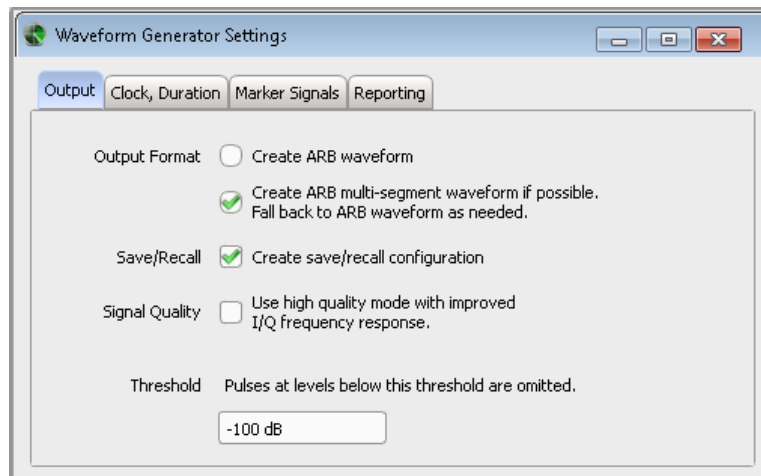
In the general case, the software calculates a waveform, most suitable for the current signal generator; e.g. an ARB file or a multi-segment waveform or a waveform to be played in real time.

Access:

- ▶ In the "Scenario" dialog, select "Waveform Generation > Config".

Settings:

Output	273
Clock, Duration	274
Marker Signals	275
Reporting	275

Output**"Output Format"**

Defines the type of the generated waveform file:

- A single ARB file
- An ARB multi-segment waveform file

"Save/Recall"

Creates the `PS_Recall\ directory on the generator's hard drive and stores the current signal generator configuration as a save/recall file in it.`

The save/recall file has the extension `*.savrc1.txt` and its name reflects the name of the current scenario.

"Use High Quality mode"

Sets the I/Q modulator of signal generator to work in a high quality mode, that is sets the parameter "I/Q Modulator > Mode > High Quality".

For more information, see the user manual or the embedded help of signal generator.

"Threshold"

Enters the threshold. Pulses at levels below this threshold are omitted.

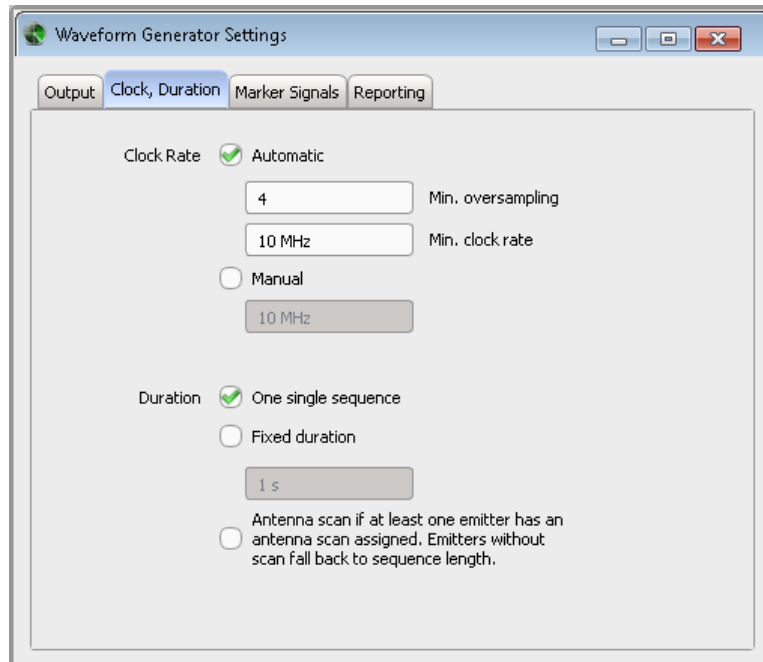
Remote command:

[SCENario:OUTPut:FORMat](#) on page 442

[SCENario:OUTPut:RECall:ENABle](#) on page 443

[SCENario:OUTPut:INSTrument:QHIG:ENABle](#) on page 443

[SCENario:OUTPut:THReshold](#) on page 445

Clock, Duration

Provides clock-related settings:

"Clock Rate" Retrieves the clock rate automatically from the generated waveform or sets a user-defined value. The clock rate influences the size of the generated waveform. To reduce the size of long waveforms, change the clock rate.

"Duration" Defines the waveform content. The generated signal may:

- Contain one single sequence; short sequences result in waveform files with small size
- Last a specified duration; short sequences are repeated, longer ones are truncated
- Contain the signal describing one emitter antenna turn; short sequences are repeated, longer ones are truncated.

Remote command:

[SCENario:OUTPut:CLOCK:MODE](#) on page 441

[SCENario:OUTPut:CLOCK:AUTO:BORDER](#) on page 442

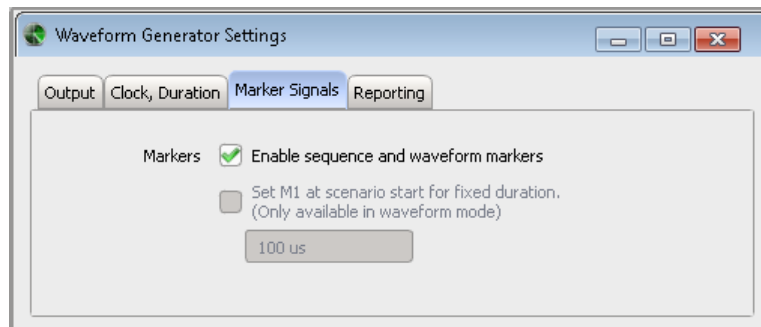
[SCENario:OUTPut:CLOCK:AUTO:OVERsampling](#) on page 441

[SCENario:OUTPut:CLOCK:USER](#) on page 442

[SCENario:OUTPut:DURation:MODE](#) on page 442

[SCENario:OUTPut:DURation:TIME](#) on page 442

Marker Signals



Enables that the configured markers are considered in the generated waveform file.

See ["Global Marker Signals"](#) on page 303.

You can also enable an additional marker, that is held high from the scenario start until a selected duration. This marker signal is output as marker 1. If other marker signals are defined, all markers are summed.

Remote command:

[SCENario:OUTPut:MARKer:ENABle](#) on page 389

[SCENario:OUTPut:MARKer:SCENario:ENABle](#) on page 389

[SCENario:OUTPut:MARKer:SCENario:DURation](#) on page 389

Reporting

Enables and configures the report files generation, see [Chapter 21, "Creating Reports and Documenting Measurement Results"](#), on page 308.

17.2 Assign Signal Source or Receiver Signals to Generator Profile Settings

In "Localized Emitters" and in "Direction Finding" scenarios, you have to select the signal of which emitter/interferer or receiver is transmitted to and played by the connected instrument.

You can assign more than one signal to each of the instrument paths.

See for example:

- [Chapter 15.5, "How to Create Scenarios with Emitters, Interferers and a Receiver"](#), on page 258
- ["To create, load and play the waveforms of a 2D scenario automatically"](#) on page 287

Access:

- ▶ Perform one of the following:
 - In the 2D map of a "Scenario Type = Localized Emitters", select "Assign Emitters".
 - In the 2D map of a "Scenario Type = Direction Finding", select "Assign Receiver Signals".

Assign Signal Source or Receiver Signals to Generator Profile Settings

- In the "Scenario" dialog, select "Upload to VSG/Local ARB File > context menu > Assign Emitters".

The name of the dialog changes depending on the scenario but the settings and the workflow are similar.

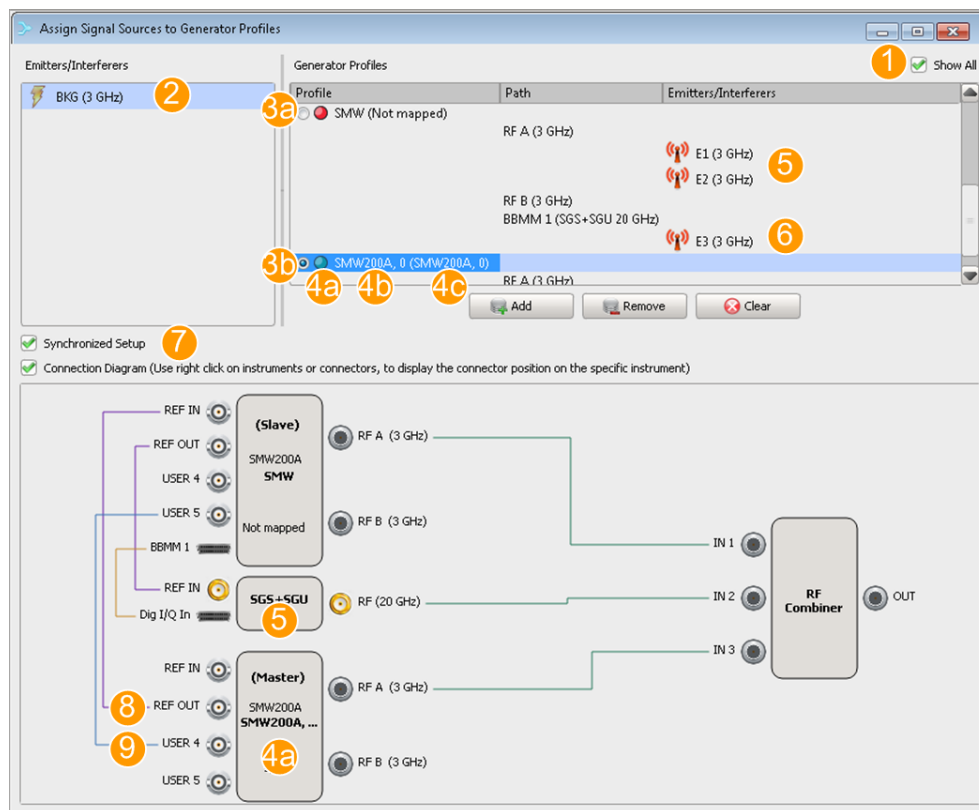


Figure 17-1: Assign Signal Sources to Generator Profiles: understanding the displayed information

- 1 = "Show All"; listed are all available generator profiles, also the unused, or not linked
- 2 = Unassigned Emitter/Interferer
- 3a = Not linked profile
- 3b = Profile mapped to a physical instrument
- 4a = Master instrument
- 4b = Generator profile (see Chapter 14.2, "Generator Profile Settings", on page 240)
- 4c = Physical instrument and serial number
- 5 = Signals of two emitters, assigned to the same RF path
- 6 = Signal assigned to the external RF, connected to and controlled from the R&S SMW
- 7 = Synchronized or phase coherent setup
- 8 = Reference frequency signal of the master instrument, fed in to the slave instrument
- 9 = Trigger signal of the master instrument, fed in to the slave instrument

Settings:

Emitters/Interferers.....277

Receiver Signals..... 277

Show All..... 277

Generator Profiles.....277

Add, Remove, Clear.....278

Connection diagram.....278

Show Connector.....	279
Synchronized Setup.....	279
Phase Coherent Setup.....	279

Emitters/Interferers

Lists the alias names of all configured emitters and interferers, that are still not assigned to a generator.

Remote command:

`ASSignment:EMITters:LIST?` on page 363

`ASSignment:EMITters:SElect` on page 334

Receiver Signals

Lists all receiver signals, that are still not assigned to a generator.

There is one receivers signal per receiver antenna element and emitter combination. The name of the receiver signals follows the syntax:

"<Antenna Element Alias> - <Emitter Name Alias>", as selected with the parameters:

- [Antenna Element Name](#)
- [Emitter/Interferer Alias Name](#)

Remote command:

`ASSignment:ANTennas:LIST?` on page 363

`ASSignment:ANTennas:SElect` on page 334

Show All

Filter the displayed generator profiles:

"On" Shows all available generator profiles.

"Off" Hides the unused or unmapped generator profiles.

Generator Profiles

With "Show All > On", lists the available generator profiles with their paths, supported maximum RF per path and assigned emitters/interferers.

Double click on a profile name accesses the [Generator Profile Settings](#).

The color LED are status indication of the mapping status:

- **Green** - a physical instrument is mapped to the profile.
The signal generator is indicated with its name and serial number, as displayed in the "Instruments Config" dialog.
- **Red** - a physical instrument is mapped to the profile but the instrument is unavailable.
Compare the instrument status in the "Instruments Config > [List of connected instruments](#).
- **Red (Not Linked)** - no physical instrument is linked to the profile.
You can assign signals to not linked profiles, if you prefer to store the calculated waveform in ARB files ("Upload to VSG/Local ARB File > Target > File").

Test setups with more than one signal generators require a master instrument. Master instrument is the signal generator that is the source of reference frequency. Select the generator profile of the master instrument according to your particular setup.

Remote command:

[ASSignment:GENerator:LIST?](#) on page 364

[ASSignment:GENerator:SElect](#) on page 334

[ASSignment:GENerator:CAPabilities?](#) on page 364

[ASSignment:GENerator:MASTer](#) on page 364

[ASSignment:GENerator:PATH:LIST?](#) on page 365

[ASSignment:GENerator:PATH:SElect](#) on page 334

[ASSignment:GENerator:PATH:EMITter:LIST?](#) on page 365

[ASSignment:GENerator:PATH:ANTenna:LIST?](#) on page 365

Add, Remove, Clear

Adds (or removes) a selected emitter/interferer to the selected RF path, incl. connected external RF instruments (R&S SGT, R&S SGS also together with R&S SGU).

Alternatively, use the Drag&Drop method.

Note:

You can assign several signals to the same RF path, as long as the RF frequency and the bandwidth of the resulting signal do not exceed the supported RF frequency and Baseband bandwidth of the physical instrument.

For details, see description R&S®SMW200A User Manual.

Remote command:

[ASSignment:GENerator:PATH:EMITter:ADD](#) on page 332

[ASSignment:GENerator:PATH:EMITter:SElect](#) on page 334

[ASSignment:GENerator:PATH:EMITter:DElete](#) on page 334

[ASSignment:GENerator:PATH:EMITter:CLEar](#) on page 335

[ASSignment:GENerator:PATH:ANTenna:ADD](#) on page 332

[ASSignment:GENerator:PATH:ANTenna:SElect](#) on page 334

[ASSignment:GENerator:PATH:ANTenna:DElete](#) on page 334

[ASSignment:GENerator:PATH:ANTenna:CLEar](#) on page 335

Connection diagram


Visualizes the required physical connections between the signal generators and the DUT.

In test setups with more than one signal generators, the keyword "(Master)" indicates the signal generator that acts as a master device.

See also [Chapter 17.6, "How to Play the Waveforms of Emitter-Based 2D Scenarios"](#), on page 287.

The diagram is interactive:

- A mouse over displays additional information and hints, like max. RF, and assigned signal.
- A double click on an instrument works like "Edit" and accesses the "Generator Profile" dialog.
- Right-click to access the context menu.

Tip: Use the split window icon  to detach the diagram and open it in a separate window.

Show Connector

Accesses the "Find Connector" dialog that displays the physical location of the required connector on the front/rear panel of the instrument.

Synchronized Setup

Some test scenarios require that all signal generators in the test setup generate synchronous signals. The signal generators use a *common reference frequency signal and a trigger signal*.

Note:

If a synchronized setup is required:

- Provide all instruments with the reference frequency and the trigger signal of the master (*Master*) signal generator.
(See [Figure 17-1](#).)
- Set "*Synchronized Setup > On*".

"On" The connection diagram indicates the required connections.
The required settings are set automatically. The software sends SCPI commandos to the master and to the slave instruments, that adjust the reference frequency settings, connector direction, and the mapped signals.

"Off" All signals are generated individually.
Signal delays are not considered and not compensated.

Remote command:

[SCENario:LOCalized:SYNChronize:ENABle](#) on page 440

Phase Coherent Setup

Direction finding scenarios usually require that all signal generators in the test setup generate phase coherent signals. The signal generators use a *common local oscillator frequency signal*.

Note:

If phase coherent signals are required:

- Distribute the local oscillator signal of the master (*Master*) signal generator to all instruments in the setup.
Connect the instruments in a daisy chain.
- In setup with more than one R&S SMW generators, distribute also the reference frequency and the trigger signal.
- Set "*Phase Coherent Setup > On*".

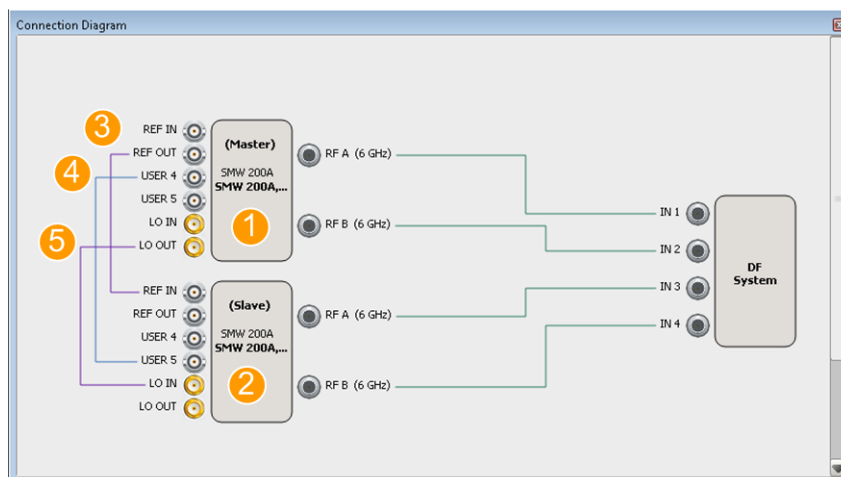


Figure 17-2: Connection Diagram if "Phase Coherent Setup > On": understanding the displayed information

- 1 = Master instrument
- 2 = Slave instrument
- 3 = Reference frequency signal of the master instrument, fed in to the slave instrument
- 4 = Trigger signal of the master instrument, fed in to the slave instrument
- 5 = Local oscillator frequency, fed in to the slave instrument

- "On" The connection diagram indicates the required connections. The required settings are set automatically. The software sends SCPI commandos to the master and to the slave instruments, that adjust the local oscillator frequency settings, connector direction, and the mapped signals.
- "Off" All signals are generated individually. Signal delays are not considered and not compensated.

Remote command:

[SCENario:DIRection:SYNChronize:ENABLE](#) on page 440

17.3 Signal Generator Remote Control Settings

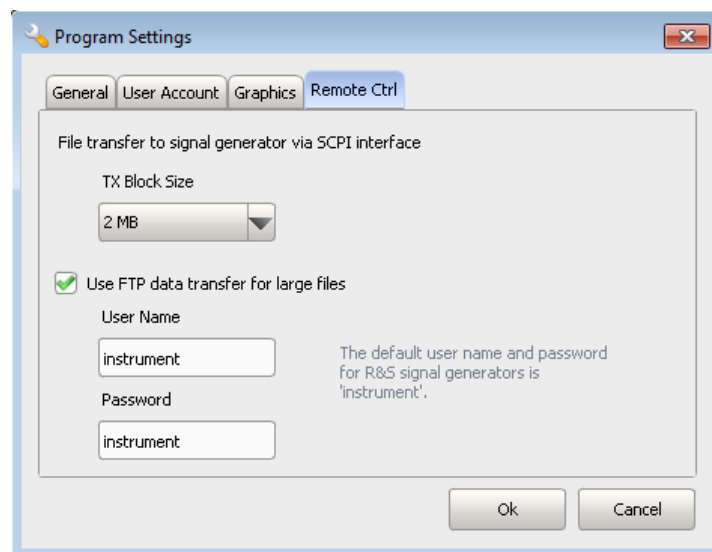
Waveform files generated with the software are automatically uploaded to the connected signal generator in one of the following alternative ways:

- Via the SCPI interface
- Via FTP transfer
If enabled, this second option is automatically applied when the calculated files are large.

Both methods require that the signal generator and the PC on that the software is installed are connected to the same LAN.

Access:

1. In the "Menu bar", select "Configure > Settings > Remote Control".



2. If required, change the size of the transport blocks transmitted via SCPI interface.
3. To use the FTP data transfer:
 - a) *In the signal generator*, enable the FTP interface.
You find the required settings in the "Setup > Security" dialog.
For more information, see the signal generator's embedded help or refer to its user manual (see [Chapter G, "References"](#), on page 524).
 - b) *In the R&S Pulse Sequencer*, enable an FTP data transfer to speed up the data upload to the connected signal generator.
The FTP transmission requires "User Name" and "Password". The default "User Name" and "Password" is *instrument*.

Note: The R&S Pulse Sequencer attempts to connect to *any* connected signal generator with the *same credentials* (i.e. "User Name" and "Password").

Depending on the file size, the software uses either the SCPI interface or the FTP data transfer.

17.4 Instrument Control Panel Settings

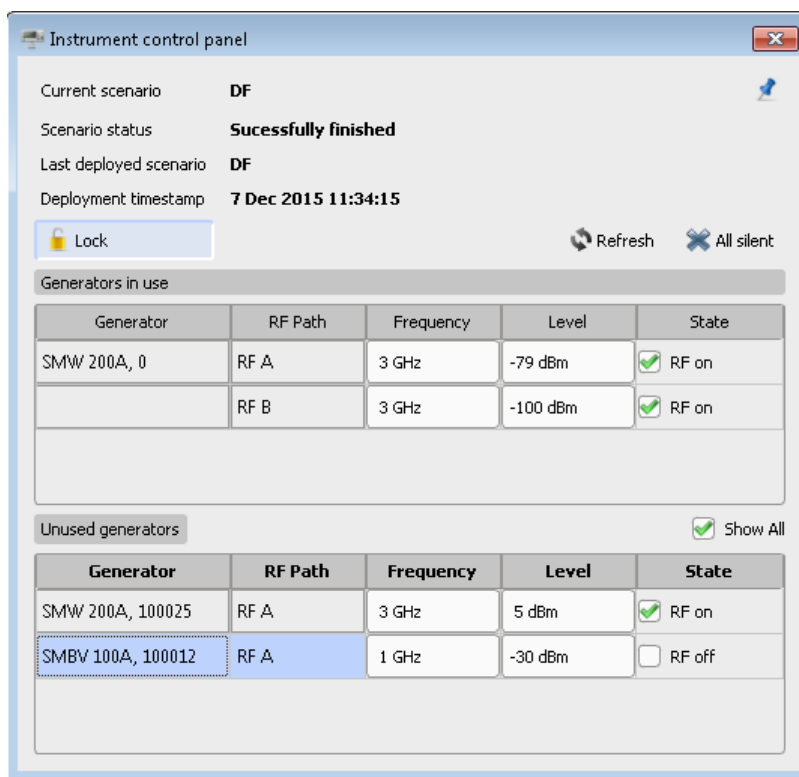
The "Instrument Control Panel" is a dashboard window. It displays key status information on all physical instruments that are configured in the R&S Pulse Sequencer.

From the "Instrument Control Panel" dialog, you can change the level and frequency settings directly in these instruments. No need to install extra software, write dedicated remote control programs or change the settings manually.

This dialog is especially useful, if your test setup consists of several R&S SGT instruments.

Access:

- In the menu bar, select the "Window > Dashboard".



Current scenario

Indicates the name of the current used scenario.

Remote command:

[DASHBOARD:SCENARIO:CURRENT:NAME?](#) on page 357

Scenario status

Status information, like "Calculating", "Deployed", "Successfully finished" or "Failed".

"Unknown" indicates that there is no loaded or calculated scenario, for example right after starting up the R&S Pulse Sequencer.

Remote command:

[DASHBOARD:SCENARIO:CURRENT:STATUS?](#) on page 357

Last deployed scenario

Indicates the name of the last deployed scenario. That is, the scenario that has been calculated and stored (locally or on the hard drive of the instrument) last.

Remote command:

[DASHBOARD:SCENARIO:DEPLOYED:NAME?](#) on page 357

Deployment timestamp

Indicates the date and time the scenario has been deployed.

"Unknown" indicates that there is no loaded or calculated scenario, for example right after starting up the R&S Pulse Sequencer.

Remote command:

[DASHBOARD:SCENARIO:DEPLOYED:TIME?](#) on page 358

Pin/Unpin

Displays the window always on the top of all other windows.

Unlock/Lock

Unlocks the RF frequency, level, and output state settings for editing.

Refresh

Updates the displayed information.

Remote command:

[DASHBOARD:REFRESH](#) on page 357

Mute All

Deactivates the RF outputs of all signal generators.

Remote command:

[DASHBOARD:MUTE](#) on page 357

Show All

If enabled, the table lists all signal generators that are configured in the Profile to Instrument Links dialog.

"Off" Listed are only the signal generators to that a generator profile is assigned.

"On" Listed are all available signal generators. That is, the generators listed in the Profile to Instrument Links dialog

Remote command:

[DASHBOARD:UNUSED:ALL](#) on page 358

Generators in use, Unused generators

An overview of assigned (used) and the available but not used (unused) signal generators with their main characteristics:

"Generator" Instrument name as listed in the list of instruments in the Profile to Instrument Links dialog.

Remote command:

[DASHBOARD:USED:LIST?](#) on page 358

[DASHBOARD:USED:SELECT](#) on page 334

[DASHBOARD:UNUSED:LIST?](#) on page 358

[DASHBOARD:UNUSED:SELECT](#) on page 334

"RF path, Frequency, Level, State"

Available RF paths, current RF frequency, level and output state, as set for the last deployed scenario.

Indicated are also the RF frequency, level, and output states of the RF extensions. RF extensions is a term describing the instruments connected to the R&S SMW, like R&S SGT.

To edit the values and the states, select "Unlock".

Remote command:

DASHBOARD:USED:PATH:LIST? on page 358

DASHBOARD:UNUSED:PATH:LIST on page 358

DASHBOARD:USED:PATH:SELEct on page 334

DASHBOARD:UNUSED:PATH:SELEct on page 334

DASHBOARD:USED:FREQuency on page 359

DASHBOARD:UNUSED:FREQuency on page 359

DASHBOARD:USED:LEVEl on page 359

DASHBOARD:UNUSED:LEVEl on page 359

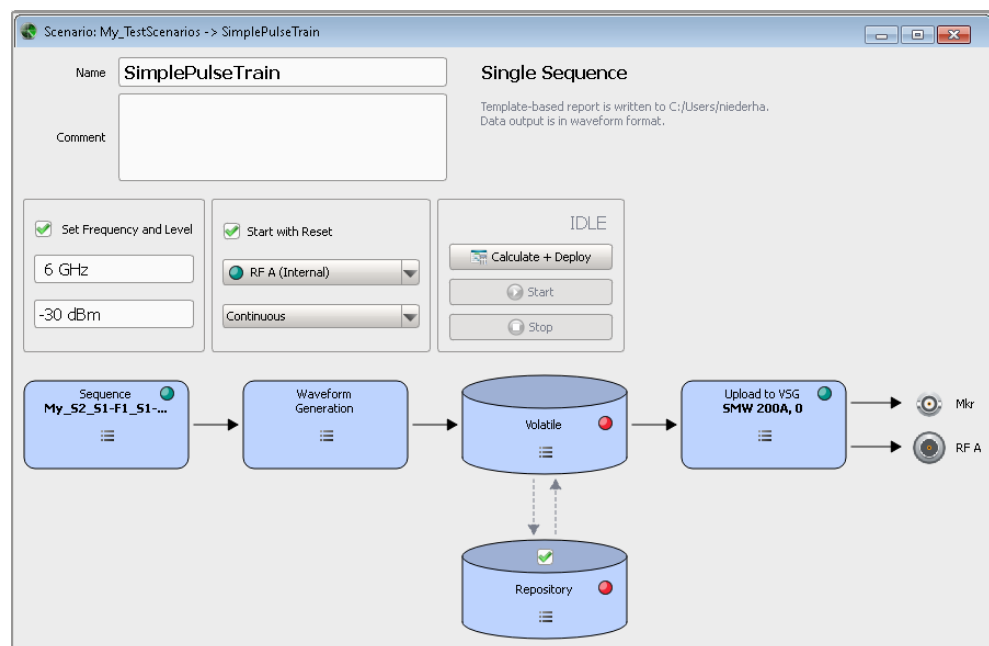
DASHBOARD:USED:STATe on page 359

DASHBOARD:UNUSED:STATe on page 359

17.5 How to Create, Transfer and Play the Waveforms of Simple Scenarios

This example uses the test setup as shown on [Figure 4-1](#)

1. Open a suitable scenario.
2. Configure the settings as follows:



See:

- [Chapter 8.3, "How to Create Sequences and Use the Control Elements"](#), on page 124
- [Chapter 14.3, "How to Create and Map Generator Profiles"](#), on page 244

3. Use the default "Waveform Generator" settings.

See also [Chapter 17.1, "Waveform Generation Settings"](#), on page 272.

All prerequisites are fulfilled. The "Calculate + Deploy" button is enabled.

4. Select "Calculate + Deploy".

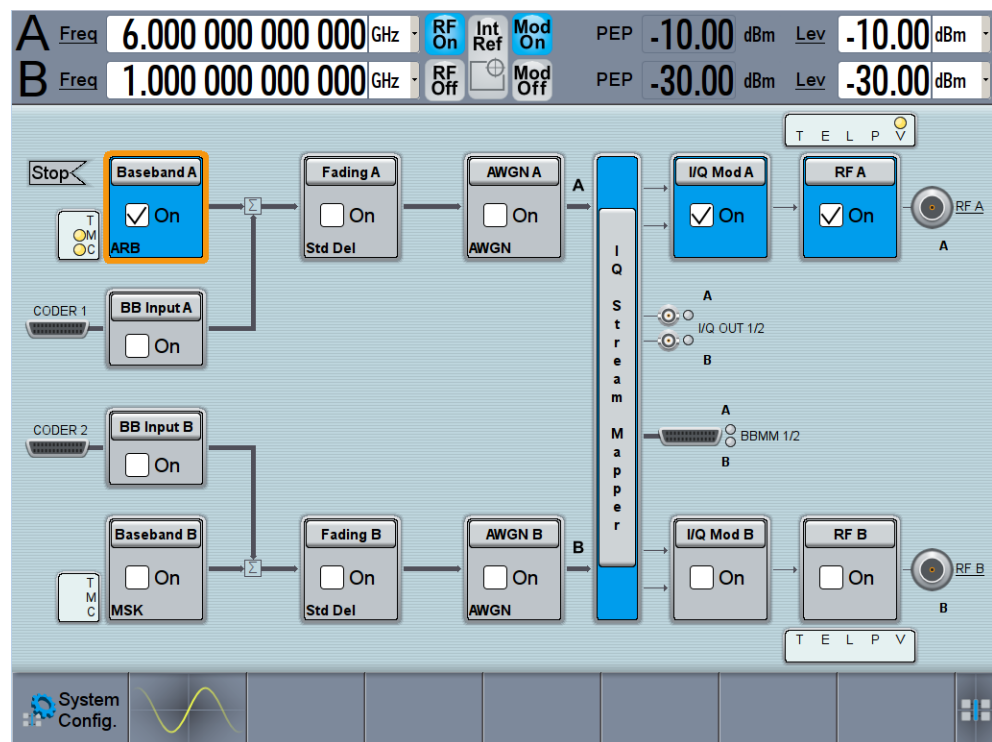
The software calculates the waveform and stores it in the "Volatile" memory.

Progress information indicates the calculation progress.

The waveform is automatically transferred to the instrument.

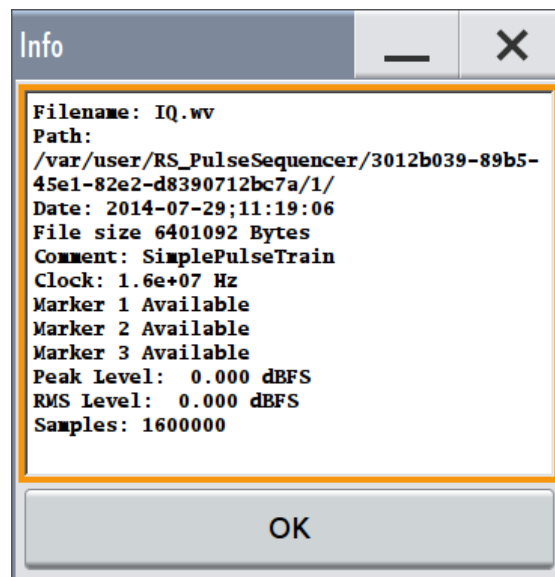
5. Select "Start".

- The waveform is automatically loaded in the "ARB".
- The output frequency and level of the signal generator are set accordingly.
- Signal generation is started and the signal is output at the RF A connector.



In the signal generator, perform further configurations, e.g.:

- Select "Baseband > ARB > General > Waveform Info" to confirm that the waveform generated with the R&S Pulse Sequencer. You retrieve also information on the waveform file name, used clock rate, storage location, etc.



- The generated signal contains marker signals. Use the default routing or change the mapping of the marker signals to the output connectors.

For details, see description R&S®SMW200A User Manual.

In the R&S Pulse Sequencer:

- Select "Volatile > View" to visualize the generated waveform. The display resembles the one on [Figure 8-9](#).



Playing the waveforms of 2D scenarios

If your current scenario type is "Localized Emitters", you also have to:

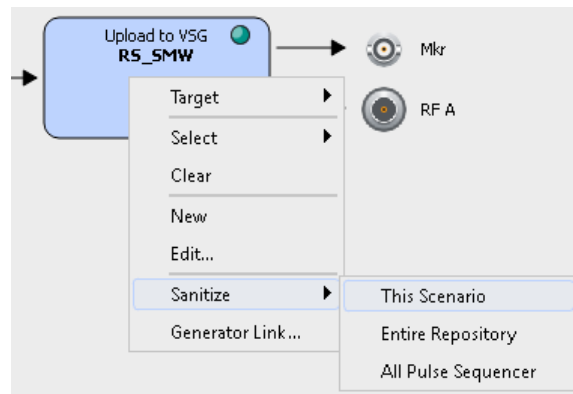
- Select the signal of which emitter/interferer is transmitted to the connected signal generator
- In a test setup with two or more signal generators:
 - Select "Connection diagram > On"
 - Define the master device
 - Connect the instruments as illustrated on the diagram

For detailed description, see [Chapter 17.6, "How to Play the Waveforms of Emitter-Based 2D Scenarios"](#), on page 287.

To delete waveforms form the hard disk of the signal generator

If the waveforms uploaded on the vector signal generator are no longer required, they can be deleted automatically form the R&S Pulse Sequencer:

1. Open the "Scenario" dialog.
2. In the context menu of the "Upload to VSG" block, select "Sanitize" and select e.g. "This Scenario".



The waveforms belonging to the current scenario are deleted from the hard disk of the signal generator.

17.6 How to Play the Waveforms of Emitter-Based 2D Scenarios

To create, load and play the waveforms of a 2D scenario automatically

1. Open a test scenario with "Scenario Type > Localized Emitters".
2. Select "Upload to VSG > Target > Generator".
3. In the "Scenario" dialog, select "Upload to VSG > Assign Emitters".
The "Assign Signal Source to Generator Profiles" dialog opens.
4. Drag&Drop an Emitter/Interferer to assign its signal to a generator profile.
Tip: You can assign several signals to the same RF path, as long as the RF frequency and the bandwidth of the resulting signal do not exceed the supported RF frequency and Baseband bandwidth of the physical instrument.
See the example on [Chapter 17.2, "Assign Signal Source or Receiver Signals to Generator Profile Settings"](#), on page 275.
5. Enable "Connection diagram > On".
The diagram visualizes the required connections.
The keyword **(Master)** indicates the signal generator that acts as a master device.
6. In "Generator Profile" section, change the master device according to your particular setup.
7. If a synchronized setup is used:
 - a) Select "Synchronized Setup > On"
 - b) Provide all other instruments with the reference frequency and the trigger signal of the master signal generator.

The R&S Pulse Sequencer sends the required SCPI commands to the instruments. All required signal generator settings are configured automatically.

In this example for instance, the direction and the mapped signal to USER 4 and USER 5 connectors of both signal generators are configured automatically.

8. In the scenario dialog, select "Calculate + Deploy".

The software calculates the waveforms. Progress information indicates the calculation progress.

The waveforms are stored in the volatile memory and transferred to the instruments.

9. Select "Start".

The waveforms are loaded in the "ARB".

The output frequency and level of the signal generators are set accordingly.

Signal generation is started synchronous with a *fixed delay of 1 ms*; the signal is output at the RF connectors.

10. In the block diagram of the current scenario, select "Volatile > View > E1" to visualize the waveform of the emitter "E1".

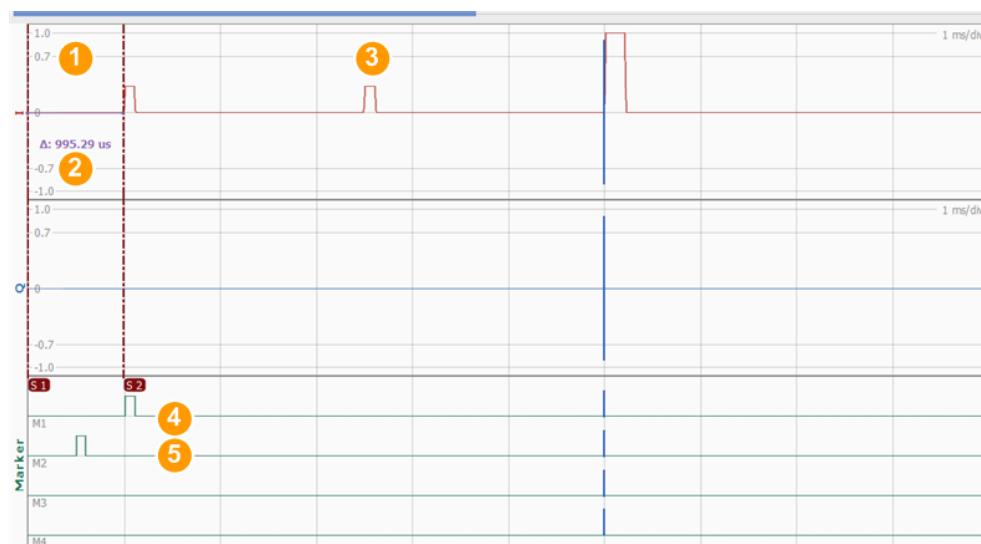


Figure 17-3: Data View: Example of a waveform in a test setup with multiple instruments

- 1 = Automatically appended blank segment
- 2 = Blank segment with fixed duration of 1 ms
- 3 = Second segment
- 4 = Marker M1: the sequence start marker which appears at the first pulse of the sequence; M1 can be used to trigger e.g. an analyzer to the start of the scenario
- 5 = Marker M2: automatically generated marker at the middle of the blank segment; M2 is used for synchronization purposes

The "Data View" shows that a blank segment of 1 ms is appended to the waveform. In the middle of this segment, i.e. after 500 us, the automatically generated marker signal M2 appears (see [Figure 17-3](#)). This marker M2 is used to synchronize test setups with multiple instruments.

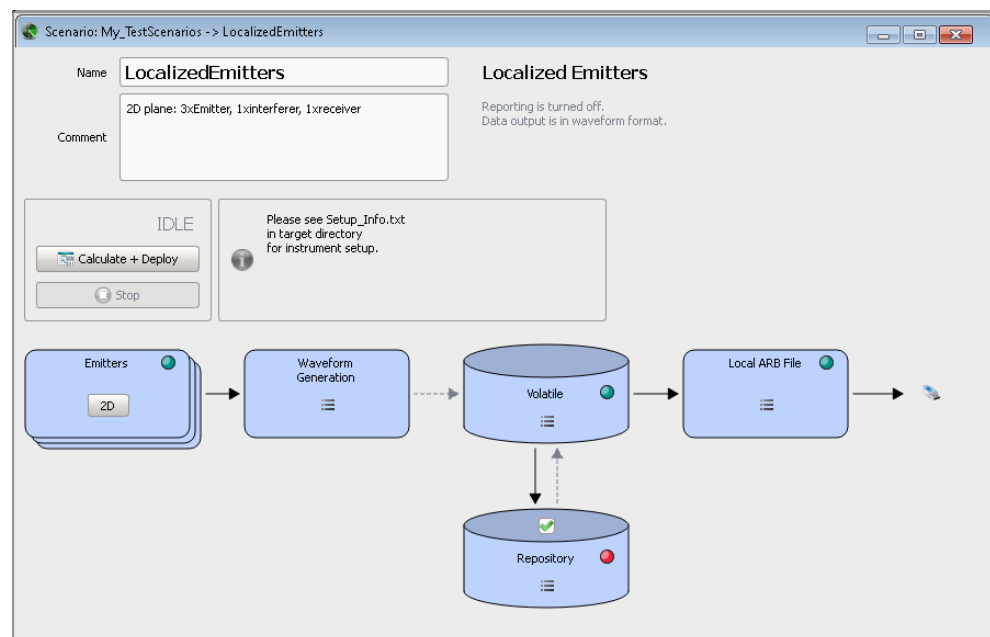
The master instrument, the R&S SMW in this example, is triggered by the R&S Pulse Sequencer. Once the waveform is loaded into the ARB of the master instrument, the blank segment is played first. The marker signal M2 triggers the ARB of the slave instrument, that is the second R&S SMW. The duration of the blank segment and the time interval of 500 us are selected to be longer than the processing time of the slave instrument. An internal procedure guarantees that the period from generating the marker signal M2 in the master instrument until start of the ARB in the slave instrument is exactly 500 us.

Thus, the ARBs of all signal generators in the setup start synchronously.

To create, transfer and play the waveforms of 2D scenario manually

During the initial tests, it is useful to create test waveforms and store them on a local or network drive. These waveforms can be later transferred to and played with a signal generator.

1. Open a test scenario with "Scenario Type > Localized Emitters".
2. Select "Upload to VSG > Target > File".
3. Select "Local ARB File > Set Path" and select the storage location for the generated waveform.
4. Select "Repository > On".
5. Select "Calculate + Deploy".



The software calculates the waveform and stores it in the "Volatile" memory. Progress information indicates the calculation progress. The waveforms are created, automatically named and stored in the selected directory.

6. If your user rights are sufficient, you can store the generated file also in the repository.
Select "Volatile > Release to Repository".
7. Select "Local ARB File > Open Path in Explorer".
In this example, the directory contains the `Setup_Info.txt` file and three waveforms (`My_RS_SMW_PathA_1_E1.wv`, `My_RS_SMW_PathB_3_E2.wv`, and `RS_SMW_PathA_4_E3.wv`), one for each mapped emitter.
8. Open the `Setup_Info.txt` file.

```
Repository: My_TestScenarios
Scenario:   LocalizedEmitters
Creator:    R&S
Date:      Mi 21. Jan 11:24:25 2015
```

Setup table

Generator	Path	waveform File	Frequency	Level	Trigger offset
My_RS_SMW	Path A	My_RS_SMW_PathA_1_E1.wv	6 GHz	-1.1 dBm	11.082 us
My_RS_SMW(*)	Path B	My_RS_SMW_PathB_3_E2.wv	6 GHz	3.54 dBm	0 s
RS_SMW	Path A	RS_SMW_PathA_4_E3.wv	6 GHz	46.64 mdBm	7.769 us

Master Emitter/Interferer in table (marked with asterisk) should trigger all other Emitters/Interferers.
Connect the trigger signals according to the block diagram and configure the trigger modes of the individual basebands manually.
Ensure, that the reference clock of the signal generator with the master Emitter/Interferer is distributed to all other generators

9. Follow the instructions listed in the file.
10. To display the test setup and the connection diagram, select "Emitters > 2D > Assign Emitters" and enable "Connection diagram > On".

18 Defining the Data Source

The R&S Pulse Sequencer provides a wide range of internally defined data sources, such as patterns or PRBS generators. Data sources are available globally within the project. Once created, they can be used in all pulses. Each pulse retrieves its data individually.

Data is retrieved cyclically, starting from the first list entry. Once all bits are used, the data is retrieved from the second one. When the list end is reached, the data is retrieved from the first entry.

If the data entry contains fewer bits than the required, the data bits are repeated cyclically.

Example: Data source processing

Configure the data source as shown on [Figure 18-1](#).

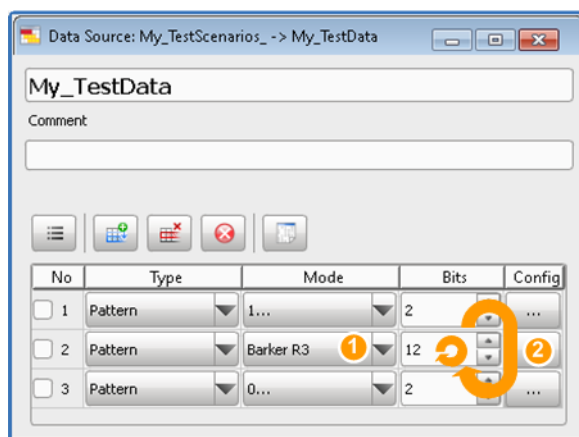
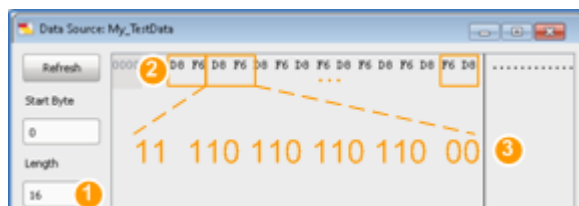


Figure 18-1: Data Source: understanding the displayed information

- 1 = Barker R3 pattern = 110; required are 12 bits, i.e. the pattern is repeated 4 times
- 2 = If more than 16 bits are required, the 16 bits are repeated

Use the [Preview Data Source](#) to visualize the content.



- 1 = Displayed length is 16 bytes
- 2 = Configured 16 bits (2 bytes), repeated 8 times
- 3 = Content of each 2 byte: 2x"Pattern = 1" 4x"Pattern = Barker R3" 2x"Pattern = 0"

Supported types of data source

The R&S Pulse Sequencer provides the following types of data sources:

- *Pattern*
 - Simple data patterns such as binary 0 ("0...") strings or 1 ("1...") strings

- Variable bit strings ("1010") with alternating 0 and 1 and a maximum length of 999 bits
- Different Barker codes: Barker R3, Barker R4a, etc.
- *User*
A user-defined sequence of 0 bits and 1 bits
The following rules apply for user defined data source:
 - Accepted are 0 and 1, and numbers in hexadecimal format
 - Comments can be enclosed in slashes (/)
 - Blank characters are ignored
 - The sequence #x starts a hexadecimal input for the remainder of the line
 - A new line turns hexadecimal input and comments off
 - Ones and zeroes are evaluated as single bits
 - Strings can be enclosed in quotation marks ("")
- *PRBS (Pseudo-Random Binary Sequence)*
The 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, see [Table 18-1](#).

Example:

By way of example, the diagram on [Figure 18-2](#) shows 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.

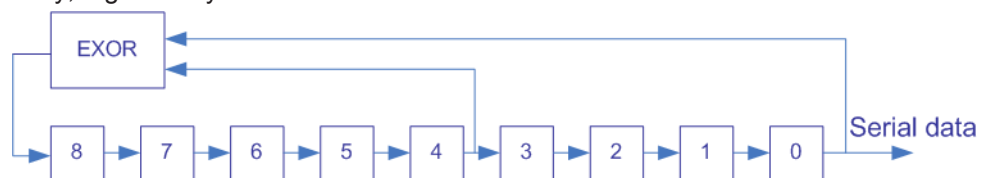


Figure 18-2: A 9-bit PRBS generator

Table 18-1: Overview of PRBS generators

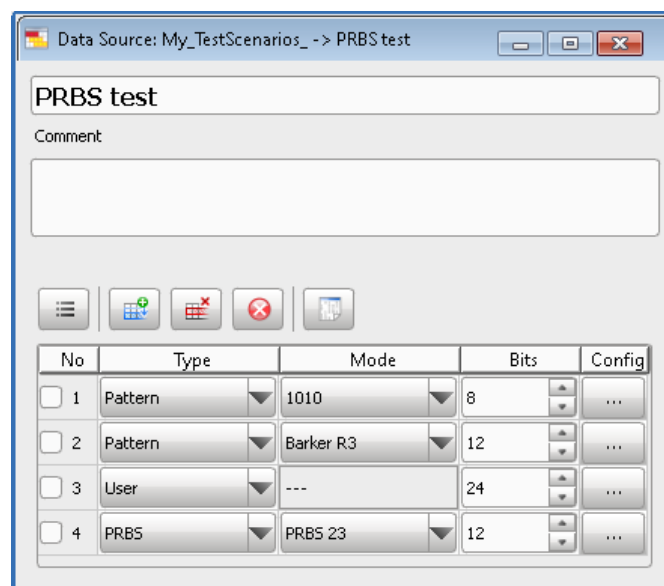
PRBS generator	Length in bits	Feedback to
7-bit	$2^7 - 1 = 127$	Registers 1, 0
9-bit	$2^9 - 1 = 511$	Registers 4, 0
11-bit	$2^{11} - 1 = 2047$	Registers 2, 0
15-bit	$2^{15} - 1 = 32767$	Registers 1, 0
16-bit	$2^{16} - 1 = 65535$	Registers 5, 3, 2, 0
20-bit	$2^{20} - 1 = 1048575$	Registers 3, 0

PRBS generator	Length in bits	Feedback to
21-bit	$2^{21} - 1 = 2097151$	Registers 2, 0
23-bit	$2^{23} - 1 = 8388607$	Registers 5, 0

18.1 Data Sources Settings

Access:

- ▶ Select "Repository Tree > Data Source > New".



Available settings:

Data Source Name.....	293
Comment.....	294
Data Source Table.....	294
Shift register.....	294
User Data.....	295
Preview Data Source.....	295

Data Source Name

Enter a name for the data source.

Remote command:

- [DSRC:NAME](#) on page 330
- [DSRC:CREate](#) on page 329
- [DSRC:CATalog?](#) on page 329
- [DSRC:SElect](#) on page 330
- [DSRC:REMove](#) on page 331
- [DSRC:CLEar](#) on page 335

Comment

Enter a short description.

Remote command:

[DSRC:COMMeNt](#) on page 331

Data Source Table

A table with one or more items that described the content of the data source.

"Item Select" Selects an item for performing standard tasks, like inserting an item above the current one, or deleting an item.

Remote command:

[DSRC:ITEM:SELEct](#) on page 333

[DSRC:ITEM:ADD](#) on page 332

[DSRC:ITEM:DELEte](#) on page 334

"Data Source Type, Data Source Mode"

Sets the type of the data and the data content for the current item, see ["Supported types of data source"](#) on page 291.

See also:

- [Chapter 18.2, "How to Configure the Bit Stream Used by the MOP"](#), on page 295
- [Example "Data source processing"](#) on page 291

Remote command:

[DSRC:ITEM:TYPE](#) on page 362

[DSRC:ITEM:PATTeRn](#) on page 361

[DSRC:ITEM:PRBS:TYPE](#) on page 361

"Bits" Sets the length of the current item in bits.

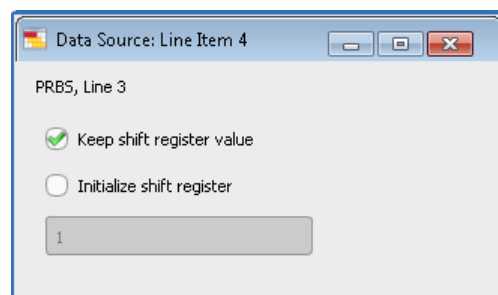
Remote command:

[DSRC:ITEM:BITS](#) on page 360

"Config" Accesses a dialog with more settings, see: ["Shift register"](#) on page 294

Shift register

A PRBS sequence is defined by a length and a register shift number, see [Table 18-1](#).



"Keep shift register value"

Use the default values.

"Initialize shift register"

Customize the shift register.

"Initialization value"

Set a new initialization value.

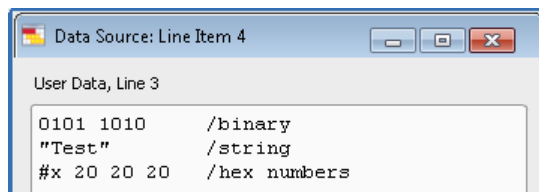
Remote command:

[DSRC:ITEM:PRBS:INIT](#) on page 361

[DSRC:ITEM:PRBS:INIT:VALue](#) on page 361

User Data

For "Data Source > Type > User", you can define your own data pattern, see ["Supported types of data source"](#) on page 291.



Remote command:

[DSRC:ITEM:DATA](#) on page 360

Preview Data Source

Displays the data content.

See also [Example "Data source processing"](#) on page 291.

"Refresh" Perform refresh to display the current data content.

"Start Byte, Length"

Sets the data bits currently displayed.

18.2 How to Configure the Bit Stream Used by the MOP

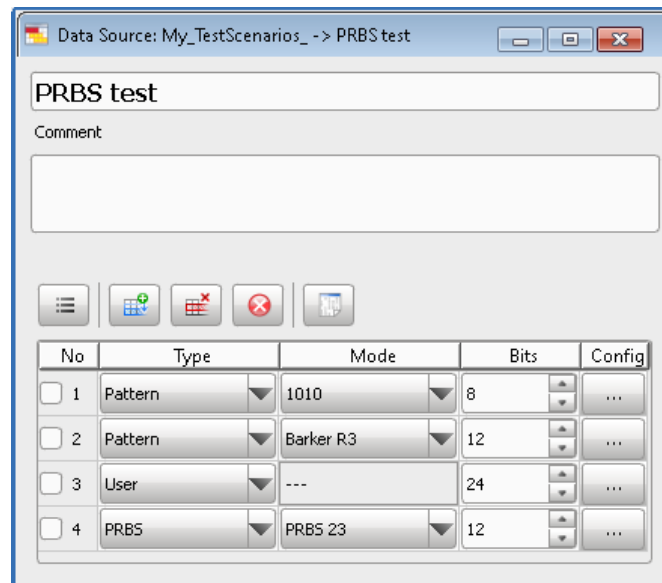
See:

- ["To configure the data source"](#) on page 295
- ["To assign the data source to a MOP"](#) on page 296

To configure the data source

To access the "Data Source" dialog:

1. Select "Repository Tree > Data Source > New".
2. Select the "Append new item" icon.
3. Enter a name and add a comment.
4. For each item, select the data type, the data mode, and the data length in bits.



5. To view the data source content, select the "Preview" icon.

Data is retrieved cyclically, starting from the first list entry, see [Example "Data source processing"](#) on page 291.

To assign the data source to a MOP

1. In the repository tree, select "Pulse > PulseName e.g. My_TestPulse > MOP".
2. Select "MOP > Settings" and select a MOP that requires a data source, e.g. "MOP Type > FSK".
3. Select "Data Source > e.g. My_TestData".

19 Defining Complex Modulation Schemes and IPM Profiles

Custom modulation schemes and custom envelope shapes are defined by an external plugin. Plugins are Microsoft Windows DDL modules that contain the maths that is required for the envelope shaping and the modulation on pulse.

Some example plugins are provided with the software as binary and source code. These examples can serve as a starting point for own applications. Once imported, the software handles the plugins automatically.

Each plugin:

- Have to provide a range of functions to identify itself and perform the calculations required for the modulation on pulse
- Can register a set of configuration parameters that can be used as variables inside the plugin.

You can load plugins to:

- Specify your custom modulation scheme ([MOP](#))
- Define a custom inter-pulse modulation profile (IPM)
- Create a report file during the waveform creation process
- Define the file format of the custom antenna pattern files.

How to import and assign user defined plugins

See:

- [Chapter C, "Plugin Programming API"](#), on page 495 for description of the Plugin programming API and instruction on how to load the plugin examples.
- ["To import a plugin"](#) on page 297
- ["To assign the user defined modulation scheme to a pulse"](#) on page 298
- ["To assign the user defined IPM profile"](#) on page 299

See also:

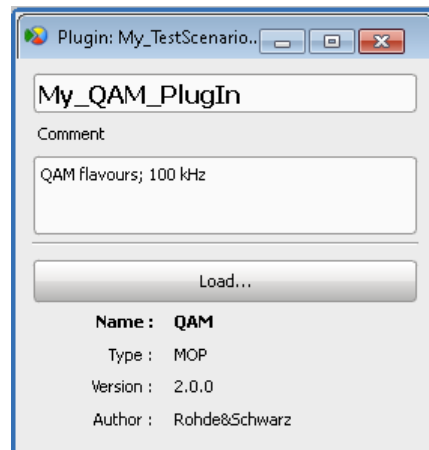
- ["To import an antenna pattern file in custom file format"](#) on page 193
- [Chapter 21, "Creating Reports and Documenting Measurement Results"](#), on page 308

To import a plugin

1. In the repository tree, select "Plugins > New".
2. In the "Plugin" dialog, enter name and a description.
3. Select "Load", navigate to the *.dll file, and select it.

List and loaded are only files containing modulation data.

The software loads the *.dll file, retrieves information from it and displays it in the "Plugin" dialog.



You can use the plugins to create:

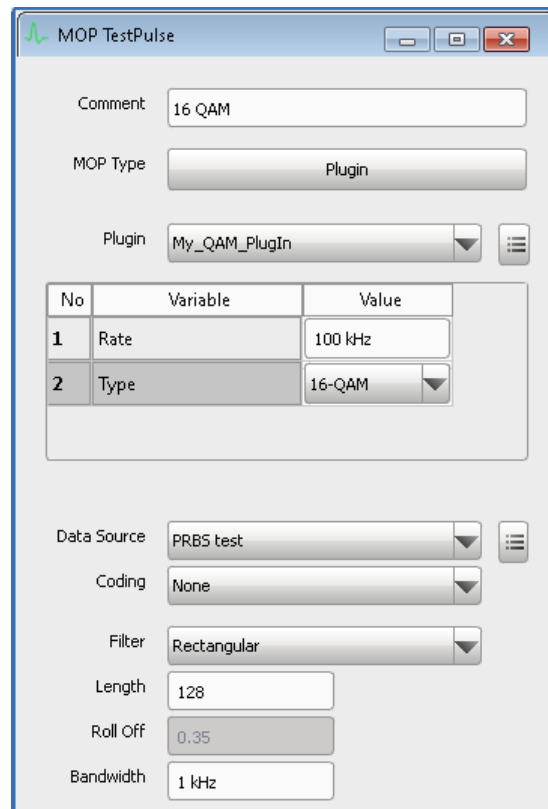
- Create custom modulation schemes, see ["To assign the user defined modulation scheme to a pulse"](#) on page 298.
See also [Chapter 7.2.6.7, "Plugins"](#), on page 99.
- Create custom IPM profile, see ["To assign the user defined IPM profile"](#) on page 299
- Import custom antenna pattern file formats, see ["To import an antenna pattern file in custom file format"](#) on page 193.

Remote Commands:

- [PLUGin:CREate](#) on page 329
- [PLUGin:CATalog?](#) on page 329
- [PLUGin:NAME](#) on page 330
- [PLUGin:COMMeNt](#) on page 331
- [PLUGin:LOAD](#) on page 392
- [PLUGin:SELeCt](#) on page 330
- [PLUGin:REMOve](#) on page 331
- [PLUGin:MODule:NAME?](#) on page 330
- [PLUGin:MODule:TYPE?](#) on page 392
- [PLUGin:MODule:VERSion?](#) on page 392
- [PLUGin:MODule:AUTHor?](#) on page 392
- [PLUGin:MODule:COMMeNt?](#) on page 392
- [PLUGin:MODule:DATA?](#) on page 392

To assign the user defined modulation scheme to a pulse

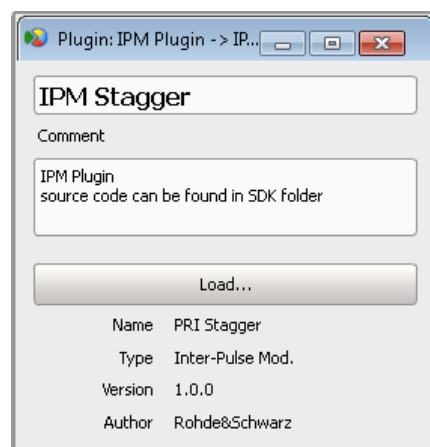
1. In the repository tree, select "Pulse > PulseName > MOP".
2. Select "MOP > Settings".
3. Select "MOP > MOP Type > Plugin".
4. Select "Plugin" and select a loaded user defined modulation scheme, e.g. My_QAM_PlugIn.



5. Select "Pulse > PulseName > MOP > Enable Modulation on Pulse (MOP) > On".
6. Use the [Restrict modulation to a certain area of the pulse](#) settings to determine the exact pulse part the modulation is applied to.

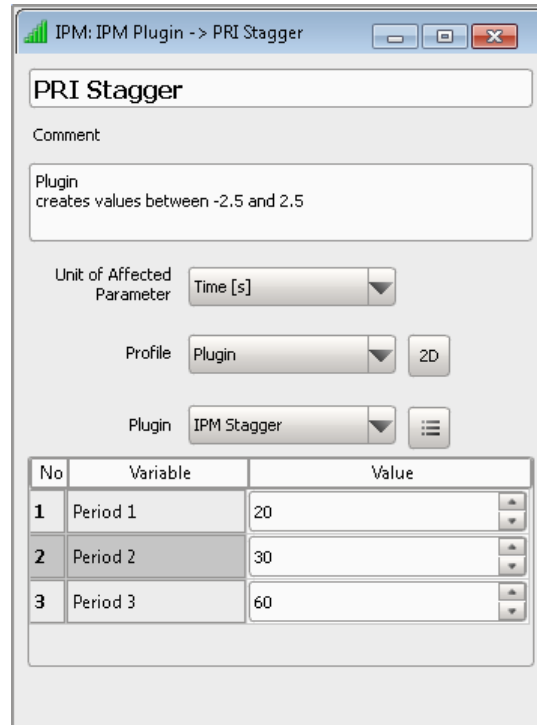
To assign the user defined IPM profile

1. Import a suitable plugin.



2. In the repository tree, select "Inter-Pulse Mods > IPM-Name".
3. Select "Profile > Plugin".

4. Select "Plugin" and select a loaded user defined IPM profile, e.g. IPM Stagger.



5. Select "2D" to visualize the IPM profile.
6. If required, change the values of the used variables and observe the effect on the "2D" diagram.

20 Defining and Enabling Marker Signals

Markers signals or markers are binary signals that are additionally generated and added to the generated signal. Markers are commonly used to synchronize a device under test (DUT) or a spectrum analyzer to the generated signal, for example to the beginning of a pulse. In the vector signal generator, generated marker signals can be routed to the corresponding output connectors.

For details, see description R&S®SMW200A User Manual.

20.1 Marker Settings

In the R&S Pulse Sequencer, you define markers on three levels: on pulse basis, in the sequence or define global conditions like the start of a scenario.

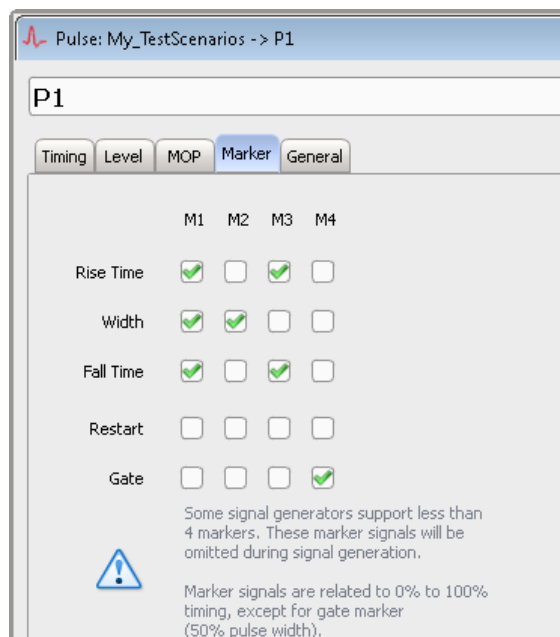
Access:

1. Select "Repository Tree > Pulse > Marker".
See "[Pulse Markers](#)" on page 301.
2. Select "Repository Tree > Sequence > Sequence Description > Marker".
See "[Sequence Markers](#)" on page 302.
3. Select "Repository Tree > Scenario > Waveform Generator > Config".
4. Select "Marker Signals > Enable sequence and waveform markers".
See "[Global Marker Signals](#)" on page 303.

Pulse Markers

With the provided settings, you can assign up to 4 markers to the different pulse phases, for example rise and fall time, pulse width. A restart marker indicates the first 10% of the entire PRI.

Marker information is directly added to the resulting waveform and the marker signal output is therefore synchronous with the waveform playback.



Remote command:

[PULSe:MARKer:RISE](#) on page 387

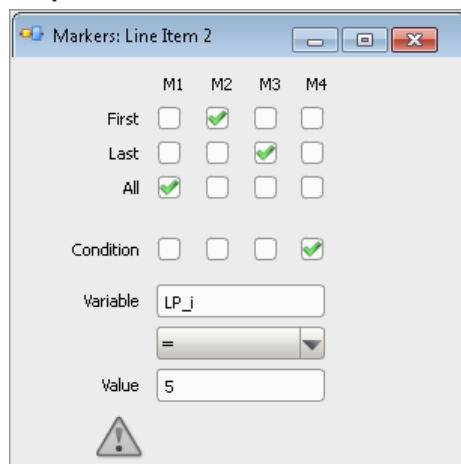
[PULSe:MARKer:WIDTH](#) on page 387

[PULSe:MARKer:FALL](#) on page 387

[PULSe:MARKer:AUTO](#) on page 387

[PULSe:MARKer:GATE](#) on page 387

Sequence Markers



"First, Last, All" If pulse markers are defined, you can also define markers on multiple repeating pulses and mark the first, last or all pulses.

Remote command:

[SEquence:ITEM:MARKer:FIRSt](#) on page 387

[SEquence:ITEM:MARKer:LAST](#) on page 387

[SEquence:ITEM:MARKer:ALL](#) on page 387

"Condition, Variable, Value"

A marker signal is generated if a defined condition is fulfilled. Conditions are defined as logical expressions, where the variable value ("Variable") is compared to a fixed value ("Value"). The variable must exist.

Example:

If a sequence item is included into a loop, you can create marker signals that mark for example one specific loop run number.

See:

- [Chapter 8.2.3, "Loop Settings"](#), on page 119
- ["To enable sequence markers"](#) on page 304

Remote command:

[SEquence:ITEM:MARKer:CONDition](#) on page 387

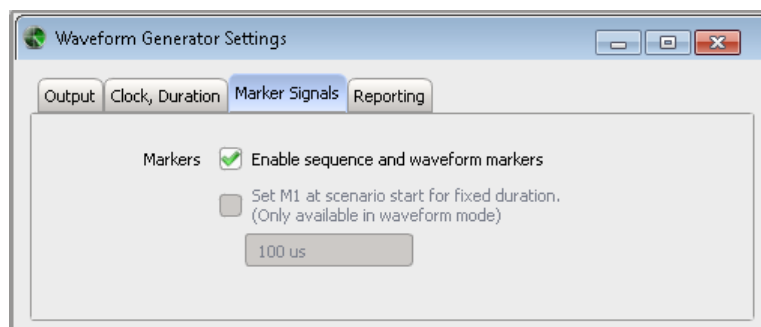
[SEquence:ITEM:MARKer:CONDition:VARiable](#) on page 388

[SEquence:ITEM:MARKer:CONDition:TYPE](#) on page 388

[SEquence:ITEM:MARKer:CONDition:VALue](#) on page 388

Global Marker Signals

In scenarios, if the selected sequence contains markers, you can select that these markers are considered by the generation of the output waveform file.



Remote command:

[SCENario:OUTPut:MARKer:ENABLE](#) on page 389

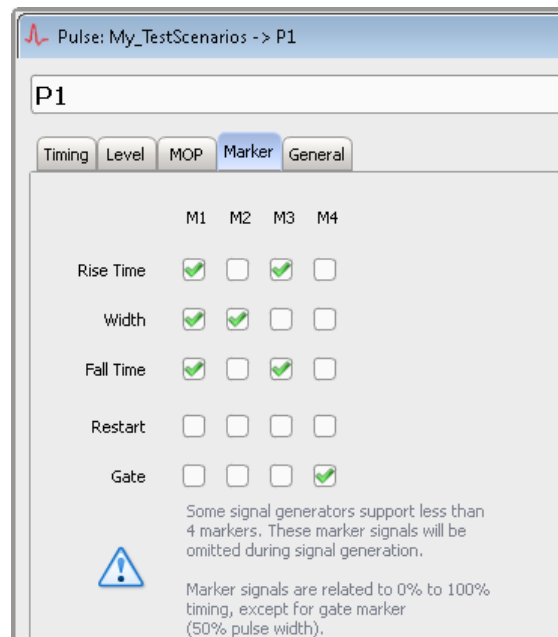
[SCENario:OUTPut:MARKer:SCENario:ENABLE](#) on page 389

[SCENario:OUTPut:MARKer:SCENario:DURation](#) on page 389

20.2 How to Configure and Visualize Markers

To enable pulse markers

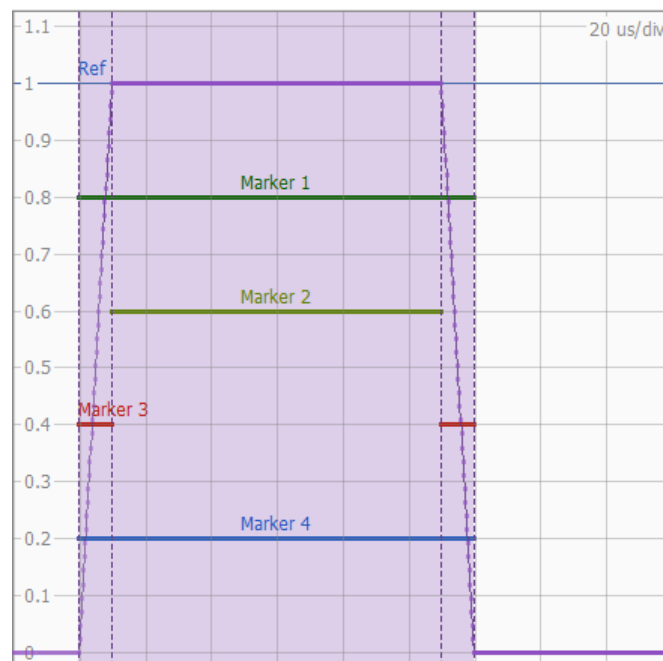
1. Select "Repository Tree > Pulse > Marker".
2. Enable Marker 1 ("M1") for "Rise Time", "Width" and "Fall Time".



The Marker 1 is active for the entire number of samples. That is, the marker is held high for the whole time from beginning of the rising edge to end of the falling edge.

3. Enable "M2 > Width", "M3 > Rise and Fall Time" and "M4 > Gate".

The envelope graph marks the time periods during that the markers are held high.



To enable sequence markers

1. Select "Repository Tree > Sequence".

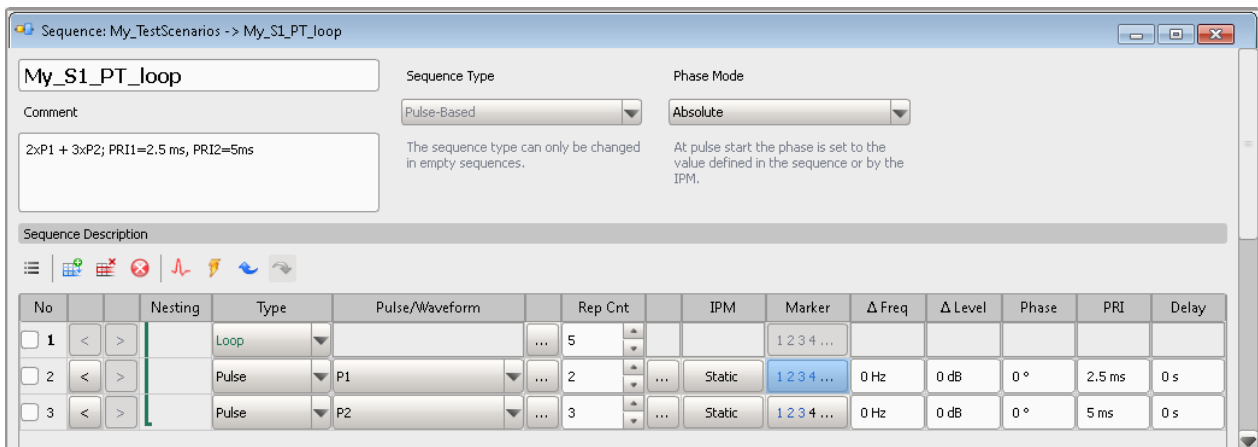
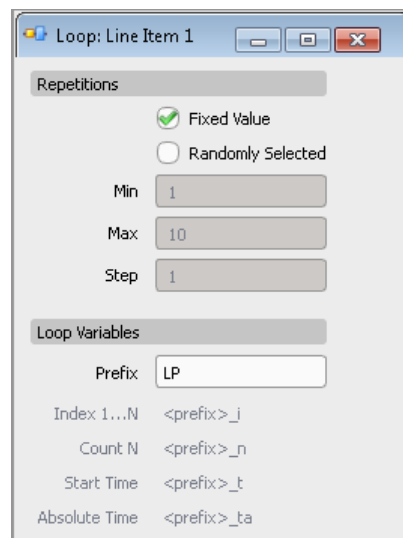
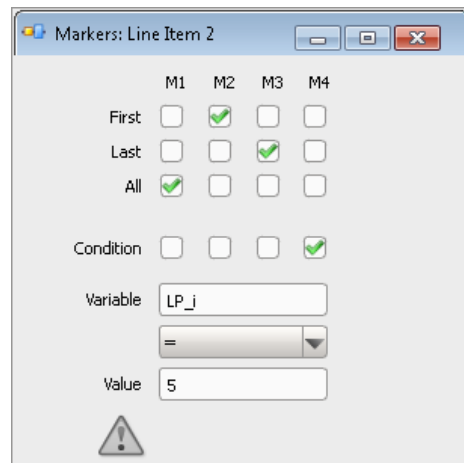


Figure 20-1: Example of a sequence composed of two pulses included into a loop

2. Select "Loop > ..." and enter "Loop Variables > Prefix = LP".



3. Select "Sequence Description table > Item#2 > Marker".
4. Select "M1 > All", "M2 > First", "M3 > Last".



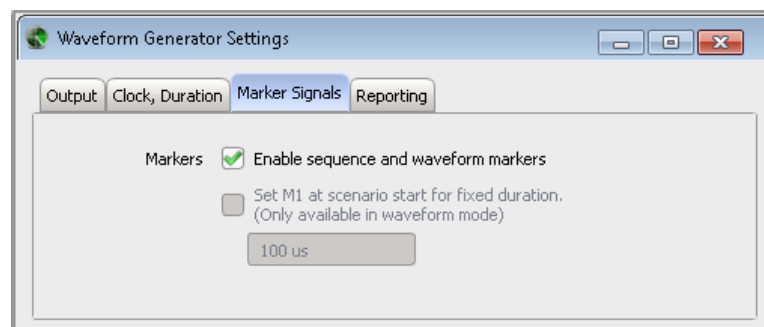
The Marker 2 is generated only for the first pulse out of all repetitions.

5. Select "M4 > Condition", "Variable = LP_i" and "Value = 5".

If the condition $LP_i = 5$ is fulfilled, a marker is generated. That is, marker 4 is generated during the fifth loop run, see [Figure 20-2](#).

To enable marker signals in the output waveform

1. Select "Repository Tree > Scenario > Waveform Generator > Config".
2. Select "Marker Signals > Enable sequence and waveform markers".



See also "[Waveform Generation](#)" on page 59.

3. Optionally, select "Set at scenario start for fixed duration > On" and set "Duration = 1 ms", to enable an extra global signal.

This marker signal is output as marker 1. If other marker signals, like for example pulse and sequence markers, are defined, all markers are summed.

How to visualize the configured markers

1. Open the "Waveform/Data Viewer", see "[To access the "Waveform/Data View" dialog](#)" on page 269.

2. Select "Marker > On".

The "Waveform/Data Viewer" evaluates the marker information and displays the marker traces M1 to M4.

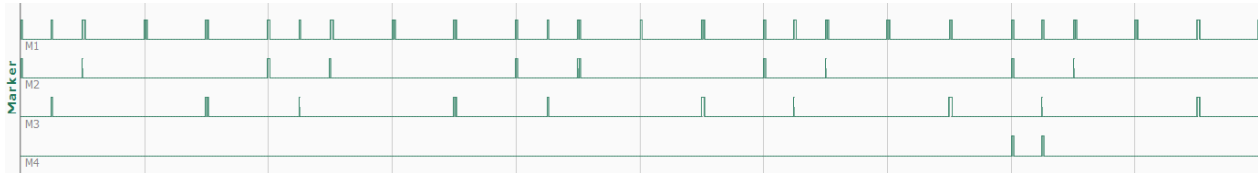


Figure 20-2: Example: Waveform View > Marker traces (signal portion)

M1 = All

M2 = First

M3 = Last

M4 = Condition $LP_i = 5$, i.e. M4 is generated for all P1 pulses during the fifth loop run

21 Creating Reports and Documenting Measurement Results

In R&S Pulse Sequencer, you can document your measurement results and software configuration in the following ways:

- Create and export a repository archive, so that you can repeat measurements under the same conditions
- Create automatically scaled hardcopy of current screens or dialogs
- Print antenna diagrams together with additional information
- Create and store reports in text files, in HTML format or in Excel files with several spread sheets.

This section focuses on the built-in report generation function. For information on the other functions, see:

- ["To create and export a repository archive"](#) on page 51
- ["To set the size and color scheme of the hardcopies"](#) on page 37
- [Chapter 10, "Defining Antenna Patterns and Antenna Scans"](#), on page 161

Report generation

The R&S Pulse Sequencer provides a built-in report generation function so that all parameters used during the waveform generations are automatically stored into a printable file. The target directory, in that the generated reports are stored, is configurable.

Report files can be formatted based on:

- **Template**
The reporting uses a predefined but editable template. Generated is an ASCII text file or an HTML page, where data is formatted in columns, including header description. The reporting files in ASCII format use the predefined file extension *.pwd

– ASCII text file

This is a template-based report.

```
User:      R&S
Rep Path:  C:\_PS_files\reports

Date:      28.10.2015 10:07:25
ISO Date:  2015-10-28T10:07:25

Repository: My_TestScenarios
Comment:
Version:   2.3
Author:    Rohde&Schwarz
Created:   25 Jun 2014 15:27:04
Path:      C:\Users\Public\Documents\Rohde-Schwarz\Pulse Sequencer\Repositories\20141124_100351

Scenario:  SimplePulseTrain
Comment:
```

TOA ns	RF GHZ	PW us	PA dBm	MF	MOP	BW kHz	Rep
0	3.0	120.000	-30.0	0	NONE	0	1
2500000	3.0	120.000	-30.0	0	NONE	0	2
5000000	3.0	220.000	-30.0	0	NONE	0	1
10000000	3.0	220.000	-30.0	0	NONE	0	2
15000000	3.0	220.000	-30.0	0	NONE	0	3
22000000	3.0	120.000	-40.0	0	NONE	0	1
24500000	3.0	120.000	-40.0	0	NONE	0	2

Figure 21-1: Example of an ASCII report file ("Type > Template")

- HTML page

PDW Report Generated with PS Example

User	R&S
Repository Path	C:\PS_files\reports
Date	28.10.2015 08:15:18
ISO Date	2015-10-28T08:15:18

Repository	My_TestScenarios
Comment	
Version	2.3
Author	Rohde.&Schwarz
Created	25 Jun 2014 15:27:04
Path	C:\Users\Public\Documents\Rohde-Schwarz\Pulse Sequencer\Repositories\20141124_100351

SimplePulseTrain

TOA	RF	PW	PA	MF	MOP	BW	Rep
ns	GHz	us	dBm			kHz	
0	3.0	120.000	-30.0	0	NONE	0	1
2500000	3.0	120.000	-30.0	0	NONE	0	2
5000000	3.0	220.000	-30.0	0	NONE	0	1
10000000	3.0	220.000	-30.0	0	NONE	0	2
15000000	3.0	220.000	-30.0	0	NONE	0	3
22000000	3.0	120.000	-40.0	0	NONE	0	1
24500000	3.0	120.000	-40.0	0	NONE	0	2

Figure 21-2: Example of an HTML report file ("Type > Template")

- **User-defined plugin**

This reporting creates reports according to custom templates, like DFS or DIV8 PDW.

You can for example create a Microsoft Excel plugin for example, that retrieves the pulse parameters and it fills them in a spread sheet.

For detailed description of the file formats, the template and the plugin, see:

- [Chapter A.1, "File Format of the Reporting Template"](#), on page 481
- [Chapter C, "Plugin Programming API"](#), on page 495
- [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.

Status information on whether report logging is enabled or not is displayed in the "Scenario" dialog, see [Figure 6-1](#).

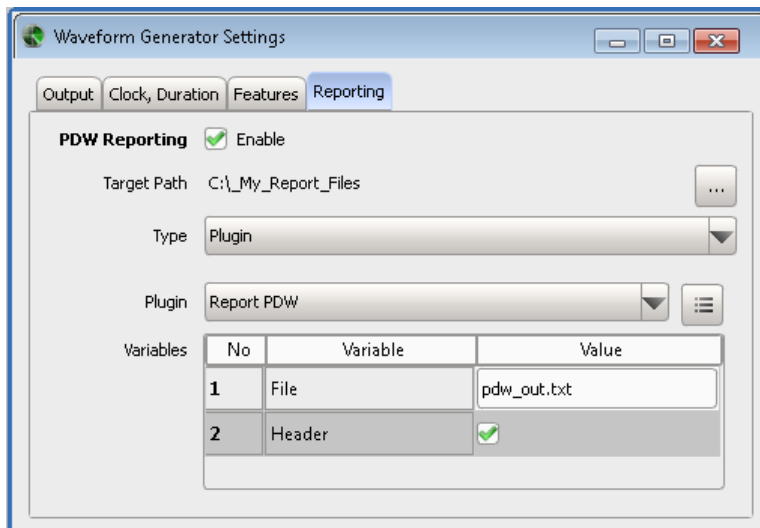
See also [Chapter 21.2, "How to Create Test Reports"](#), on page 312.

21.1 Reporting Settings

Access:

1. In the "Scenario" dialog, select "Waveform Generation > Config".

2. Select "Reporting".



Available settings:

Reporting Enable..... 310
 Target Path, Set Path..... 310
 Type..... 310
 Edit Template..... 311
 Plugin and Plugin Variables..... 312

Reporting Enable

Enables generation of reports.

Remote command:

[SCENario:PDW:ENABle](#) on page 393

Target Path, Set Path

Indicates current storage location.

Per default, report logs are stored in the user directory of the current user, that is the %HOMEPATH% directory.

To change the storage location, select the "Set Path" icon.

See also "[To set the storage location for the generated report files](#)" on page 312.

Remote command:

[SCENario:PDW:PATH](#) on page 393

Type

Sets the template used by the reporting function.

- "Default" Loads the predefined Template.
 The report is a plain text file. The format is simple, column-based and cannot be modified. Further settings are not required.
- "Template" Enables the predefined template for editing, see "[Edit Template](#)" on page 311.

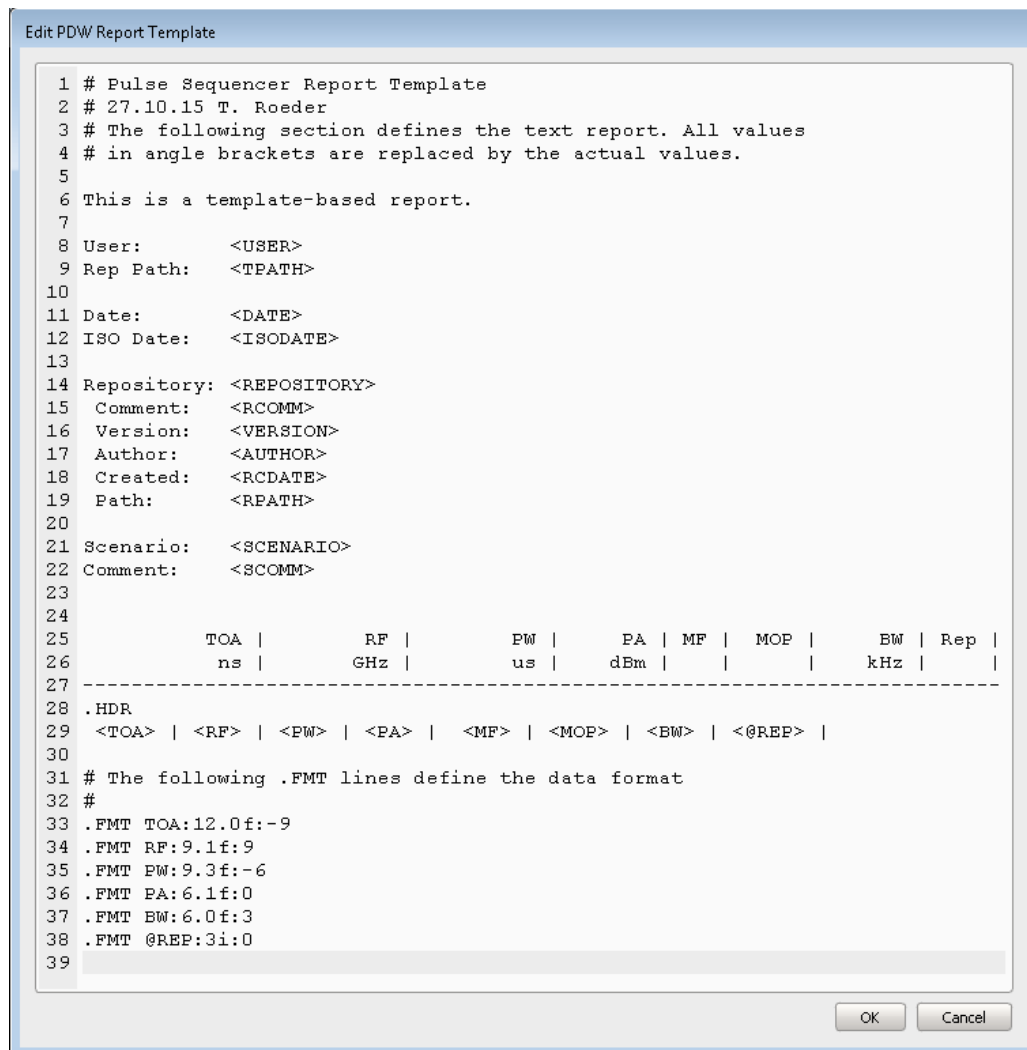
"Plugin" Enables loading according to custom templates, like DFS, or DIV8 PDW.
 Custom templates are used in the same way as the predefined template, see ["Plugin and Plugin Variables"](#) on page 312.

Remote command:

`SCENario:PDW:TYPE` on page 394

Edit Template

Opens the predefined template in edit mode.



```

Edit PDW Report Template

1 # Pulse Sequencer Report Template
2 # 27.10.15 T. Roeder
3 # The following section defines the text report. All values
4 # in angle brackets are replaced by the actual values.
5
6 This is a template-based report.
7
8 User:      <USER>
9 Rep Path:  <TPATH>
10
11 Date:     <DATE>
12 ISO Date: <ISODATE>
13
14 Repository: <REPOSITORY>
15 Comment:   <RCOMM>
16 Version:   <VERSION>
17 Author:    <AUTHOR>
18 Created:   <RCDATE>
19 Path:      <RPATH>
20
21 Scenario:  <SCENARIO>
22 Comment:   <SCOMM>
23
24
25          TOA |      RF |      PW |      PA | MF | MOP |      BW | Rep |
26          ns |     GHz |     us |     dBm |  |  |     kHz |  |
27 -----
28 .HDR
29 <TOA> | <RF> | <PW> | <PA> | <MF> | <MOP> | <BW> | <@REP> |
30
31 # The following .FMT lines define the data format
32 #
33 .FMT TOA:12.0f:-9
34 .FMT RF:9.1f:9
35 .FMT PW:9.3f:-6
36 .FMT PA:6.1f:0
37 .FMT BW:6.0f:3
38 .FMT @REP:3i:0
39
  
```

Figure 21-3: Example: Template that creates text reports in ASCII format

Note: The software uses the last modified template version. Previous versions cannot be recalled.

Before you modify the template, store the template content as a text file.

See [Chapter A.1, "File Format of the Reporting Template"](#), on page 481 for description of the file format.

Remote command:

[SCENario:PDW:TEMPlate](#) on page 394

Plugin and Plugin Variables

Selects and loads a reporting template from a *.dll file. This template must exist in the "Plugin" library.

Report plugins can register a set of variables that can be used to control the plugin's algorithms.

See:

- [Chapter C, "Plugin Programming API"](#), on page 495
- ["To import a plugin"](#) on page 297.

Remote command:

[SCENario:PDW:PLUGIn:NAME](#) on page 394

[SCENario:PDW:PLUGIn:VARiable:CATalog?](#) on page 393

[SCENario:PDW:PLUGIn:VARiable:SElect](#) on page 393

[SCENario:PDW:PLUGIn:VARiable:VALue](#) on page 393

21.2 How to Create Test Reports

This section shows how to:

- ["To enable report file generation"](#) on page 312
- ["To set the storage location for the generated report files"](#) on page 312
- ["To change the report type"](#) on page 313

To enable report file generation

The "Scenario" dialog provides information on the report logging status. The related information is displayed to the right of the "Comment" field.

Sequences (Collection)

Plugin-based report is written to C:_My_Report_Files.
Data output is in MSW or waveform format.

To enable or disable the report file generation, proceed as following:

1. In the "Scenario" dialog, select "Waveform Generation > Config".
2. In the "Waveform Generation Settings" dialog, select "Reporting > Reporting Enable > On".

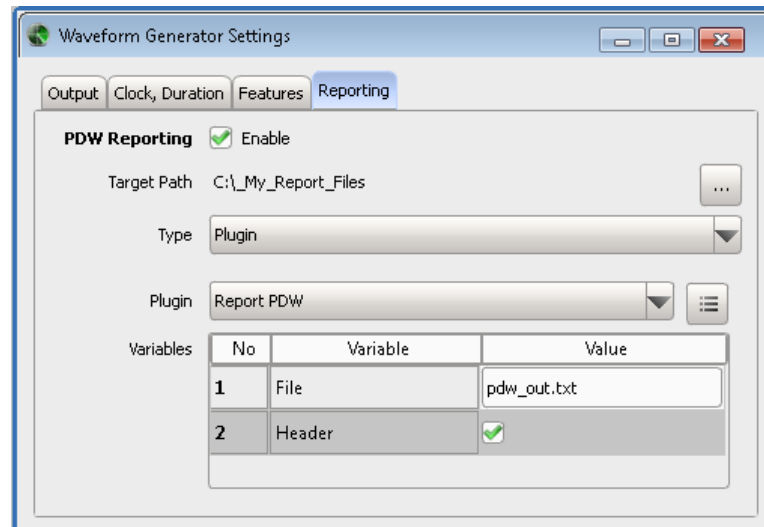
To set the storage location for the generated report files

Per default, report files are stored in the user home directory of the current user, that is the %HOMEPATH% directory.

1. Open the "Scenario" dialog to retrieve information on the current storage location. Information is displayed to the right of the "Comment" field.

2. Select "Waveform Generation > Config > Reporting".

The parameter "Target Path" indicates the current storage location.



3. To change this location, select the "Set Path" icon, navigate to the directory, and confirm with ok.

To change the report type

You may prefer to store the test reports in text form or as HTML page, if for example Microsoft Excel is not installed on your PC.

To change the type of the generated reports, proceed as following:

1. In the "Scenario" dialog, select "Waveform Generation > Config > Report".
2. Select "Type > Template".
3. Select "Edit" to display the current template.

```

1 # Pulse Sequencer Report Template
2 # 27.10.15 T. Roeder
3 # The following section defines the text report. All values
4 # in angle brackets are replaced by the actual values.
5
6 This is a template-based report.
7
8 User:      <USER>
9 Rep Path:  <TPATH>
10
11 Date:     <DATE>
12 ISO Date: <ISODATE>
13
14 Repository: <REPOSITORY>
15 Comment:   <RCOMM>
16 Version:   <VERSION>
17 Author:    <AUTHOR>
18 Created:   <RCDATE>
19 Path:      <RPATH>
20
21 Scenario:  <SCENARIO>
22 Comment:   <SCOMM>
23
24
25          TOA |      RF |      PW |      PA | MF | MOP |      BW | Rep |
26          ns |      GHz |      us |      dBm |  |  |      kHz |  |
27 -----
28 .HDR
29 <TOA> | <RF> | <PW> | <PA> | <MF> | <MOP> | <BW> | <@REP> |
30
31 # The following .FMT lines define the data format
32 #
33 .FMT TOA:12.0f:-9
34 .FMT RF:9.1f:9
35 .FMT PW:9.3f:-6
36 .FMT PA:6.1f:0
37 .FMT BW:6.0f:3
38 .FMT @REP:3i:0
39

```

You can also change the used templates or create and load your plugin.

See:

- [Chapter A.1, "File Format of the Reporting Template"](#), on page 481
- [Chapter C, "Plugin Programming API"](#), on page 495
- [Chapter 19, "Defining Complex Modulation Schemes and IPM Profiles"](#), on page 297.

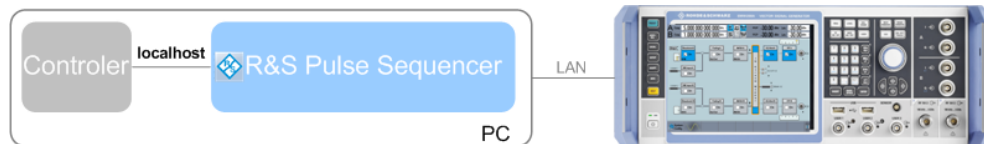
22 Automation of R&S Pulse Sequencer

In addition to working with the R&S Pulse Sequencer software interactively, it is also possible to operate and control it from a remote PC. Remote control operation allows you to automate the configuration process and is useful when a higher configuration speed is required.

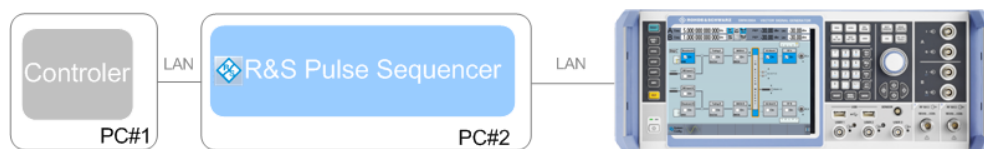
Remote control is an operation by which the software is operated remotely from a so called controller. A controller can be any PC on which some required software is installed and configured. R&S Pulse Sequencer might be but not necessarily is installed on the same PC. In case R&S Pulse Sequencer is not physically located on the controller's PC, a LAN connection between both PCs is required.

The following figures depict the following three possible cases:

- The controller is a script in the R&S Pulse Sequencer itself and controls the software and possibly also other instruments.
- The controller and R&S Pulse Sequencer are on the same PC.



- In some rare cases, the controller and the software are installed on two separate PCs.



In remote control operation, the software itself is operated by remote control commands. For description of specific SCPI commands, refer to [Chapter 23, "Remote-Control Commands"](#), on page 321.



We assume that the controller and the signal generator have already been set up for remote operation in a network as described in the documentation of the instrument. A knowledge about the remote control operation and the SCPI command syntax are assumed.

Supported interfaces and protocols

R&S Pulse Sequencer support remote control over the LAN interface and with the socket protocol. The socket controller is sufficient; a VISA (Virtual Instrument Software Architecture) library is not required.

Socket communication requires the specification of the port (commonly referred to as port number); the registered port is 5025.

Socket communication

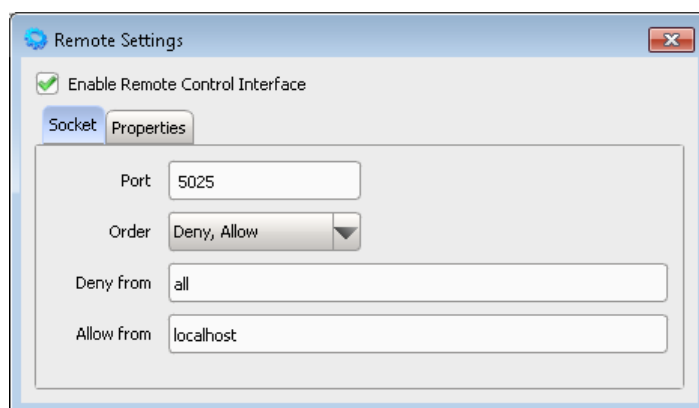
Socket communication is a simple network communication and is also referred to as "Raw Ethernet communication". It is available by default on all operating systems.

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 programs, user-defined sockets can be programmed.

22.1 Remote Settings

Access:

- In the menu bar, select "Configure > Remote Control".



Enable Remote Control Interface

Enables the remote control interface.

Socket

Configures the remote control interface.

- "Port" Do not change the default socket port!
Sets the port number of the remote control interface.
- "Order" Sets the order of the deny/allow rules.
- "Deny from/Allow from"
Deny or allow remote access.

Properties

Set the serial number of the software. The serial number is a unique 6-digit number, returned as a result of the *IDN? query.

See ["To identify the R&S Pulse Sequencer by its serial number"](#) on page 319.

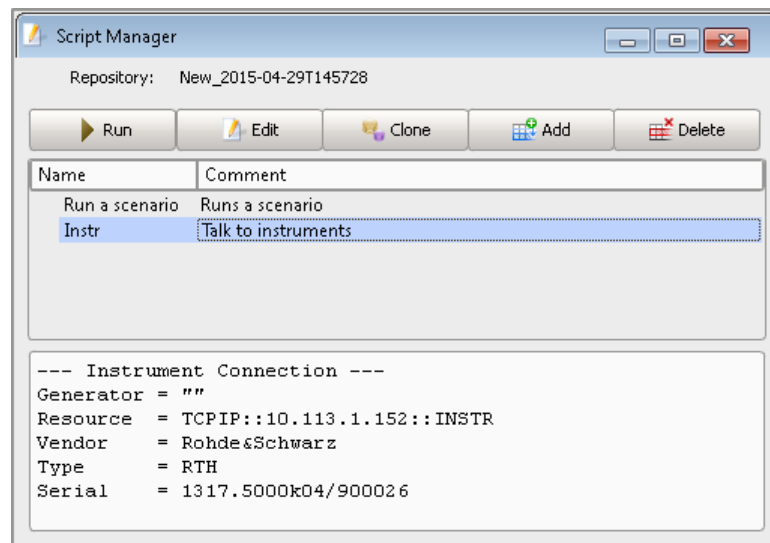
22.2 Script Manager Settings

The R&S Pulse Sequencer provides also a built-in java script editor. With this function, you can write scripts that control not only the software itself but also the instruments, that are part of the test setup, like one or more signal generators, or an oscilloscope.

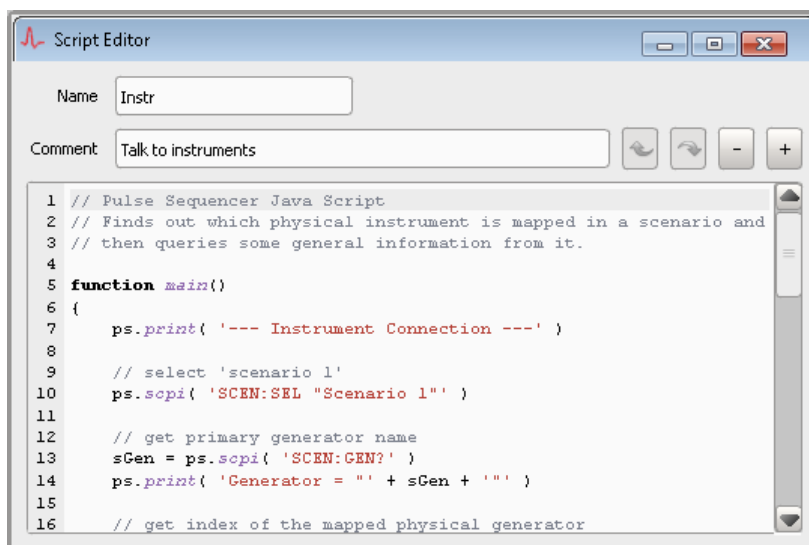
To access the script manager

1. In the menu bar, select "Script > Manager".

The "Script Manager" lists the available scripts with their name and comment.



2. To execute a script, select "Run".
The script output from the `ps.print` function is displayed in the "Script Manager".
3. To create a script, select "Add" or use the "Clone" function.
4. To edit a script, double click on a script in the script list or select a script and select "Edit".



You can write or copy your java script file here.

For description of the script syntax and script examples, see:

- [Chapter D, "Scripting API"](#), on page 510
- [Chapter D.3, "Script Examples"](#), on page 518

[Script Manager](#).....318
[Script Editor](#).....318

Script Manager

The dialog indicates the current repository and lists the available scripts with their name and comment.

If a script is executed, the script output, i.e. the output from the `ps.print` function, is displayed in the lower part of the dialog.

- "Run" Executes the selected script.
- "Edit" Opens the selected script for editing in the "Script Editor".
- "Clone, Add, Delete" Performs the corresponding action.

Script Editor

The script editor is a dialog, where you can change the name of a script, add a comment and create or edit the script code.

The icons provide common "Undo/Redo" and "Zoom In/Out" functions.

22.3 How to Configure and Enable Remote Control of R&S Pulse Sequencer

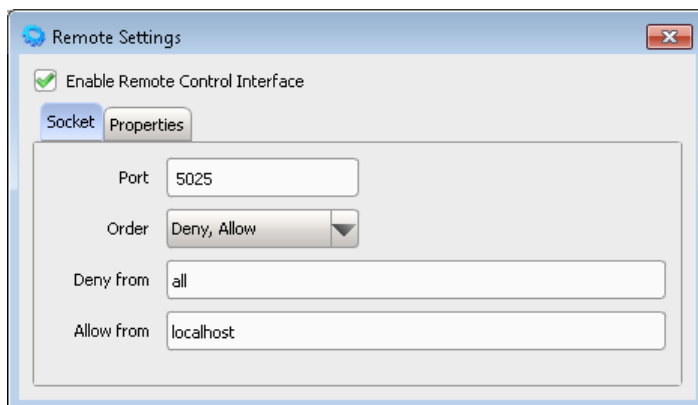
See:

- ["To access the required settings"](#) on page 319

- ["To use the console to test SCPI commands"](#) on page 320
- ["To find the SCPI command corresponding to a parameter on the user interface"](#) on page 320

To access the required settings

1. In the menu bar, select "Configure > Remote Control".
2. Select "Enable Remote Control Interface".
3. Do not change the default socket port.

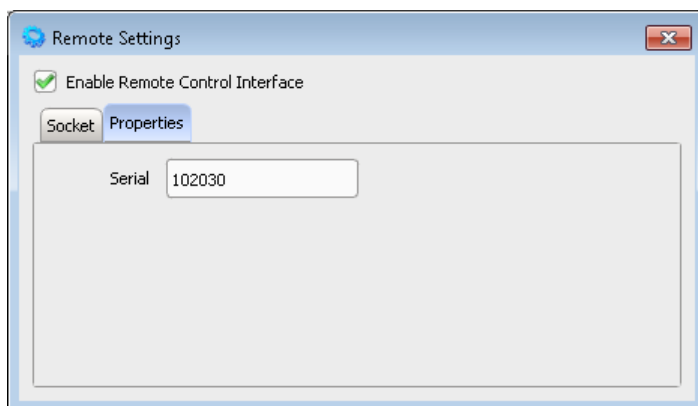


To identify the R&S Pulse Sequencer by its serial number

If you remotely control several R&S Pulse Sequencer from the same controller, it is useful to use serial numbers to distinguish between the different installations.

To set the serial number of the software:

1. In the menu bar, select "Configure > Remote Control".
2. Open the "Properties" tab.
3. In the "Serial" field, enter a unique 6-digit number.



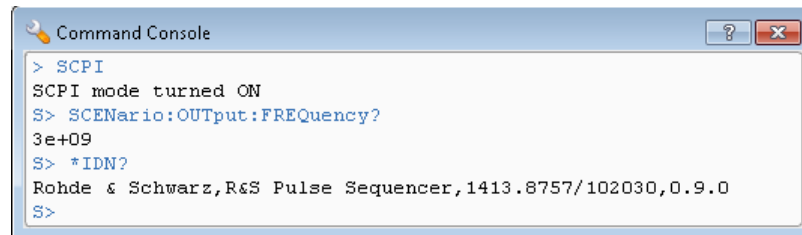
To query the serial number, use the command `*IDN?`.

See also ["To use the console to test SCPI commands"](#) on page 320.

To use the console to test SCPI commands

R&S Pulse Sequencer provides a built-in console window that allows you to test SCPI commands or commands sequences.

1. To access this window, in the menu bar, select "Window > Console"
2. In the "Command Console" window, type `SCPI` to start the remote control mode.
3. Enter any SCPI command, see "[To use the console to test SCPI commands](#)" on page 320.



```
Command Console
> SCPI
SCPI mode turned ON
S> SCENario:OUTput:FREQuency?
3e+09
S> *IDN?
Rohde & Schwarz,R&S Pulse Sequencer,1413.8757/102030,0.9.0
S>
```

To find the SCPI command corresponding to a parameter on the user interface

Perform one of the following:

1. Right mouse click to open the context menu of the parameter.
2. Press F1 to open the Help window. You find a link to the description of the SCPI command right after the parameter description.

For a concise description of the available SCPI commands, see [Chapter 23, "Remote-Control Commands"](#), on page 321.

23 Remote-Control Commands

The following commands are required to perform signal generation with the R&S Pulse Sequencer option in a remote environment. We assume that the R&S Pulse Sequencer has already been set up for remote operation in a network as described in the R&S Pulse Sequencer documentation. A knowledge about the remote control operation and the SCPI command syntax are assumed.

• Conventions used in SCPI Command Descriptions	321
• Programming Examples	322
• Common Commands	322
• Commands with Similar Syntax	323
• Antenna Pattern Commands	337
• Antenna Scan Commands	348
• Dashboard Commands	356
• Data Source Commands	360
• Signal to Generator Mapping Commands	362
• Emitter Commands	366
• Generator Profiles and Instruments Commands	369
• Inter-Pulse Modulation Commands	373
• Marker Commands	385
• Plugin and Reporting Commands	390
• Pulse Commands	395
• Receiver Commands	415
• Repository Commands	418
• Scenario Commands	425
• Sequence Commands	445
• Status Commands	454
• System Commands	455
• Waveform Commands	457
• Waveform Viewer Commands	461
• List of Remote Commands	462

23.1 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 Pulse Sequencer 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.

23.2 Programming Examples

The following sections provide simple programming examples for the R&S Pulse Sequencer. The purpose of the examples is to present **all** commands for a given task. In real applications, one would rather reduce the examples to an appropriate subset of commands.

The programming examples have been tested with the built-in console which provides an environment for the development and execution of remote tests. To keep the examples as simple as possible, only the "clean" SCPI-like syntax elements are reported. Non-executable command lines (for example comments) start with two // characters.

23.3 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:

*IDN?	323
*OPT?	323
*RST	323

***IDN?**

Identification

Returns the instrument identification.

Return values:

<ID> "Rohde&Schwarz,<device type>,<serial number>,<firmware version>"

Example:

Rohde&Schwarz,K300,1407.6004k02/000000,
3.1.17.1-03.01.158

Usage:

Query only

***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

***RST**

Reset

Sets the software to a defined default status. The default settings are indicated in the description of commands.

Unsaved settings are lost.

Usage:

Setting only

23.4 Commands with Similar Syntax

This section describes commands, that follow similar syntax and are common to the command groups listed in the following sections.

Commands for handling of repository elements

Repository elements are referenced by their *unique name*. The name may contain empty spaces. Almost in each of the command groups, there is a `...:SElect` command. Use this command to select one repository element to which the subsequent commands apply.

A typical remote control sequence would look like in the [Example "Handling repository elements"](#) on page 324.

Example: Handling repository elements

The following is a simple example that explains the principle of repository elements handling by using the common commands.

```
// activate the remote control mode
SCPI
// create new empty repository
REPository:CREate "Repository for tests"
REPository:CATalog?
// "Repository for tests"
REPository:SAVE

// create two new scenarios
SCENario:CREate "New 1"
SCENario:CREate "New 2"
SCENario:SElect "New 1"
SCENario:COMment "simple test scenario"
// rename the scenario
SCENario:NAME "New"
// remove it from the repository
SCENario:CATalog?
// "New", "New 2"
SCENario:REMove "New 2"
SCENario:CATalog?
// "New"

// create new pulse
// PULSe:CREate "New"
PULSe:CATalog?
// "New"
// although the pulse and the scenario use the same name,
// they are different and unique elements for the repository
```

Commands for handling of list and table items

Several repository elements are described in table form or as lists, like for example the sequence description table, the FM and AM Step modulations, the data source, IPM profiles in list form, or lists of multiple emitters.

Lists and tables are composed of items, where items are referenced by their *number*. Use the `...:SElect` command to select one item to which the subsequent commands apply.

A typical remote control sequence would look like in the [Example "Handling items"](#) on page 325.

Example: Handling items

The following is a simple example that explains the principle of items handling by using the common commands.

```
// activate the remote control mode
SCPI
// creates a pulse-based sequence with 2 items (i.e. segments)
SEquence:CREate "Seq1"
SEquence:SElect "Seq1"
SEquence:TYPE PULSe
SEquence:ITEM:ADD
SEquence:ITEM:COUNT?
// 1
SEquence:ITEM:SElect 1; SEquence:ITEM:TYPE PULSe
SEquence:ITEM:PULSe "P1"
SEquence:ITEM:ADD
SEquence:ITEM:COUNT?
// 2
SEquence:ITEM:DElete 2
SEquence:ITEM:COUNT?
// 1
SEquence:ITEM:CLEar
SEquence:ITEM:COUNT?
// 0
```

ANTenna:CATalog?.....	329
DSRC:CATalog?.....	329
EMITter:CATalog?.....	329
RECeiver:CATalog?.....	329
GENerator:CATalog?.....	329
IPM:CATalog?.....	329
PLUGin:CATalog?.....	329
PULSe:CATalog?.....	329
SCAN:CATalog?.....	329
SCENario:CATalog?.....	329
SEquence:CATalog?.....	329
WAVEform:CATalog?.....	329
REPository:CATalog?.....	329
ANTenna:CREate.....	329
DSRC:CREate.....	329
EMITter:CREate.....	329
RECeiver:CREate.....	329
GENerator:CREate.....	329
IPM:CREate.....	329
PLUGin:CREate.....	329
PULSe:CREate.....	329
SCAN:CREate.....	329
SCENario:CREate.....	329
SEquence:CREate.....	329
WAVEform:CREate.....	329

REPository:CREate.....	329
ANTenna:SElect.....	330
DSRC:SElect.....	330
EMITter:SElect.....	330
RECeiver:SElect.....	330
GENerator:SElect.....	330
IPM:SElect.....	330
PLUGin:SElect.....	330
PULSe:SElect.....	330
SCAN:SElect.....	330
SCENario:SElect.....	330
SEQuence:SElect.....	330
WAVeform:SElect.....	330
REPository:SElect.....	330
ANTenna:NAME.....	330
DSRC:NAME.....	330
EMITter:MODE:NAME.....	330
EMITter:NAME.....	330
RECeiver:NAME.....	330
GENerator:NAME.....	330
IPM:NAME.....	330
IPM:PLUGin:NAME.....	330
PLUGin:MODule:NAME?.....	330
PLUGin:NAME.....	330
PULSe:MOP:PLUGin:NAME.....	330
PULSe:NAME.....	330
SCAN:NAME.....	331
SCENario:NAME.....	331
SEQuence:NAME.....	331
WAVeform:NAME.....	331
ANTenna:COMMeNt.....	331
DSRC:COMMeNt.....	331
EMITter:COMMeNt.....	331
RECeiver:COMMeNt.....	331
GENerator:COMMeNt.....	331
IPM:COMMeNt.....	331
PLUGin:COMMeNt.....	331
PULSe:COMMeNt.....	331
PULSe:MOP:COMMeNt.....	331
SCAN:COMMeNt.....	331
SCENario:COMMeNt.....	331
SEQuence:COMMeNt.....	331
WAVeform:COMMeNt.....	331
REPository:COMMeNt.....	331
ANTenna:REMOve.....	331
DSRC:REMOve.....	331
EMITter:REMOve.....	331
RECeiver:REMOve.....	331
GENerator:REMOve.....	331
IPM:REMOve.....	331

PLUGin:REMove.....	331
PULSe:REMove.....	331
SCAN:REMove.....	331
SCENario:REMove.....	331
SEQuence:REMove.....	331
WAVeform:REMove.....	332
REPository:REMove.....	332
ASSignment:GENerator:PATH:EMITter:ADD.....	332
ASSignment:GENerator:PATH:ANTenna:ADD.....	332
EMITter:MODE:ADD.....	332
EMITter:MODE:BEAM:ADD.....	332
RECeiver:ANTenna:ADD.....	332
PULSe:MOP:AMSTep:ADD.....	332
PULSe:MOP:FMSTep:ADD.....	332
PULSe:MOP:PCHirp:ADD.....	332
PULSe:MOP:PLISt:ADD.....	332
SCENario:LOCalized:ADD.....	332
SCENario:DIRection:ADD.....	332
SCENario:CSEQuence:ADD.....	332
SCENario:CEMit:ADD.....	332
DSRC:ITEM:ADD.....	332
IPM:LIST:ITEM:ADD.....	332
PULSe:ENVelope:DATA:ITEM:ADD.....	332
SEQuence:ITEM:IPM:ADD.....	332
SEQuence:ITEM:ADD.....	332
INSTrument:COUNt?.....	333
EMITter:MODE:BEAM:COUNt?.....	333
EMITter:MODE:COUNt?.....	333
PULSe:ENVelope:DATA:ITEM:COUNt?.....	333
PULSe:MOP:AMSTep:COUNt?.....	333
PULSe:MOP:FMSTep:COUNt?.....	333
PULSe:MOP:PCHirp:COUNt?.....	333
PULSe:MOP:PLISt:COUNt?.....	333
IPM:LIST:ITEM:COUNt?.....	333
SEQuence:ITEM:IPM:COUNt?.....	333
SEQuence:ITEM:COUNt?.....	333
SCENario:LOCalized:SElect.....	333
SCENario:DIRection:SElect.....	333
SCENario:CSEQuence:SElect.....	333
SCENario:CEMit:SElect.....	333
EMITter:MODE:BEAM:SElect.....	333
EMITter:MODE:SElect.....	333
RECeiver:ANTenna:SElect.....	333
PULSe:ENVelope:DATA:ITEM:SElect.....	333
PULSe:MOP:AMSTep:SElect.....	333
PULSe:MOP:FMSTep:SElect.....	333
PULSe:MOP:PCHirp:SElect.....	333
PULSe:MOP:PLISt:SElect.....	333
DSRC:ITEM:SElect.....	333
IPM:LIST:ITEM:SElect.....	333

SEquence:ITEM:IPM:SElect.....	333
SEquence:ITEM:SElect.....	333
INSTRument:SElect.....	334
DASHboard:USED:SElect.....	334
DASHboard:UNUSed:SElect.....	334
DASHboard:UNUSed:PATH:SElect.....	334
DASHboard:USED:PATH:SElect.....	334
ASSignment:ANTennas:SElect.....	334
ASSignment:EMITters:SElect.....	334
ASSignment:GENerator:SElect.....	334
ASSignment:GENerator:PATH:SElect.....	334
ASSignment:GENerator:PATH:EMITter:SElect.....	334
ASSignment:GENerator:PATH:ANTenna:SElect.....	334
PULSe:MOP:AMSTep:INSert.....	334
PULSe:MOP:FMSTep:INSert.....	334
PULSe:MOP:PCHirp:INSert.....	334
PULSe:MOP:PLISt:INSert.....	334
INSTRument:DELeTe.....	334
ASSignment:GENerator:PATH:EMITter:DELeTe.....	334
ASSignment:GENerator:PATH:ANTenna:DELeTe.....	334
DSRC:ITEM:DELeTe.....	334
EMITter:MODE:BEAM:DELeTe.....	334
EMITter:MODE:DELeTe.....	334
RECeiver:ANTenna:DELeTe.....	334
IPM:LIST:ITEM:DELeTe.....	334
PULSe:ENVELOpe:DATA:ITEM:DELeTe.....	334
PULSe:MOP:AMSTep:DELeTe.....	334
PULSe:MOP:FMSTep:DELeTe.....	335
PULSe:MOP:PCHirp:DELeTe.....	335
PULSe:MOP:PLISt:DELeTe.....	335
SCENario:LOCALized:DELeTe.....	335
SCENario:DIRection:DELeTe.....	335
SCENario:CSEquence:DELeTe.....	335
SCENario:CEMit:DELeTe.....	335
SEquence:ITEM:IPM:DELeTe.....	335
SEquence:ITEM:DELeTe.....	335
SCENario:GENerator:CLEar.....	335
INSTRument:CLEar.....	335
ANTenna:MODEL:USER:CLEar.....	335
DSRC:CLEar.....	335
EMITter:MODE:ANTenna:CLEar.....	335
EMITter:MODE:BEAM:CLEar.....	335
EMITter:MODE:CLEar.....	335
EMITter:MODE:SCAN:CLEar.....	335
RECeiver:ANTenna:CLEar.....	335
IPM:LIST:CLEar.....	335
PULSe:ENVELOpe:DATA:CLEar.....	335
PULSe:MOP:AMSTep:CLEar.....	335
PULSe:MOP:FMSTep:CLEar.....	335
PULSe:MOP:PCHirp:CLEar.....	335

PULSe:MOP:PLISt:CLEAr.....	335
ASSignment:GENerator:PATH:EMITter:CLEAr.....	335
ASSignment:GENerator:PATH:ANTenna:CLEAr.....	335
SCENario:EMITter:CLEAr.....	335
SCENario:CSEquence:CLEAr.....	335
SCENario:CEMit:CLEAr.....	335
SCENario:LOCalized:CLEAr.....	335
SCENario:DIRection:CLEAr.....	335
SEquence:ITEM:CLEAr.....	335

ANTenna:CATalog?
DSRC:CATalog?
EMITter:CATalog?
RECeiver:CATalog?
GENerator:CATalog?
IPM:CATalog?
PLUGin:CATalog?
PULSe:CATalog?
SCAN:CATalog?
SCENario:CATalog?
SEquence:CATalog?
WAVEform:CATalog?
REPository:CATalog?

Queries the available repository elements in the database.

Return values:

<Catalog> string

Example: See [Example "Working with repositories"](#) on page 418

Usage: Query only

Manual operation: See ["General Repository Settings"](#) on page 43

ANTenna:CREate <Create>
DSRC:CREate <Create>
EMITter:CREate <Create>
RECeiver:CREate <Create>
GENerator:CREate <Create>
IPM:CREate <Create>
PLUGin:CREate <Create>
PULSe:CREate <Create>
SCAN:CREate <Create>
SCENario:CREate <Create>
SEquence:CREate <Create>
WAVEform:CREate <Create>
REPository:CREate <Create>

Creates a repository element with the selected name.

Setting parameters:

<Create> string
Must be unique for the particular type of repository elements.
May contain empty spaces.

Example: see [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["General Repository Settings"](#) on page 43

ANTenna:SElect <Select>
DSRC:SElect <Select>
EMITter:SElect <Select>
RECeiver:SElect <Select>
GENerator:SElect <Select>
IPM:SElect <Select>
PLUGin:SElect <Select>
PULSe:SElect <Select>
SCAN:SElect <Select>
SCENario:SElect <Select>
SEquence:SElect <Select>
WAVEform:SElect <Select>
REPository:SElect <Select>

Selects the repository element to which the subsequent commands apply.

Parameters:

<Select> string
Element name, as defined with the ...:CREate or ...:NAME command.
To query the existing elements, use the ...:CATalog? command.
For example, `REPository:CATalog?`.

Example: See [Example "Working with repositories"](#) on page 418.

Manual operation: See ["General Repository Settings"](#) on page 43

ANTenna:NAME <Name>
DSRC:NAME <Name>
EMITter:MODE:NAME <Name>
EMITter:NAME <Name>
RECeiver:NAME <Name>
GENerator:NAME <Name>
IPM:NAME <Name>
IPM:PLUGin:NAME <Name>
PLUGin:MODule:NAME?
PLUGin:NAME <Name>
PULSe:MOP:PLUGin:NAME <Name>
PULSe:NAME <Name>

SCAN:NAME <Name>
SCENario:NAME <Name>
SEQuence:NAME <Name>
WAVeform:NAME <Name>

Renames the selected repository element.

Parameters:

<Name> string
 Must be unique for the particular type of repository elements.
 May contain empty spaces.

Example: see [Example "Handling repository elements"](#) on page 324

Manual operation: See ["Waveform Name and Comment"](#) on page 224

ANTenna:COMMeNt <Comment>
DSRC:COMMeNt <Comment>
EMITter:COMMeNt <Comment>
RECeiver:COMMeNt <Comment>
GENerator:COMMeNt <Comment>
IPM:COMMeNt <Comment>
PLUGin:COMMeNt <Comment>
PULSe:COMMeNt <Comment>
PULSe:MOP:COMMeNt <Comment>
SCAN:COMMeNt <Comment>
SCENario:COMMeNt <Comment>
SEQuence:COMMeNt <Comment>
WAVeform:COMMeNt <Comment>
REPository:COMMeNt <Comment>

Adds a description to the selected repository element.

Parameters:

<Comment> string

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["Comment"](#) on page 44

ANTenna:REMOve <Remove>
DSRC:REMOve <Remove>
EMITter:REMOve <Remove>
RECeiver:REMOve <Remove>
GENerator:REMOve <Remove>
IPM:REMOve <Remove>
PLUGin:REMOve <Remove>
PULSe:REMOve <Remove>
SCAN:REMOve <Remove>
SCENario:REMOve <Remove>
SEQuence:REMOve <Remove>

WAVeform:REMOve <Remove>

REPository:REMOve <File>[,<Username>,<Passwd>]

Removes the selected element from the workspace.

The element must not reference any child elements. Remove the referenced elements first.

Setting parameters:

<File>	string	Element name, as defined with the . . . :CREate or . . . :NAME command.
<Username>	string	Required if the repository is password protected
<Passwd>	string	Required if the repository is password protected

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["General Repository Settings"](#) on page 43

ASSignment:GENerator:PATH:EMITter:ADD
ASSignment:GENerator:PATH:ANTenna:ADD
EMITter:MODE:ADD
EMITter:MODE:BEAM:ADD
RECeiver:ANTenna:ADD
PULSe:MOP:AMSTep:ADD
PULSe:MOP:FMSTep:ADD
PULSe:MOP:PCHirp:ADD
PULSe:MOP:PLIST:ADD
SCENario:LOCalized:ADD
SCENario:DIRection:ADD
SCENario:CSEQuence:ADD
SCENario:CEMit:ADD
DSRC:ITEM:ADD
IPM:LIST:ITEM:ADD
PULSe:ENVelope:DATA:ITEM:ADD
SEQuence:ITEM:IPM:ADD
SEQuence:ITEM:ADD

Appends new item.

Example: see [Example "Handling items"](#) on page 325

Usage: Event

Manual operation: See ["New, Insert, Append, Remove, Clear Items, Undo/Redo"](#) on page 116

INSTrument:COUNT?
EMITter:MODE:BEAM:COUNT?
EMITter:MODE:COUNT?
PULSe:ENVELOpe:DATA:ITEM:COUNT?
PULSe:MOP:AMSTep:COUNT?
PULSe:MOP:FMSTep:COUNT?
PULSe:MOP:PCHirp:COUNT?
PULSe:MOP:PLIST:COUNT?
IPM:LIST:ITEM:COUNT?
SEQuence:ITEM:IPM:COUNT?
SEQuence:ITEM:COUNT?

Queries the number of existing items.

Return values:

<Count> integer
 *RST: 0

Example: see [Example "Handling items"](#) on page 325

Usage: Query only

Manual operation: See ["Select, No."](#) on page 116

SCENario:LOCalized:SElect <Select>
SCENario:DIRection:SElect <Select>
SCENario:CSEquence:SElect <Select>
SCENario:CEMit:SElect <Select>
EMITter:MODE:BEAM:SElect <Select>
EMITter:MODE:SElect <Select>
RECEiver:ANTenna:SElect <Select>
PULSe:ENVELOpe:DATA:ITEM:SElect <Select>
PULSe:MOP:AMSTep:SElect <Select>
PULSe:MOP:FMSTep:SElect <Select>
PULSe:MOP:PCHirp:SElect <Select>
PULSe:MOP:PLIST:SElect <Select>
DSRC:ITEM:SElect <Select>
IPM:LIST:ITEM:SElect <Select>
SEQuence:ITEM:IPM:SElect <Select>
SEQuence:ITEM:SElect <Select>

Selects the item to which the subsequent commands apply.

Parameters:

<Select> float
 Item number within the range 1 to . . . :COUNT.
 For example, [SEQuence:ITEM:COUNT?](#).
 Range: 1 to 4096

Example: See [Example "Handling items"](#) on page 325

Manual operation: See ["Select, No."](#) on page 116

INSTrument:SElect <Select>
DASHboard:USED:SElect <Select>
DASHboard:UNUSed:SElect <Select>
DASHboard:UNUSed:PATH:SElect <Select>
DASHboard:USED:PATH:SElect <Select>
ASSignment:ANTennas:SElect <Select>
ASSignment:EMITters:SElect <Select>
ASSignment:GENerator:SElect <Select>
ASSignment:GENerator:PATH:SElect <Select>
ASSignment:GENerator:PATH:EMITter:SElect <Select>
ASSignment:GENerator:PATH:ANTenna:SElect <Select>

Selects the element to which the subsequent commands apply.

Parameters:

<Select> string
 Available element as queried with the corresponding . . . :LIST
 command.
 For example, `ASSignment:GENerator:PATH:ANTenna:
 LIST?`

Example: See [Example "Receiver signals to generator mapping in direction finding scenario"](#) on page 363.

Manual operation: See ["Add, Remove, Clear"](#) on page 278

PULSe:MOP:AMSTep:INSert <Insert>
PULSe:MOP:FMSTep:INSert <Insert>
PULSe:MOP:PCHirp:INSert <Insert>
PULSe:MOP:PLISt:INSert <Insert>

Inserts a new item before the selected one.

Setting parameters:

<Insert> float
 *RST: 0

Usage: Setting only

Manual operation: See ["Custom Phase"](#) on page 96

INSTrument:DELeTe <Delete>
ASSignment:GENerator:PATH:EMITter:DELeTe
ASSignment:GENerator:PATH:ANTenna:DELeTe
DSRC:ITEM:DELeTe <Delete>
EMITter:MODE:BEAM:DELeTe <Delete>
EMITter:MODE:DELeTe <Delete>
RECeiver:ANTenna:DELeTe <Delete>
IPM:LIST:ITEM:DELeTe <Delete>
PULSe:ENVELOpe:DATA:ITEM:DELeTe <Delete>
PULSe:MOP:AMSTep:DELeTe <Delete>

PULSe:MOP:FMSTep:DELeTe <Delete>
PULSe:MOP:PCHirp:DELeTe <Delete>
PULSe:MOP:PLISt:DELeTe <Delete>
SCENario:LOCAlized:DELeTe <Delete>
SCENario:DIRectioN:DELeTe <Delete>
SCENario:CSEquence:DELeTe <Delete>
SCENario:CEMit:DELeTe <Delete>
SEquence:ITEM:IPM:DELeTe <Delete>
SEquence:ITEM:DELeTe <Delete>

Deletes the particular item.

Setting parameters:

<Delete> float
 *RST: 0

Example: see [Example "Handling items"](#) on page 325

Usage: Setting only

Manual operation: See ["New, Insert, Append, Remove, Clear Items, Undo/Redo"](#) on page 116

SCENario:GENerator:CLEar
INSTrument:CLEar
ANTenna:MODEl:USER:CLEar
DSRC:CLEar
EMITter:MODE:ANTenna:CLEar
EMITter:MODE:BEAM:CLEar
EMITter:MODE:CLEar
EMITter:MODE:SCAN:CLEar
RECEiver:ANTenna:CLEar
IPM:LIST:CLEar
PULSe:ENVELOpe:DATA:CLEar
PULSe:MOP:AMSTep:CLEar
PULSe:MOP:FMSTep:CLEar
PULSe:MOP:PCHirp:CLEar
PULSe:MOP:PLISt:CLEar
ASSignment:GENerator:PATH:EMITter:CLEar
ASSignment:GENerator:PATH:ANTenna:CLEar
SCENario:EMITter:CLEar
SCENario:CSEquence:CLEar
SCENario:CEMit:CLEar
SCENario:LOCAlized:CLEar
SCENario:DIRectioN:CLEar
SEquence:ITEM:CLEar

Deletes all items from the list or the table.

Example: see [Example "Handling items"](#) on page 325

Usage: Event

Manual operation: See "[New, Insert, Append, Remove, Clear Items, Undo/Redo](#)" on page 116

23.5 Antenna Pattern Commands

Example: Configuring antenna patterns

```
SCPI
ANTenna:CREate "My_PlanarAntenna"
ANTenna:COMMeNt "planar phased antenna with parabolic aperture distribution"
ANTenna:MODEl:TYPE ARRAy
ANTenna:MODEl:FREQuency 1e+10
ANTenna:MODEl:BANDwidth 1e+9
ANTenna:MODEl:POLarization HORizontal
ANTenna:MODEl:ARRay:DISTRibution PARabolic
ANTenna:MODEl:ARRay:NX 12
ANTenna:MODEl:ARRay:NZ 12
ANTenna:MODEl:ARRay:XDISTance 0.6
ANTenna:MODEl:ARRay:ZDISTance 0.3
ANTenna:MODEl:ARRay:ELEMent:COSSine 0
ANTenna:MODEl:ARRay:PEDestal 0.1

ANTenna:CREate "Test"
ANTenna:MODEl:TYPE HORN
ANTenna:MODEl:ROTation:Z 0
ANTenna:MODEl:ROTation:X 0
ANTenna:MODEl:FREQuency 1e+10
ANTenna:MODEl:POLarization HORizontal
ANTenna:MODEl:HORN:LX 0.076
ANTenna:MODEl:HORN:LZ 0.05

ANTenna:CREate "My_PencilBeam"
ANTenna:MODEl:TYPE SINC
ANTenna:MODEl:ROTation:Z 0
ANTenna:MODEl:ROTation:X 0
ANTenna:MODEl:POLarization HORizontal
ANTenna:MODEl:SINC:HPBW 3

ANTenna:CREate "Test CSC"
ANTenna:MODEl:TYPE COSecant
ANTenna:MODEl:POLarization HORizontal
ANTenna:MODEl:COSecant:HPBW 2
ANTenna:MODEl:COSecant:T1 5
ANTenna:MODEl:COSecant:T2 70

ANTenna:CREate "User"
ANTenna:MODEl:TYPE USER
ANTenna:MODEl:USER:LOAD "C:/_PS_files/antenna.ant_pat"
ANTenna:MODEl:ROTation:Z -90
ANTenna:MODEl:ROTation:X 0
ANTenna:MODEl:POLarization HORizontal
ANTenna:MODEl:USER:CLEAR
```

```

ANTenna:CREate "Custom"
ANTenna:MODEl:TYPE CUSTom
ANTenna:MODEl:POLarization HORIZontal
ANTenna:MODEl:CUSTom:HPBW:XY 10
ANTenna:MODEl:CUSTom:HPBW:YZ 3
ANTenna:MODEl:CUSTom:SLStart 30
ANTenna:MODEl:CUSTom:SLRolloff 10
ANTenna:MODEl:CUSTom:SLScale 0.5
ANTenna:MODEl:BACKlobe:ENABLE 1
ANTenna:MODEl:BACKlobe:TYPE MIRROR
ANTenna:MODEl:BACKlobe:ATTenuation 40
    
```

```

ANTenna:NAME "Custom Phased Array"
ANTenna:MODEl:TYPE CARRAY
ANTenna:MODEl:POLarization VERTical
ANTenna:MODEl:CARRay:GEOMetry RECTANGULAR
ANTenna:MODEl:CARRay:RECTangular:NX 25
ANTenna:MODEl:CARRay:RECTangular:NZ 25
ANTenna:MODEl:CARRay:RECTangular:XDISTance 0.015
ANTenna:MODEl:CARRay:RECTangular:ZDISTance 0.015
ANTenna:MODEl:CARRay:ELEMent:COSine 1
ANTenna:MODEl:CARRay:RECTangular:LATTice RECTANGULAR
ANTenna:MODEl:CARRay:DISTRibution COSN
ANTenna:MODEl:CARRay:COSN 4
    
```

ANTenna:MODEl:TYPE.....	339
ANTenna:MODEl:POLarization.....	339
ANTenna:MODEl:CARRay:COSN.....	340
ANTenna:MODEl:ARRay:COSN.....	340
ANTenna:MODEl:CARRay:DISTRibution.....	340
ANTenna:MODEl:ARRay:DISTRibution.....	340
ANTenna:MODEl:ARRay:ELEMent:COSine.....	340
ANTenna:MODEl:CARRay:ELEMent:COSine.....	340
ANTenna:MODEl:CARRay:LINear:N.....	340
ANTenna:MODEl:CARRay:HEXagonal:N.....	340
ANTenna:MODEl:CARRay:RECTangular:NX.....	340
ANTenna:MODEl:CARRay:RECTangular:NZ.....	340
ANTenna:MODEl:ARRay:NX.....	340
ANTenna:MODEl:ARRay:NZ.....	340
ANTenna:MODEl:CARRay:PEDestal.....	341
ANTenna:MODEl:ARRay:PEDestal.....	341
ANTenna:MODEl:CARRay:CIRCular:DISTance.....	341
ANTenna:MODEl:CARRay:HEXagonal:DISTance.....	341
ANTenna:MODEl:CARRay:LINear:DISTance.....	341
ANTenna:MODEl:CARRay:RECTangular:XDISTance.....	341
ANTenna:MODEl:CARRay:RECTangular:ZDISTance.....	341
ANTenna:MODEl:ARRay:XDISTance.....	341
ANTenna:MODEl:ARRay:ZDISTance.....	341
ANTenna:MODEl:BACKlobe:ENABLE.....	341

ANTenna:MODel:BACKlobe:TYPE.....	342
ANTenna:MODel:BACKlobe:ATTenuation.....	342
ANTenna:MODel:COSecant:T1.....	342
ANTenna:MODel:COSecant:T2.....	342
ANTenna:MODel:CARRay:CIRCular:LATTice.....	342
ANTenna:MODel:CARRay:RECTangular:LATTice.....	342
ANTenna:MODel:CARRay:CIRCular:RADIus.....	342
ANTenna:MODel:CARRay:GEOMetry.....	343
ANTenna:MODel:CARRay:ELEMent.....	343
ANTenna:MODel:CUSTom:HPBW:XY.....	343
ANTenna:MODel:CUSTom:HPBW:YZ.....	343
ANTenna:MODel:CUSTom:SLStart.....	344
ANTenna:MODel:CUSTom:SLRolloff.....	344
ANTenna:MODel:CUSTom:SLScale.....	344
ANTenna:MODel:BANDwidth.....	344
ANTenna:MODel:FREQuency.....	344
ANTenna:MODel:GAUSSian:HPBW.....	345
ANTenna:MODel:COSecant:HPBW.....	345
ANTenna:MODel:SINC:HPBW.....	345
ANTenna:MODel:HORN:LX.....	345
ANTenna:MODel:HORN:LZ.....	345
ANTenna:MODel:PARAbolic:DIAMeter.....	345
ANTenna:MODel:ROTation:X.....	346
ANTenna:MODel:ROTation:Z.....	346
ANTenna:MODel:USER:LOAD.....	346
ANTenna:MODel:USER:CSV:FORMat.....	346
PLOT:POLar:TYPE.....	347
PLOT:POLar:CUT.....	347
PLOT:POLar:LOG:MIN.....	347

ANTenna:MODel:TYPE <Type>

Sets the antenna pattern.

Parameters:

<Type> DIPole | PARAbolic | GAUSSian | SINC | HORN | COSecant |
ARRay | USER | CUSTom | CARRay | CARDoid

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Antenna Model"](#) on page 165

ANTenna:MODel:POLarization <Polarization>

Sets the antenna polarisation.

Parameters:

<Polarization> VERTical | HORizontal | CRIGHt | CLEFt | SRIGHt | SLEFt

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Polarization"](#) on page 166

ANTenna:MODEl:CARRay:COsN <Cosn>

ANTenna:MODEl:ARRay:COsN <Cosn>

Sets Cos^N of the Planar Phased Array antenna.

Parameters:

<Cosn> float
Range: 2 to 10

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Planar Phased Array Antenna Settings"](#) on page 171

ANTenna:MODEl:CARRay:DIStRiBution <Distribution>

ANTenna:MODEl:ARRay:DIStRiBution <Distribution>

Sets the aperture distribution of the Planar Phased Array antenna.

Parameters:

<Distribution> UNIFORM | PARabolic | COSine | CSquared | COSN |
TRIangular

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Planar Phased Array Antenna Settings"](#) on page 171

ANTenna:MODEl:ARRay:ELEMent:COsine <Cosine>

ANTenna:MODEl:CARRay:ELEMent:COsine <Cosine>

Sets the characteristic of individual antenna elements.

Parameters:

<Cosine> ON | OFF | 1 | 0
0|OFF
Omnidirectional characteristic
1|ON
Cosine characteristic

Example: See [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Phased Array Settings"](#) on page 173

ANTenna:MODEl:CARRay:LINear:N <N>

ANTenna:MODEl:CARRay:HEXagonal:N <N>

ANTenna:MODEl:CARRay:RECTangular:Nx <Nx>

ANTenna:MODEl:CARRay:RECTangular:Nz <Nz>

ANTenna:MODEl:ARRay:Nx <Nx>

ANTenna:MODEl:ARRay:Nz <Nz>

Sets the number of elements of the antenna array.

Parameters:

<Nz> float
 Range: 2 to 1000 (planar phased array; linear phase array), 100 (rectangular phase array), 50 (hexagonal phase array)

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Planar Phased Array Antenna Settings"](#) on page 171

ANTenna:MODEl:CARRay:PEDestal <Pedestal>

ANTenna:MODEl:ARRay:PEDestal <Pedestal>

Sets the Pedestal of the Planar Phased Array antenna.

Parameters:

<Pedestal> float
 Range: 0 to 1

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Planar Phased Array Antenna Settings"](#) on page 171

ANTenna:MODEl:CARRay:CIRCular:DISTance <Distance>

ANTenna:MODEl:CARRay:HEXagonal:DISTance <Distance>

ANTenna:MODEl:CARRay:LINear:DISTance <Distance>

ANTenna:MODEl:CARRay:RECTangular:XDISTance <Xdistance>

ANTenna:MODEl:CARRay:RECTangular:ZDISTance <Zdistance>

ANTenna:MODEl:ARRay:XDISTance <Xdistance>

ANTenna:MODEl:ARRay:ZDISTance <Zdistance>

Sets the spacing between the elements of the array antenna.

Parameters:

<Zdistance> float
 Range: 0.0001 to 1

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Planar Phased Array Antenna Settings"](#) on page 171

ANTenna:MODEl:BACKlobe:ENABLE <Enable>

Enables the simulation of a back lobe.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Simulate Back Lobe, Attenuation, Type "](#) on page 167

ANTenna:MODEl:BACKlobe:TYPE <Type>

Sets the shape of the back lobe pattern.

Parameters:

<Type> MIRRor | OMNidirect

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See "[Simulate Back Lobe, Attenuation, Type](#)" on page 167

ANTenna:MODEl:BACKlobe:ATTenuation <Attenuation>

Sets the attenuation of the back lobe.

Parameters:

<Attenuation> float
Range: 0 to 100

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See "[Simulate Back Lobe, Attenuation, Type](#)" on page 167

ANTenna:MODEl:COSecant:T1 <T1>**ANTenna:MODEl:COSecant:T2** <T2>

Sets the Theta parameters.

Parameters:

<T2> float
Range: 1 to 90

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See "[Cosecant Squared Antenna Settings](#)" on page 171

ANTenna:MODEl:CARRay:CIRCular:LATTice <Lattice>**ANTenna:MODEl:CARRay:RECTangular:LATTice** <Lattice>

Sets the lattice.

Parameters:

<Lattice> RECTangular | TRIangular

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See "[Custom Phased Array Settings](#)" on page 173

ANTenna:MODEl:CARRay:CIRCular:RADius <Radius>

Set the radius of the circular phased array antenna.

Parameters:

<Radius> float
Range: 1 to 50

Manual operation: See ["Custom Phased Array Settings"](#) on page 173

ANTenna:MODEl:CARRay:GEOMetry <Geometry>

Sets the geometry of the custom phased array antenna.

Parameters:

<Geometry> RECTangular | LINear | HEXagonal | CIRCular

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Phased Array Settings"](#) on page 173

ANTenna:MODEl:CARRay:ELEMent <Element>

Set the state of one or more antenna elements.

Antenna elements are indicated by their subsequent number, as it is displayed in the "Antenna Geometry" dialog ([Figure 10-3](#)).

Parameters:

<Element> numeric list
A comma-separated list of numbers or of range of numbers enclosed in brackets. In the case of a range, the end points are inclusive.
The listed numbers indicate the *active* antenna elements; if a number is omitted, the antenna element is disabled.

Example:

```
ANTenna:MODEl:CARRay:GEOMetry LINEAR
ANTenna:MODEl:CARRay:LINear:N 7
ANTenna:MODEl:CARRay:ELEMent? (1,3:7)
// element 1 and 3 through 7 are active
ANTenna:MODEl:CARRay:ELEMent (1,3,5,7)
// antenna elements 1, 3, 5 and 7 are active
// antenna elements 2, 4 and 6 are deactivated
```

Manual operation: See ["Custom Phased Array Settings"](#) on page 173

ANTenna:MODEl:CUSTom:HPBW:XY <Xy>**ANTenna:MODEl:CUSTom:HPBW:YZ** <Yz>

Sets the required HPBW of the custom antenna.

Parameters:

<Yz> float
Range: 0.1 to 45

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Antenna Settings"](#) on page 172

ANTenna:MODEl:CUSTom:SLStart <Slstart>

Sets the power level of the first pairs of side lobes.

Parameters:

<Slstart> float
Range: 1 to 90

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Antenna Settings"](#) on page 172

ANTenna:MODEl:CUSTom:SLRolloff <Slrolloff>

Sets the factor used to calculate the HPBW of the side lobes.

Parameters:

<Slrolloff> float
Range: 1 to 45

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Antenna Settings"](#) on page 172

ANTenna:MODEl:CUSTom:SLScale <Slscale>

Sets the step size to calculate the power level of the side lobes.

Parameters:

<Slscale> float
Range: 0.01 to 10

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Custom Antenna Settings"](#) on page 172

ANTenna:MODEl:BANDwidth <Bandwidth>

Sets the antenna bandwidth.

Parameters:

<Bandwidth> float
Range: 1e+06 to 1e+11

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Frequency, Bandwidth"](#) on page 166

ANTenna:MODEl:FREQUENCY <Frequency>

Sets the frequency.

Parameters:

<Frequency> float
 Range: 1e+06 to 1e+11
 Default unit: Hz

Example: See [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Frequency, Bandwidth"](#) on page 166

ANTenna:MODEl:GAUSSian:HPBW <Hpbw>

ANTenna:MODEl:COSecant:HPBW <Hpbw>

ANTenna:MODEl:SINC:HPBW <Hpbw>

Sets the Half-Power Beam Width for the Gaussian, Sin(x)/x and Cosecant Squared antennas.

Parameters:

<Hpbw> float
 Range: 0.1 to 30
 Default unit: degree

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Sin\(x\)/x Antenna Settings"](#) on page 170

ANTenna:MODEl:HORN:LX <Lx>

ANTenna:MODEl:HORN:LZ <Lz>

Sets the length of the rectangular sides of the Pyramidal Horn antenna.

Parameters:

<Lz> float
 Range: 0.01 to 100
 Default unit: m

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Pyramidal Horn Antenna Settings"](#) on page 170

ANTenna:MODEl:PARAbolic:DIAMeter <Diameter>

Sets the diameter of the parabolic dish antenna.

Parameters:

<Diameter> float
 Range: 0.05 to 100
 Default unit: m

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See ["Parabolic Antenna Settings"](#) on page 169

ANTenna:MODEl:ROtation:X <X>

ANTenna:MODEl:ROtation:Z <Z>

Sets the X and Z antenna rotation.

Parameters:

<Z> float
 Range: -180 to 180
 Default unit: degree

Example: [Example "Configuring antenna patterns"](#) on page 337

Manual operation: See "[Z-Rotation, X-Rotation](#)" on page 165

ANTenna:MODEl:USER:LOAD <Load>

Loads a custom antenna pattern file.

Setting parameters:

<Load> string

Example: [Example "Configuring antenna patterns"](#) on page 337

Usage: Setting only

Manual operation: See "[Load/Delete](#)" on page 177

ANTenna:MODEl:USER:CSV:FORMat

<start_row>,<col_sep>,<decimal>,<col>{,<col>}

Defines how the data in the selected * .csv file should be interpreted.

The settings in this command are not permanent. The command affects only the currently selected antenna.

For description of the * .csv file format, see [Chapter A.2, "Antenna Pattern File Formats"](#), on page 485.

Parameters:

<col_sep> COMMA | SEMICOLON | WHITESPACE
 Column separator

<decimal> POINT | COMMA
 Decimal delimiter

<col> THETA | PHI | GAIN | ETHRE | ETHIM | EPHRE | EPHIM
 Data content of the column: Theta, Phi, Gain, Re(E_Theta),
 Im(E_Theta), Re(E_Phi), Im(E_Phi)
 At least one out of maximum 6 columns should exist

Setting parameters:

<start_row> First row with data
 Range: 1 to max

Example:
ANTenna:MODeL:TYPE USER
ANTenna:MODeL:USER:CSV:FORMat "2","COMMA","POINT","PHI","THETA","GAIN"
ANTenna:MODeL:USER:LOAD "D:/PS/ansys.csv"

Usage: Setting only

Manual operation: See ["Import Wizard"](#) on page 178

PLOT:POLar:TYPE <Type>

Sets the coordinates of the 2D antenna pattern diagram.

Setting parameters:

<Type> POLar | CARTesian

Example:
PLOT:POLar:TYPE CARTesian
PLOT:POLar:CUT XY
PLOT:POLar:LOG:MIN -90

Usage: Setting only

Manual operation: See ["2D"](#) on page 168

PLOT:POLar:CUT <Cut>

Sets the diagram cut.

Setting parameters:

<Cut> XY | YZ

Example: see [PLOT:POLar:TYPE](#) on page 347

Usage: Setting only

Manual operation: See ["2D"](#) on page 168

PLOT:POLar:LOG:MIN <Min>

Sets the minimum value displayed on the y axis.

Setting parameters:

<Min> float

Example: see [PLOT:POLar:TYPE](#) on page 347

Usage: Setting only

Manual operation: See ["2D"](#) on page 168

23.6 Antenna Scan Commands

Example: Defining antenna scans

```

SCPI
SCAN:CREate "My_RasterScan"
SCAN:TYPE RASter
SCAN:RASter:WIDTh 70
SCAN:RASter:BARWidth 3
SCAN:RASter:RATE 15
SCAN:RASter:BARs 10
SCAN:RASter:DIRection HORizontal
SCAN:RASter:RETRace 0.001
SCAN:RASter:UNIDirection 1
SCAN:RASter:FLYBack 0.001
SCAN:RASter:PALMer 1
SCAN:RASter:PRATe 5
SCAN:RASter:PSQuint 1.5

SCAN:CREate "Test Antenna Scan"
SCAN:TYPE HELical
SCAN:HELical:RPM 45
SCAN:HELical:TURNs 8
SCAN:HELical:ELEVation:STEP 3
SCAN:HELical:RETRace 1
SCAN:HELical:ROTation CW

```

SCAN:CIRCular:ROTation.....	349
SCAN:CONical:ROTation.....	349
SCAN:HELical:ROTation.....	349
SCAN:LSW:ROTation.....	349
SCAN:SPIRal:ROTation.....	349
SCAN:CIRCular:RPM.....	349
SCAN:HELical:RPM.....	349
SCAN:CONical:RATE.....	350
SCAN:SECTor:RATE.....	350
SCAN:RASter:RATE.....	350
SCAN:HELical:ELEVation:STEP.....	350
SCAN:HELical:TURNs.....	350
SCAN:LSW:DIRection.....	350
SCAN:LSW:DWELI.....	351
SCAN:LSW:LOBes.....	351
SCAN:CONical:SQUint.....	351
SCAN:LSW:SQUint.....	351
SCAN:RASter:BARs.....	351
SCAN:RASter:BARWidth.....	351
SCAN:RASter:FLYBack.....	352
SCAN:SECTor:FLYBack.....	352
SCAN:HELical:RETRace.....	352

SCAN:RASTer:RETRace.....	352
SCAN:SPIRAl:RETRace.....	352
SCAN:SPIRAl:UNIDirection.....	352
SCAN:RASTer:UNIDirection.....	352
SCAN:SECTor:UNIDirection.....	352
SCAN:RASTer:WIDTh.....	353
SCAN:SECTor:WIDTh.....	353
SCAN:RASTer:DIRection.....	353
SCAN:SPIRAl:ROUNds.....	353
SCAN:SPIRAl:RTIME.....	353
SCAN:SPIRAl:STEP.....	353
SCAN:TYPE.....	354
SCAN:CIRCUlar:NODDing.....	354
SCAN:SECTor:NODDing.....	354
SCAN:CIRCUlar:NElevation.....	354
SCAN:SECTor:NElevation.....	354
SCAN:CIRCUlar:NRATe.....	354
SCAN:SECTor:NRATe.....	354
SCAN:CIRCUlar:PALMer.....	355
SCAN:RASTer:PALMer.....	355
SCAN:SECTor:PALMer.....	355
SCAN:CIRCUlar:PRATe.....	355
SCAN:RASTer:PRATe.....	355
SCAN:SECTor:PRATe.....	355
SCAN:CIRCUlar:PSQuint.....	355
SCAN:RASTer:PSQuint.....	355
SCAN:SECTor:PSQuint.....	355

SCAN:CIRCUlar:ROTation <Rotation>

SCAN:CONical:ROTation <Rotation>

SCAN:HELical:ROTation <Rotation>

SCAN:LSW:ROTation <Rotation>

SCAN:SPIRAl:ROTation <Rotation>

Sets the rotation direction of the antenna.

Parameters:

<Rotation> CW | CCW

Example: see [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Spiral Scan Settings"](#) on page 184

SCAN:CIRCUlar:RPM <Rpm>

SCAN:HELical:RPM <Rpm>

Sets the rotation speed of the antenna.

Parameters:

<Rpm> float

Range: 0.01 to 1000

Default unit: degree/s

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Helical Scan Settings"](#) on page 184

SCAN:CONical:RATE <Rate>

SCAN:SECTor:RATE <Rate>

SCAN:RASTer:RATE <Rate>

Sets the turning speed.

Parameters:

<Rate> float
Range: 0.01 to 1000
Default unit: degree/s

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Raster Scan Settings"](#) on page 182

SCAN:HELical:ELEVation:STEP <Step>

Sets the step width with that the antenna changes its elevation.

Parameters:

<Step> float
Range: 0.01 to 11.25
Default unit: degree

Example: See [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Helical Scan Settings"](#) on page 184

SCAN:HELical:TURNs <Turns>

Sets the number of turns.

Parameters:

<Turns> float
Range: 1 to 30

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Helical Scan Settings"](#) on page 184

SCAN:LSW:DIRection <Direction>

Sets the horizontal or vertical switching direction.

Parameters:

<Direction> H | V

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Lobe Switching Scan Settings"](#) on page 185

SCAN:LSW:DWELI <Dwell>

Sets the speed with that the antenna switches between the lobes.

Parameters:

<Dwell> float
Range: 1e-06 to 1

Example: See [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Lobe Switching Scan Settings"](#) on page 185

SCAN:LSW:LOBes <Lobes>

Set the number of lobes.

Parameters:

<Lobes> 2 | 4

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Lobe Switching Scan Settings"](#) on page 185

SCAN:CONical:SQUint <Squint>**SCAN:LSW:SQUint** <Squint>

Sets the offset angle of the antenna beam, that means for the conical antenna the parameter sets the radius of the scanned circle.

Parameters:

<Squint> float
Range: 0.05 to 15
Default unit: degree

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Lobe Switching Scan Settings"](#) on page 185

SCAN:RASTer:BARs <Bars>

Sets the number of scanned bars (sectors).

Parameters:

<Bars> float
Range: 1 to 30

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Raster Scan Settings"](#) on page 182

SCAN:RASTer:BARWidth <Barwidth>

Sets the distance between two consecutive scanned bars (sectors).

Parameters:

<Barwidth> float
 Range: 0.1 to 9
 Default unit: m

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Raster Scan Settings"](#) on page 182

SCAN:RASTer:FLYBack <Flyback>

SCAN:SECTor:FLYBack <Flyback>

Sets the Flyback time for the antenna working in unidirectional mode.

Parameters:

<Flyback> float
 Range: 0 to 1
 Default unit: s

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Sector Scan Settings"](#) on page 181

SCAN:HELical:RETRace <Retrace>

SCAN:RASTer:RETRace <Retrace>

SCAN:SPIRAl:RETRace <Retrace>

Sets the speed for the antenna to return to the initial orientation.

Parameters:

<Retrace> float
 Range: 0 to 1

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Spiral Scan Settings"](#) on page 184

SCAN:SPIRAl:UNIDirection <Unidirection>

SCAN:RASTer:UNIDirection <Unidirection>

SCAN:SECTor:UNIDirection <Unidirection>

Enables a unidirectional scan mode.

Parameters:

<Unidirection> ON | OFF | 1 | 0

Example: See [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Sector Scan Settings"](#) on page 181

SCAN:RASTer:WIDTh <Width>**SCAN:SECTor:WIDTh** <Width>

Sets the width of the sector to be scanned.

Parameters:

<Width> float
Range: 0.1 to 180
Default unit: degree

Example: [Example "Defining antenna scans"](#) on page 348**Manual operation:** See ["Sector Scan Settings"](#) on page 181

SCAN:RASTer:DIRectioN <Direction>

Sets the scanning direction.

Parameters:

<Direction> HORizontal | VERTical

Manual operation: See ["Raster Scan Settings"](#) on page 182see [Example "Defining antenna scans"](#) on page 348

SCAN:SPIRAl:ROUNds <Rounds>

Sets the number of rounds the antenna performs.

Parameters:

<Rounds> float
Range: 0.1 to 15

Example: [Example "Defining antenna scans"](#) on page 348**Manual operation:** See ["Spiral Scan Settings"](#) on page 184

SCAN:SPIRAl:RTIME <Rtime>

Sets the turning speed of the antenna.

Parameters:

<Rtime> float
Range: 0.01 to 10
Default unit: degree/s

Example: [Example "Defining antenna scans"](#) on page 348**Manual operation:** See ["Spiral Scan Settings"](#) on page 184

SCAN:SPIRAl:STEP <Step>

Determines the step size to increase the scan radius.

Parameters:

<Step> float
 Range: 1 to 11.25
 Default unit: degree

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Spiral Scan Settings"](#) on page 184

SCAN:TYPE <Type>

Sets the scan type.

Parameters:

<Type> CIRCular | SECTor | RASTer | CONical | HELical | SPIRal | LSW

Example: [Example "Defining antenna scans"](#) on page 348

Manual operation: See ["Scan Type"](#) on page 180

SCAN:CIRCular:NODDing <Nodding>**SCAN:SECTor:NODDing** <Nodding>

Enables superimposing a horizontal nodding on the scan.

Parameters:

<Nodding> ON | OFF | 1 | 0

Example:

```
SCAN:SECTor:NODDing 1
SCAN:SECTor:NRATe 500
SCAN:SECTor:NELevation 15
```

Manual operation: See ["Sector Scan Settings"](#) on page 181

SCAN:CIRCular:NELevation <Nelevation>**SCAN:SECTor:NELevation** <Nelevation>

Sets the elevation angle.

Parameters:

<Nelevation> float
 Range: 0.01 to 90

Example: see [SCAN:SECTor:NODDing](#) on page 354

Manual operation: See ["Sector Scan Settings"](#) on page 181

SCAN:CIRCular:NRATe <Nrate>**SCAN:SECTor:NRATe** <Nrate>

Sets the elevation rate.

Parameters:

<Nrate> float
Range: 0.01 to 2000

Example: see [SCAN:SECTor:NODDing](#) on page 354

Manual operation: See "[Sector Scan Settings](#)" on page 181

SCAN:CIRCular:PALMer <Palmer>

SCAN:RASTer:PALMer <Palmer>

SCAN:SECTor:PALMer <Palmer>

Enables superimposing a conical scan on the current scan.

Parameters:

<Palmer> ON | OFF | 1 | 0

Manual operation: See "[Sector Scan Settings](#)" on page 181

see [Example "Defining antenna scans"](#) on page 348

SCAN:CIRCular:PRATe <Prate>

SCAN:RASTer:PRATe <Prate>

SCAN:SECTor:PRATe <Prate>

Sets the scan rate.

Parameters:

<Prate> float
Range: 0.1 to 1000

Manual operation: See "[Sector Scan Settings](#)" on page 181

see [Example "Defining antenna scans"](#) on page 348

SCAN:CIRCular:PSQuint <Psquint>

SCAN:RASTer:PSQuint <Psquint>

SCAN:SECTor:PSQuint <Psquint>

Sets the squint angle.

Parameters:

<Psquint> float
Range: 0.05 to 15

Manual operation: See "[Sector Scan Settings](#)" on page 181

see [Example "Defining antenna scans"](#) on page 348

23.7 Dashboard Commands

Example: Configuring connected instruments

```

SCPI
INSTRument:COUNT?
// 3

SCENario:NAME "DF"
SCENario:CALCulate
SCENario:NAME "LocalizedEmitters"
SCENario:CALCulate
DASHboard:SCENario:CURRent:NAME?
// LocalizedEmitters
DASHboard:SCENario:CURRent:STATe?
// Calculation has failed
DASHboard:SCENario:DEPLoyed:NAME?
// DF
DASHboard:SCENario:DEPLoyed:TIME?
// 10 Dec 2015 08:38:34

DASHboard:REFresh
INSTRument:LIST?
// 'SMW 200A, 0','SMW 200A, 100025','SMBV 100A, 100012'
DASHboard:UNUSed:ALL 1
DASHboard:UNUSed:LIST?
// 'SMW 200A, 100025','SMBV 100A, 100012'

DASHboard:USED:LIST?
// 'SMW 200A, 0'
DASHboard:USED:SElect 'SMW 200A, 0'
DASHboard:USED:PATH:LIST?
// 'RF A', 'RF B'
DASHboard:USED:PATH:SElect 'RF B'
DASHboard:USED:FREQuency 1e+09
DASHboard:USED:LEVel -30
DASHboard:USED:STATe 1

DASHboard:MUTE
DASHboard:USED:SElect 'SMW 200A, 0'
DASHboard:USED:STATe?
// 0

DASHboard:REFresh..... 357
DASHboard:MUTE..... 357
DASHboard:SCENario:DEPLoyed:NAME?..... 357
DASHboard:SCENario:CURRent:NAME?..... 357
DASHboard:SCENario:CURRent:STATus?..... 357
DASHboard:SCENario:DEPLoyed:TIME?..... 358

```

DASHboard:UNUSed:ALL.....	358
DASHboard:UNUSed:LIST?.....	358
DASHboard:USED:LIST?.....	358
DASHboard:UNUSed:PATH:LIST.....	358
DASHboard:USED:PATH:LIST?.....	358
DASHboard:UNUSed:FREQUency.....	359
DASHboard:USED:FREQUency.....	359
DASHboard:UNUSed:LEVel.....	359
DASHboard:USED:LEVel.....	359
DASHboard:UNUSed:STATe.....	359
DASHboard:USED:STATe.....	359

DASHboard:REFresh

Refreshes the displayed information.

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Event

Manual operation: See ["Refresh"](#) on page 283

DASHboard:MUTE

Deactivates the RF outputs of all signal generators.

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Event

Manual operation: See ["Mute All"](#) on page 283

DASHboard:SCENario:DEPLoyed:NAME?

DASHboard:SCENario:CURRent:NAME?

Queries the name of the current/last deployed scenario.

Return values:

<Name> string

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Query only

Manual operation: See ["Current scenario"](#) on page 282

DASHboard:SCENario:CURRent:STATus?

Queries the scenario status.

Return values:

<Status> string

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Query only
Manual operation: See ["Scenario status"](#) on page 282

DASHboard:SCENario:DEPLoyed:TIME?

Queries the date and time the scenario has been deployed.

Return values:
 <Time> <DD MMM YYYY hh:mm:ss>

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Query only

Manual operation: See ["Deployment timestamp"](#) on page 282

DASHboard:UNUSed:ALL <All>

Shows all unused signal generators.

Parameters:
 <All> ON | OFF | 1 | 0

Example: See [Example "Configuring connected instruments"](#) on page 356.

Manual operation: See ["Show All"](#) on page 283

DASHboard:UNUSed:LIST?**DASHboard:USED:LIST?**

Queries the names of the used/unused signal generators.

Return values:
 <List> "<Instr#1>","<Instr#2>",...

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Query only

Manual operation: See ["Generators in use, Unused generators"](#) on page 283

DASHboard:UNUSed:PATH:LIST <List>**DASHboard:USED:PATH:LIST?**

Queries the available RF paths of the selected signal generators.

Return values:
 <List> "<Path#1>","<Paht#2>",...

Example: See [Example "Configuring connected instruments"](#) on page 356.

Usage: Query only

Manual operation: See ["Generators in use, Unused generators"](#) on page 283

DASHboard:UNUSed:FREQuency <Frequency>

DASHboard:USED:FREQuency <Frequency>

Sets the RF frequency.

Parameters:

<Frequency> float

Example: See [Example "Configuring connected instruments"](#) on page 356.

Manual operation: See ["Generators in use, Unused generators"](#) on page 283

DASHboard:UNUSed:LEVel <Level>

DASHboard:USED:LEVel <Level>

Sets the RF level.

Parameters:

<Level> float

Example: See [Example "Configuring connected instruments"](#) on page 356.

Manual operation: See ["Generators in use, Unused generators"](#) on page 283

DASHboard:UNUSed:STATe <State>

DASHboard:USED:STATe <State>

Activates the RF output of the selected signal generator.

Parameters:

<State> ON | OFF | 1 | 0

Example: See [Example "Configuring connected instruments"](#) on page 356.

Manual operation: See ["Generators in use, Unused generators"](#) on page 283

23.8 Data Source Commands

Example: Creating data source

```

SCPI
DSRC:CREate "DataSource"
DSRC:SElect "DataSource"
DSRC:ITEM:ADD
DSRC:ITEM:SElect 1
DSRC:ITEM:TYPE PATtern
DSRC:ITEM:PATtern ONE
DSRC:ITEM:BITS 2

DSRC:ITEM:ADD
DSRC:ITEM:SElect 2
DSRC:ITEM:TYPE PRBS
DSRC:ITEM:PRBS:TYPE P15
DSRC:ITEM:PRBS:INIT 1
DSRC:ITEM:PRBS:INIT:VALue 10
DSRC:ITEM:BITS 12

```

DSRC:ITEM:BITS.....	360
DSRC:ITEM:DATA.....	360
DSRC:ITEM:PATtern.....	361
DSRC:ITEM:PRBS:INIT.....	361
DSRC:ITEM:PRBS:INIT:VALue.....	361
DSRC:ITEM:PRBS:TYPE.....	361
DSRC:ITEM:TYPE.....	362

DSRC:ITEM:BITS <Bits>

Sets the length of the selected item in bits.

Parameters:

<Bits> float
 Range: 0 to 4096

Example: see [Example "Creating data source"](#) on page 360

Manual operation: See ["Data Source Table"](#) on page 294

DSRC:ITEM:DATA <Data>

Sets the user defined data pattern.

Parameters:

<Data> string

Example:

```

DSRC:ITEM:TYPE USER
DSRC:ITEM:DATA?
// 0101 1010    /binary
// "Test"       /string
// #x 20 20 20  /hex numbers

```

Manual operation: See ["User Data"](#) on page 295

DSRC:ITEM:PATtern <Pattern>

Sets the data pattern of the selected item.

Parameters:

<Pattern> ZERO | ONE | ALT | R3 | R4A | R4B | R5 | R7 | R11 | R13

ZERO|ONE

Binary 0 and 1

ALT

Variable bit strings ("1010") with alternating 0 and 1 and a maximum length of 999 bits

R3|R4A|R4B|R5|R7|R11|R13

Barker codes

Example: see [Example "Creating data source"](#) on page 360

Manual operation: See ["Data Source Table"](#) on page 294

DSRC:ITEM:PRBS:INIT <Init>

Enables/disables initialization of the shift register with a user-defined value.

Parameters:

<Init> ON | OFF

Example: see [Example "Creating data source"](#) on page 360

Manual operation: See ["Shift register"](#) on page 294

DSRC:ITEM:PRBS:INIT:VALue <Value>

Set a new initialization value.

Parameters:

<Value> float
Range: 1 to 511

Example: see [Example "Creating data source"](#) on page 360

Manual operation: See ["Shift register"](#) on page 294

DSRC:ITEM:PRBS:TYPE <Type>

Sets the PRBS type for the selected item.

Parameters:

<Type> P9 | P11 | P15 | P16 | P20 | P21 | P23 | P7

Example:see [Example "Creating data source"](#) on page 360**Manual operation:**See ["Data Source Table"](#) on page 294**DSRC:ITEM:TYPE <Type>**

Sets the data type of selected item.

Parameters:

<Type> PATtern | PRBS | USER

Example:see [Example "Creating data source"](#) on page 360**Manual operation:**See ["Data Source Table"](#) on page 294

23.9 Signal to Generator Mapping Commands

Example: Performing signal to generator mapping in 2D scenario

```

SCPI
ASSignment:GENerator:LIST?
// "My_RS_SMW", "My_RS_SMBV", "RS_SMW", "My_SGT", "SMW"

ASSignment:GENerator:SElect "My_RS_SMW"
ASSignment:GENerator:MASTer "My_RS_SMW"
ASSignment:GENerator:CAPabilities?
// 1K76|1B106|1B206|2K300|2K301|1K308|2K511|2K512|2K522|
// 2B10|1B13T|#0|6e+09|0|0|
ASSignment:GENerator:PATH:LIST?
// "RF A", "RF B", "BBMM 1"
ASSignment:GENerator:PATH:SElect "RF A"
ASSignment:GENerator:PATH:EMITter:LIST?
// "E1", "E2"
ASSignment:EMITters:LIST?
// "I1", "Bkg. Emitter 1"

ASSignment:EMITters:SElect "Bkg. Emitter 1"
ASSignment:GENerator:PATH:SElect "BBMM 1"
ASSignment:GENerator:PATH:EMITter:ADD
ASSignment:GENerator:PATH:EMITter:LIST?
// "Bkg. Emitter 1"

ASSignment:EMITters:LIST?
// "I1"

```

Example: Receiver signals to generator mapping in direction finding scenario

```

SCPI
ASSignment:GENerator:LIST?
// "My_RS_SMW","My_RS_SMBV","RS_SMW","My_SGT","SMW"
ASSignment:GENerator:SElect "My_RS_SMW"

ASSignment:GENerator:PATH:LIST?
// "RF A","RF B","I/Q OUT 2"
ASSignment:GENerator:PATH:SElect "RF A"
ASSignment:GENerator:PATH:ANTenna:LIST?
// "Antenna 1 - TestEmitter","Antenna 2 - TestEmitter"
ASSignment:GENerator:PATH:SElect "RF B"
ASSignment:GENerator:PATH:ANTenna:LIST?
// "Antenna 3 - TestEmitter"
ASSignment:ANTennas:LIST?
// "Antenna 4 - TestEmitter"
ASSignment:ANTennas:SElect "Antenna 4 - TestEmitter"
ASSignment:GENerator:PATH:ANTenna:ADD
ASSignment:GENerator:PATH:ANTenna:LIST?
// "Antenna 3 - TestEmitter","Antenna 4 - TestEmitter"
ASSignment:GENerator:PATH:ANTenna:SElect "Antenna 3 - TestEmitter"
ASSignment:GENerator:PATH:ANTenna:DElete
ASSignment:GENerator:PATH:ANTenna:LIST?
// "Antenna 4 - TestEmitter"

```

ASSignment:EMITters:LIST?	363
ASSignment:ANTennas:LIST?	363
ASSignment:GENerator:MASTER	364
ASSignment:GENerator:LIST?	364
ASSignment:GENerator:CAPabilities?	364
ASSignment:GENerator:PATH:LIST?	365
ASSignment:GENerator:PATH:ANTenna:LIST?	365
ASSignment:GENerator:PATH:EMITter:LIST?	365

ASSignment:EMITters:LIST?

Queries the alias names of the unassigned emitters/interfereres.

Return values:

<List> "<Emitter/Inter#1>",<Emitter/Inter#2>",...

Example: see [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362

Usage: Query only

Manual operation: See ["Emitters/Interferers"](#) on page 277

ASSignment:ANTennas:LIST?

Queries the alias names of the unassigned receiver signals.

Return values:

<List> "<ReceiverSignal#1>",<ReceiverSignal#2>",...

Example:

See [Example "Receiver signals to generator mapping in direction finding scenario"](#) on page 363.

Usage:

Query only

Manual operation:

See ["Receiver Signals"](#) on page 277

ASSignment:GENerator:MASTer <Master>

Sets the master generator profile.

Parameters:

<Master> "<GenProfile>"

To query the list of the available generator profiles, use the command [ASSignment:GENerator:LIST?](#)

Example:

see [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362

Manual operation:

See ["Generator Profiles"](#) on page 277

ASSignment:GENerator:LIST?

Queries a list of the available generator profiles.

Return values:

<List> "<GenProfile#1>",<GenProfile#2>",...

Example:

see [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362

Usage:

Query only

Manual operation:

See ["Generator Profiles"](#) on page 277

ASSignment:GENerator:CAPabilities?

Queries the capabilities of the selected generator profile.

Return values:

<Capabilities> <RFA>, <BWA>, <MEMA>, <SWA>, [<RFB>, <BWB>, <MEMB>, <SWB>]

String that lists the maximum RF and BW, available memory size, and installed options per path.

Example:

See [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362

Usage:

Query only

Manual operation:

See ["Generator Profiles"](#) on page 277

ASSignment:GENerator:PATH:LIST?

Queries the available paths.

Return values:

<List> "<Path#1>","<Path#2>",...
List of available paths.

Example: See [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362

Usage: Query only

Manual operation: See ["Generator Profiles"](#) on page 277

ASSignment:GENerator:PATH:ANTenna:LIST?

Queries the list of assigned receiver signals to the selected path.

Return values:

<List> "<ReceiverSignal#1>","<ReceiverSignal#2>",...

Example: See [Example "Receiver signals to generator mapping in direction finding scenario"](#) on page 363.

Usage: Query only

Manual operation: See ["Generator Profiles"](#) on page 277

ASSignment:GENerator:PATH:EMITter:LIST?

Queries the list of assigned emitters/interferes to the selected path.

Return values:

<List> "<Emitter/Inter#1>","<Emitter/Inter#2>",...

Example: See [Example "Performing signal to generator mapping in 2D scenario"](#) on page 362.

Usage: Query only

Manual operation: See ["Generator Profiles"](#) on page 277

23.10 Emitter Commands

Example: Creating emitters

```

EMITter:CREate "My_EmitterGuidance"
EMITter:COMMeNt "guidance&surveillance"
EMITter:EIRP 120
EMITter:FREQuency 3e+09
EMITter:MODE:ADD
EMITter:MODE:SELEct 1
EMITter:MODE:NAME "Guidance"
ANTenna:CATalog?
// "Isotropic","My_PencilBeam","Test CSC","My_PlanarAntenna","Test","Custom"
EMITter:MODE:ANTenna "My_PencilBeam"
SCAN:CATalog?
// "Test Antenna Scan","My_RasterScan"
EMITter:MODE:SCAN "My_RasterScan"
EMITter:MODE:BEAM:ADD
EMITter:MODE:BEAM:SELEct 1
EMITter:MODE:BEAM:STATe 1
SEQuence:CATalog?
// "Test Sequence","My_S2_S1-F1_S1-F2","PT"
EMITter:MODE:BEAM:SEQuence "My_S2_S1-F1_S1-F2"
EMITter:MODE:BEAM:OFFSet:FREQuency 0
EMITter:MODE:BEAM:OFFSet:ELEVation 0
EMITter:MODE:BEAM:OFFSet:AZIMuth 0

```

EMITter:EIRP.....	366
EMITter:FREQuency.....	367
EMITter:MODE:ANTenna.....	367
EMITter:MODE:BEAM:OFFSet:AZIMuth.....	367
EMITter:MODE:BEAM:OFFSet:ELEVation.....	367
EMITter:MODE:BEAM:OFFSet:FREQuency.....	367
EMITter:MODE:BEAM:SEQuence.....	368
EMITter:MODE:BEAM:STATe.....	368
EMITter:MODE:SCAN.....	368

EMITter:EIRP <Eirp>

Sets the EIRP of the emitter.

Parameters:

<Eirp> float
 Range: -100 to 200
 Default unit: dBW

Example: see [Example "Creating emitters"](#) on page 366

Manual operation: See ["EIRP"](#) on page 199

EMITter:FREQUency <Frequency>

Sets the operating frequency.

Parameters:

<Frequency> float
Range: 1000 to 4.4e+10
Default unit: Hz

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Frequency"](#) on page 199

EMITter:MODE:ANTenna <Antenna>

Assigns an existing antenna pattern, see [ANTenna:CATalog?](#).

Parameters:

<Antenna> string

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Modes"](#) on page 199

EMITter:MODE:BEAM:OFFSet:AZIMuth <Azimuth>

Sets the Azimuth value for the beam offset.

Parameters:

<Azimuth> float
Range: -180 to 180

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Beams Definition"](#) on page 200

EMITter:MODE:BEAM:OFFSet:ELEVation <Elevation>

Offsets the position of the beam in both the azimuth or elevation.

Parameters:

<Elevation> float
Range: -90 to 90

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Beams Definition"](#) on page 200

EMITter:MODE:BEAM:OFFSet:FREQUency <Frequency>

Offsets the frequency of the beam.

Parameters:

<Frequency> float
Range: -1e+09 to 1e+09

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Beams Definition"](#) on page 200

EMITter:MODE:BEAM:SEQUence <Sequence>

Assigns a pulse sequence, see [SEQUence:CREate](#).

Parameters:

<Sequence> string

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Beams Definition"](#) on page 200

EMITter:MODE:BEAM:STATe <State>

Activates a beam.

Parameters:

<State> ON | OFF | 1 | 0

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Beams Definition"](#) on page 200

EMITter:MODE:SCAN <Scan>

Assigns an antenna scan, see [SCAN:CREate](#).

Parameters:

<Scan> string

Example: [Example "Creating emitters"](#) on page 366

Manual operation: See ["Emitter Modes"](#) on page 199

23.11 Generator Profiles and Instruments Commands

Example: Creating generator's profile

```

SCPI

INSTRument:COUNT?
// 4
INSTRument:LIST?
// 'SMW 200A, 0','SMW 200A, 100025','SMBV 100A, 100012'
INSTRument:CLEar
// deletes all instruments from the list

*****
* Creating a profile from the connected instrument(automatic mapping)
*****
INSTRument:ADD "10.113.0.104"
INSTRument:CREate
// creates a profile from the capabilities of the connected instrument
INSTRument:CAPabilities?
// 1K76|1B106|1B206|2K300|2K301|1K308|2K350|2K511|2K512|2K522|2B10|1B13T|#0|0|0|0|
GENerator:CATalog?
// "My_RS_SMBV","RS_SMW","My_SGT","SMW","SMW200A, 0"
GENerator:SELEct "SMW200A, 0"
GENerator:TYPE?
// SMW
GENerator:CAPabilities?
// 1K76|1B106|1B206|2K300|2K301|2K511|2K512|2K522|2B10|1B13T|#2e+10|0|6e+09|0|
INSTRument:LINK:STATus?
// 1
INSTRument:FIND? "SMW200A, 0"
// 1
// the generator profile "SMW200A, 0" is mapped to the
// first instrument in the list
INSTRument:FIRMWare?
// 3.1.18.2-3.20.012.88
INSTRument:RESourcE?
// TCPIP::10.113.10.91::INSTR
INSTRument:MAP?
// SMW200A, 0

*****
* Creating a profile manually and mapping it to an instrument
*****
GENerator:CREate "My_RS_SMW"
GENerator:TYPE SMW
... etc.

```

GENerator:TYPE.....	370
INSTrument:CAPabilities?.....	370
GENerator:CAPabilities.....	370
INSTrument:LIST?.....	370
INSTrument:ADD.....	371
INSTrument:CREate.....	371
INSTrument:FIRMware?.....	371
INSTrument:FIND?.....	371
INSTrument:MAP.....	372
INSTrument:LINK:STATus?.....	372
INSTrument:RESource?.....	372

GENerator:TYPE <Type>

Sets the instrument's type.

Parameters:

<Type> SMW | SMU | SMBV | SMJ | SGT

Example: see [Example "Creating generator's profile"](#) on page 369

Manual operation: See ["Model"](#) on page 241

INSTrument:CAPabilities?

GENerator:CAPabilities <Capabilities>

Queries the generator capabilities regarding supported scenario types and processing of waveforms and multi-segment files.

Parameters:

<Capabilities> string

Example: See [Example "Creating generator's profile"](#) on page 369

Manual operation: See ["Installed Options"](#) on page 241
 See ["Capabilities"](#) on page 242
 See ["Basebands"](#) on page 243
 See ["RF Outputs"](#) on page 243

INSTrument:LIST?

Queries the names of the signal generators that are connected to the R&S Pulse Sequencer.

Return values:

<List> "<Instr#1>",<Instr#2>",...

Example: See [Example "Creating generator's profile"](#) on page 369.

Usage: Query only

Manual operation: See ["List of connected instruments"](#) on page 239

INSTrument:ADD <Add>

Adds an instrument with the specified IP address, computer name, or the VISA resource string.

Setting parameters:

<Add> string
 "<IP_Address>" or "<Computer_Name>" or "<GPIB_Address>"

Example: See [Example "Creating generator's profile"](#) on page 369

Usage: Setting only

Manual operation: See ["Add"](#) on page 238

INSTrument:CREate

Creates a generator profile for the selected connected generator.

Example: see [Example "Creating generator's profile"](#) on page 369

Usage: Event

Manual operation: See ["Gen Profile"](#) on page 238

INSTrument:FIRMware?

Queries the firmware version of the selected instrument.

Return values:

<Firmware> string

Example: see [Example "Creating generator's profile"](#) on page 369

Usage: Query only

Manual operation: See ["List of connected instruments"](#) on page 239

INSTrument:FIND? <GenProfileName>

Queries whether this particular generator profile is mapped to a physical instrument.

Query parameters:

<GenProfileName> string
 One of the created generation profiles, as queried with the command [GENerator:CATalog?](#).

Return values:

<Find> float
 A number that indicates the physical instrument in the "Generator Mapping" dialog
 If the profile is not mapped, the query returns -1.

Example: see [Example "Creating generator's profile"](#) on page 369

Usage: Query only

INSTRument:MAP <Map>

Maps a selected generator's profile to the selected instrument.

Parameters:

<Map> string
Existing generator profile
Use the command `GENerator:CATalog?` to query the existing profiles

Example: See [Example "Creating generator's profile"](#) on page 369

Manual operation: See "[List of connected instruments](#)" on page 239

INSTRument:LINK:STATus?

Queries whether a generator profile is linked the instrument selected with the command `INSTRument:SElect`.

Return values:

<Status> 0 | 1
0
No generator profile is selected
1
A generator profile is linked to the instrument

Example: See [Example "Creating generator's profile"](#) on page 369

Usage: Query only

Manual operation: See "[List of connected instruments](#)" on page 239

INSTRument:RESource?

Queries the resource string of the instrument selected with the command `INSTRument:SElect`.

Return values:

<Resource> string

Example: see [Example "Creating generator's profile"](#) on page 369

Usage: Query only

23.12 Inter-Pulse Modulation Commands

Example: Using list and waveform type IPM profiles

```

SCPI
IPM:CREate "My_PRI-Stagger"
IPM:COMment "List: 600 us, 750 us, 910 us"
IPM:UNIT SEConds
IPM:TYPE LIST
IPM:LIST:ITEM:ADD
IPM:LIST:ITEM:SElect 1
IPM:LIST:ITEM:VALue 0.0006
IPM:LIST:ITEM:REPetitions 1
IPM:LIST:ITEM:COUNT 1
IPM:LIST:ITEM:ADD
IPM:LIST:ITEM:SElect 2
IPM:LIST:ITEM:VALue 0.00075
IPM:LIST:ITEM:COUNT 2
...
IPM:LIST:SAVE "C:\_PS_files\IPM_PRI_Stagger.txt"
IPM:LIST:CLEar
// load an existing ASCII file
IPM:LIST:LOAD "C:\_PS_files\IPM_PRList.txt"

IPM:CREate "Stagger PRI"
IPM:UNIT SEConds
IPM:TYPE WAVeform
IPM:WAVeform:TYPE SINE
IPM:WAVeform:OFFSet 0.0004
IPM:WAVeform:PHASe 0
IPM:WAVeform:PKPK 0.00014
IPM:WAVeform:BASE PULSe
IPM:WAVeform:COUNT 30

```

Example: Assigning an IPM profile to a sequence

```

SCPI
SEquence:SElect "Test Sequence"
SEquence:ITEM:SElect 1
SEquence:ITEM:IPM:SOURce "My_PRI-Stagger"
SEquence:ITEM:IPM:TARGet:TYPE PARAMeter
SEquence:ITEM:IPM:TARGet:PARAMeter PRI
SEquence:ITEM:IPM:EQUation "0.5*x+1"
SEquence:ITEM:IPM:MODE INDividual
SEquence:ITEM:IPM:REStart 0
SEquence:ITEM:IPM:RANDom:RESet 0

```

Example: Using the step IPM profile

```

SCPI
IPM:CREate "FHOP"
IPM:COMment "Frequency Hops"
IPM:UNIT HERTz
IPM:TYPE STEPs
IPM:STEP:START 0
IPM:STEP:INCRement 1e+06
IPM:STEP:STEPS 5
IPM:STEP:BURSt 2

```

Example: Using PRI profiles of random type

```

SCPI
IPM:CREate "My_PRI-Jitter"
IPM:UNIT SEConds
IPM:TYPE RANDom
IPM:RANDom:DISTRibution UNIFORM
IPM:RANDom:UNIFORM:MINimum 0.0011
IPM:RANDom:UNIFORM:MAXimum 0.00115
IPM:RANDom:UNIFORM:STEP 2.5e-06

IPM:CREate "Test"
IPM:UNIT SEConds
IPM:TYPE RSTep
IPM:RSTep:MINimum 0.0005
IPM:RSTep:MAXimum 0.005
IPM:RSTep:STEP:MINimum 0.0001
IPM:RSTep:STEP:MAXimum 0.0005
IPM:RSTep:PERiod 10

IPM:CREate "Test2"
IPM:UNIT HERTz
IPM:TYPE RLISt
// create a List (see the "My_PRI-Stagger" example)
IPM:RLISt:BURSt 3
IPM:RLISt:REUSE 0

```

Example: Using the interpolated shape IPM profile

```

SCPI
IPM:CREate "Custom IPM"
IPM:UNIT DB
IPM:TYPE SHAPE
// create a List (see the "My_PRI-Stagger" example)
IPM:SHAPE:BASE PULSE
IPM:SHAPE:COUNT 10
// IPM:SHAPE:BASE TIME
// IPM:SHAPE:PERiod 0.5
IPM:SHAPE:INTerpol NONE

```

IPM:TYPE.....	375
IPM:UNIT.....	376
SEquence:ITEM:IPM:TARGet:PARAmeter.....	376
SEquence:ITEM:IPM:TARGet:TYPE.....	376
SEquence:ITEM:IPM:TARGet:VARiable.....	376
IPM:EQUation.....	377
IPM:LIST:ITEM:VALue.....	377
IPM:LIST:ITEM:REPetition.....	377
IPM:LIST:LOAD.....	377
IPM:LIST:SAVE.....	377
IPM:RANDom:DISTRibution.....	378
IPM:RANDom:NORMal:LIMit.....	378
IPM:RANDom:NORMal:MEAN.....	378
IPM:RANDom:NORMal:STD.....	378
IPM:RANDom:U:CENTer.....	379
IPM:RANDom:U:RANGe.....	379
IPM:RANDom:UNIForm:MINimum.....	379
IPM:RANDom:UNIForm:MAXimum.....	379
IPM:RANDom:UNIForm:STEP.....	379
IPM:RLISt:BURSt.....	379
IPM:RLISt:REUSe.....	380
IPM:SHAPE:BASE.....	380
IPM:SHAPE:COUNt.....	380
IPM:SHAPE:INTerpol.....	380
IPM:SHAPE:PERiod.....	381
IPM:STEP:BURSt.....	381
IPM:STEP:INCRe ment.....	381
IPM:STEP:STARt.....	381
IPM:STEP:STEPs.....	381
IPM:RSTep:MINimum.....	382
IPM:RSTep:MAXimum.....	382
IPM:RSTep:STEP:MINimum.....	382
IPM:RSTep:STEP:MAXimum.....	382
IPM:RSTep:PERiod.....	382
IPM:WAVeform:TYPE.....	382
IPM:WAVeform:BASE.....	383
IPM:WAVeform:COUNt.....	383
IPM:WAVeform:OFFSet.....	383
IPM:WAVeform:PHASe.....	383
IPM:WAVeform:PERiod.....	384
IPM:WAVeform:PKPK.....	384
SEquence:ITEM:IPM:EQUation.....	384
SEquence:ITEM:IPM:MODE.....	384
SEquence:ITEM:IPM:RANDom:RESet.....	384
SEquence:ITEM:IPM:REStart.....	385
SEquence:ITEM:IPM:SOURce.....	385

IPM:TYPE <Type>

Sets the shape of the profile.

Parameters:

<Type> STEPs | WAVEform | RLISt | LISt | SHAPe | RANDom | EQUation | PLUGin | RSTep

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See "Profile" on page 138

IPM:UNIT <Unit>

Sets the units of the profile.

Parameters:

<Unit> NONE | SEConds | HERTz | DB | DEGRees | PERCent

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See "Unit of Affected Parameter" on page 138

SEquence:ITEM:IPM:TARGet:PARAmeter <Parameter>

Sets the pulse parameter to that the IPM profile is assigned.

Parameters:

<Parameter> OVERshoot | LEVel | ATOP | FSKDeviation | RLEVel | DROop | AMDepth | RISE | WIDTh | DELay | FMDeviation | FALL | SRATe | FMFRequency | CDEViation | ABASe | RFRequency | PRF | PHASe | AMFRequency | PRI | FREquency

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See "Assign to" on page 149

SEquence:ITEM:IPM:TARGet:TYPE <Type>

Sets whether the profile is assigned to a parameter or to a variable.

Parameters:

<Type> PARAmeter | VARiable

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See "Assign to" on page 149

SEquence:ITEM:IPM:TARGet:VARiable <Variable>

Sets the variable to that the profile is assigned.

Parameters:

<Variable> string

Manual operation: See "Assign to" on page 149

IPM:EQUation <Equation>

Defines the IPM shape as a function.

Parameters:

<Equation> string

Example: IPM:EQUation "(i<10) ? 5 : (i*0.5)"

Manual operation: See ["Equation"](#) on page 143

IPM:LIST:ITEM:VALue <Value>

Sets the value of the selected list item.

Parameters:

<Value> float

Range: -1e+11 to 1e+11

Example: see [Example "Using list and waveform type IPM profiles"](#)
on page 373

Manual operation: See ["List"](#) on page 140

IPM:LIST:ITEM:REPetition <Repetition>

Sets the number of times a list item is repeated.

Parameters:

<Repetition> float

Range: 1 to 1e+09

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["List"](#) on page 140

IPM:LIST:LOAD <Load>

Loads an IPM profile form an ASCII file.

Parameters:

<Load> string

File path, file name, and file extension

Example: See [Using list and waveform type IPM profiles](#)

Manual operation: See ["Edit List"](#) on page 147

IPM:LIST:SAVE <Save>

Stores the IPM profile as a file.

Parameters:

<Save> string
File path incl. file name and extension.

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["Edit List"](#) on page 147

IPM:RANDom:DISTRibution <Distribution>

Sets the distribution function.

Parameters:

<Distribution> UNIFORM | NORMAL | U

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:NORMal:LIMit <Limit>

Sets the limit parameter of the normal distribution function.

Parameters:

<Limit> float
Range: -1e+09 to 1e+09

Example: see [IPM:RANDom:NORMal:MEAN](#) on page 378

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:NORMal:MEAN <Mean>

Sets the mean parameter of the normal distribution function.

Parameters:

<Mean> float
Range: -1e+09 to 1e+09

Example:

```
IPM:RANDom:DISTRibution NORMal
IPM:RANDom:NORMal:MEAN 0
IPM:RANDom:NORMal:STD 1
IPM:RANDom:NORMal:LIMit 3
```

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:NORMal:STD <Std>

Sets the standard deviation parameter of the normal distribution function.

Parameters:

<Std> float
Range: 1e-09 to 1e+06

Example: see [IPM:RANDom:NORMa1:MEAN](#) on page 378

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:U:CENTer <Center>

Sets the center parameter of the U distribution.

Parameters:

<Center> float
Range: -1e+09 to 1e+09

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:U:RANGe <Range>

Sets the range parameter of the U distribution.

Parameters:

<Range> float
Range: 1e-09 to 1e+09

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:UNIForm:MINimum <Minimum>

IPM:RANDom:UNIForm:MAXimum <Maximum>

Sets the range of the uniform distribution function.

Parameters:

<Maximum> float
Range: -1e+09 to 1e+09

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random"](#) on page 145

IPM:RANDom:UNIForm:STEP <Step>

Sets the granularity of the uniform distribution function.

Parameters:

<Step> float
Range: 1e-09 to 1e+09

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random"](#) on page 145

IPM:RLIST:BURSt <Burst>

Defines how many times an increment is repeated.

Parameters:

<Burst> float
Range: 1 to 1000

Example:

Manual operation: See ["Random List"](#) on page 143

IPM:RLIST:REUSe <Reuse>

If disabled, each value is used only once.

Parameters:

<Reuse> ON | OFF | 1 | 0

Example: IPM:RLIST:REUSe 0

Manual operation: See ["Random List"](#) on page 143

IPM:SHAPE:BASE <Base>

Defines the way the list items are processed.

Parameters:

<Base> PULSe | TIME

Example: see [Using the interpolated shape IPM profile](#)

Manual operation: See ["Interpolated Shape"](#) on page 142

IPM:SHAPE:COUNT <Count>

Sets the number of pulses for that the data from the list is used.

Parameters:

<Count> integer
Range: 1 to 1e+09

Example: see [Using the interpolated shape IPM profile](#)

Manual operation: See ["Interpolated Shape"](#) on page 142

IPM:SHAPE:INTerpol <Interpol>

Enables a linear transition between the increments.

Parameters:

<Interpol> LINear | NONE

Example: see [Using the interpolated shape IPM profile](#)

Manual operation: See ["Interpolated Shape"](#) on page 142

IPM:SHAPE:PERiod <Period>

Sets the period of time over that the list items are equally distributed.

Parameters:

<Period> float
Range: 1e-09 to 1e+09

Example: see [Using the interpolated shape IPM profile](#)

Manual operation: See ["Interpolated Shape"](#) on page 142

IPM:STEP:BURSt <Burst>

Sets the number of times an increment is repeated.

Parameters:

<Burst> float
Range: 1 to 1000

Example: see [Example "Using the step IPM profile"](#) on page 374

Manual operation: See ["Steps"](#) on page 139

IPM:STEP:INCRement <Increment>

Sets the step size.

Parameters:

<Increment> float
Range: -1e+09 to 1e+09

Example: see [Example "Using the step IPM profile"](#) on page 374

Manual operation: See ["Steps"](#) on page 139

IPM:STEP:STARt <Start>

Sets the start value.

Parameters:

<Start> float
Range: -1e+09 to 1e+09

Example: see [Example "Using the step IPM profile"](#) on page 374

Manual operation: See ["Steps"](#) on page 139

IPM:STEP:STEPs <Steps>

Sets the number of steps.

Parameters:

<Steps> float
Range: 1 to 10000

Example: see [Example "Using the step IPM profile"](#) on page 374

Manual operation: See ["Steps"](#) on page 139

IPM:RSTep:MINimum <Minimum>

IPM:RSTep:MAXimum <Maximum>

Sets the value range.

Parameters:

<Maximum> float
Range: 0 to 1e+11

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random Steps"](#) on page 144

IPM:RSTep:STEP:MINimum <Minimum>

IPM:RSTep:STEP:MAXimum <Maximum>

Sets the step size range.

Parameters:

<Maximum> float
Range: 0.1 to 0.5

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random Steps"](#) on page 144

IPM:RSTep:PERiod <Period>

Sets the pattern length.

Parameters:

<Period> float
Range: 0 to 4096

Example: see [Example "Using PRI profiles of random type"](#) on page 374

Manual operation: See ["Random Steps"](#) on page 144

IPM:WAVEform:TYPE <Type>

Sets the profile shape.

Parameters:

<Type> RAMP | SINE | TRIangular

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["Waveform"](#) on page 141

IPM:WAVeform:BASE <Base>

Defines how the waveform period is defined, as a time duration or as a number of pulses.

Parameters:

<Base> PULSe | TIME

Example: See [Using list and waveform type IPM profiles](#)

Manual operation: See ["Waveform"](#) on page 141

IPM:WAVeform:COUNT <Count>

Sets the waveform period as number of pulses.

Parameters:

<Count> integer
Range: 1 to 1e+09

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["Waveform"](#) on page 141

IPM:WAVeform:OFFSet <Offset>

Shifts the profile by the selected offset.

Parameters:

<Offset> float
Range: -1e+09 to 1e+09

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["Waveform"](#) on page 141

IPM:WAVeform:PHASe <Phase>

Enables a phase offset to change the start phase of the sine wave.

Parameters:

<Phase> float
Range: -1e+09 to 1e+09
Default unit: sec

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See ["Waveform"](#) on page 141

IPM:WAVeform:PERiod <Period>

Sets the waveform period.

Parameters:

<Period> float
Range: 1e-09 to 1e+09
Default unit: sec

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See "[Waveform](#)" on page 141

IPM:WAVeform:PKPK <Pkp>

Sets the value range of the linear ramp profile or the period of the sine profile.

Parameters:

<Pkp> float
Range: 1e-09 to 1e+09
Default unit: sec

Example: see [Using list and waveform type IPM profiles](#)

Manual operation: See "[Waveform](#)" on page 141

SEquence:ITEM:IPM:EQUation <Equation>

Defines output value of the IPM mathematically.

Parameters:

<Equation> string

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See "[Transformation f\(x\)](#)" on page 149

SEquence:ITEM:IPM:MODE <Mode>

Defines the way the variations are applied on repeating pulses.

Parameters:

<Mode> INDividual | SAME

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See "[Repetition](#)" on page 150

SEquence:ITEM:IPM:RANDom:RESet <Reset>

Resets the start seed of random generator.

Parameters:

<Reset> ON | OFF | 1 | 0

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See ["Restart"](#) on page 150

SEQUENCE:ITEM:IPM:REStart <Restart>

Restarts the IPM for this sequence line item.

Parameters:

<Restart> ON | OFF | 1 | 0

Example: see [Assigning an IPM profile to a sequence](#)

Manual operation: See ["Restart"](#) on page 150

SEQUENCE:ITEM:IPM:SOURce <Source>

Selects the profile source.

Use the command [IPM:CATalog?](#) to query the existing profiles.

Parameters:

<Source> string

Example: See [Assigning an IPM profile to a sequence](#)

Manual operation: See ["Source Profile"](#) on page 149

23.13 Marker Commands

Marker commands enable or up to four markers in any combination at once.

Markers signals are binary signals. Each marker signal is represented by a single bit within a marker byte. The [Table 23-1](#) explains the assignment.

Table 23-1: Setting parameter as function of the marker states

Marker	Marker byte	Value of the settings parameter in the remote commands
M1	0000 0001	0x01
M2	0000 0010	0x02
M3	0000 0100	0x04
M4	0000 1000	0x08

In the following examples, we assume that a pulse-base sequence "Test Sequence" and the required Pulses have been created.

See also:

- [Example "Handling items"](#) on page 325
- [Example "Creating an unmodulated pulse"](#) on page 395

Example: Defining pulse and sequence markers and enabling the global markers

```

SCPI
PULSe:SElect "P1"
// activate Marker 1 for the whole pulse on time (rise, fall, with time)
PULSe:MARKer:RISE 1
PULSe:MARKer:FALL 1
PULSe:MARKer:WIDTh 1
// enable Marker 2 on restart
PULSe:MARKer:AUTO 2
// enable Marker 3 als gate marker
PULSe:MARKer:GATE 4

SEquence:SElect "Test Sequence"
SEquence:ITEM:SElect 1
SEquence:ITEM:MARKer:FIRSt 2
SEquence:ITEM:MARKer:LAST 4
SEquence:ITEM:MARKer:ALL 1

// enable Marker 4 to mark the fifth loop run, i.e.
// enable the condition $loop_i = 5 for Marker 4
// the used loop variable $loop must exist
SEquence:ITEM:MARKer:CONDition 8
SEquence:ITEM:MARKer:CONDition:VARiable "$loop_i"
SEquence:ITEM:MARKer:CONDition:TYPE EQUAL
SEquence:ITEM:MARKer:CONDition:VALue "5"

SCENario:OUTPut:MARKer:ENABle 1
SCENario:OUTPut:MARKer:SCENario:ENABle 0
// SCENario:OUTPut:MARKer:SCENario:DURation 0.01

PULSe:MARKer:RISE..... 387
PULSe:MARKer:WIDTh..... 387
PULSe:MARKer:FALL..... 387
PULSe:MARKer:AUTO..... 387
PULSe:MARKer:GATE..... 387
SEquence:ITEM:MARKer:CONDition..... 387
SEquence:ITEM:MARKer:FIRSt..... 387
SEquence:ITEM:MARKer:LAST..... 387
SEquence:ITEM:MARKer:ALL..... 387
SEquence:ITEM:MARKer:CONDition..... 388
SEquence:ITEM:MARKer:CONDition:TYPE..... 388
SEquence:ITEM:MARKer:CONDition:VARiable..... 388
SEquence:ITEM:MARKer:CONDition:VALue..... 388
SCENario:OUTPut:MARKer:ENABle..... 389
SCENario:OUTPut:MARKer:SCENario:ENABle..... 389
SCENario:OUTPut:MARKer:SCENario:DURation..... 389

```

PULSe:MARKer:RISE <Rise>
PULSe:MARKer:WIDTh <Width>
PULSe:MARKer:FALL <Fall>

Enables up to four markers of the corresponding type.

Parameters:

<Fall> float
see [Table 23-1](#).
Range: 0 to 65535

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Pulse Markers"](#) on page 301

PULSe:MARKer:AUTO <Auto>

Enables up to four restart markers.

Parameters:

<Auto> float
see [Table 23-1](#).
Range: 0 to 65535

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Pulse Markers"](#) on page 301

PULSe:MARKer:GATE <Gate>

Enables up to four gate markers.

Parameters:

<Gate> float
see [Table 23-1](#).
Range: 0 to 65535

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Pulse Markers"](#) on page 301

SEquence:ITEM:MARKer:CONDition <Condition>
SEquence:ITEM:MARKer:FIRSt <First>
SEquence:ITEM:MARKer:LAST <Last>
SEquence:ITEM:MARKer:ALL <All>

Enables up to four markers of the corresponding type.

Parameters:

<All> float
 see [Table 23-1](#).
 Range: 0 to 65535

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Sequence Markers"](#) on page 302

SEquence:ITEM:MARKer:CONDition <Condition>**Parameters:**

<Condition> float
 see [Table 23-1](#).
 *RST: 0

Manual operation: See ["Sequence Markers"](#) on page 302

SEquence:ITEM:MARKer:CONDition:TYPE <Type>

Sets the sign in the logical condition.

Parameters:

<Type> SMALLer | GREater | EQUal | NOTequal

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Sequence Markers"](#) on page 302

SEquence:ITEM:MARKer:CONDition:VARIABLE <Variable>

Defines the value that is compared with the fixed values set with the command [SEquence:ITEM:MARKer:CONDition:VALue](#) on page 388.

Parameters:

<Variable> string

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Sequence Markers"](#) on page 302

SEquence:ITEM:MARKer:CONDition:VALue <Value>

Sets the numerical value used with the comparison.

Parameters:

<Value> string

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Sequence Markers"](#) on page 302

SCENario:OUTPut:MARKer:ENABLE <Enable>

Enables that markers are considered by the generation of the output waveform file.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Marker Signals"](#) on page 275
See ["Global Marker Signals"](#) on page 303

SCENario:OUTPut:MARKer:SCENario:ENABLE <Enable>

Enables an additional marker, that is held high from the scenario start until the duration, selected with the command `SCENario:OUTPut:MARKer:SCENario:DURection`.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Marker Signals"](#) on page 275
See ["Global Marker Signals"](#) on page 303

SCENario:OUTPut:MARKer:SCENario:DURection <Duration>

Sets the duration of the scenario marker.

Parameters:

<Duration> float
Range: 0 to 1
Increment: 1 ms
Default unit: sec

Example: see [Example "Defining pulse and sequence markers and enabling the global markers"](#) on page 386

Manual operation: See ["Marker Signals"](#) on page 275
See ["Global Marker Signals"](#) on page 303

23.14 Plugin and Reporting Commands

Example: Loading plugins in the repository

```

PLUGin:CREate "My_QAM_Plugin"
PLUGin:LOAD "C:\_PS_files\QAM.dll"
PLUGin:MODule:AUTHor?
// Rohde&Schwarz
PLUGin:MODule:NAME?
// QAM
PLUGin:MODule:TYPE?
// MOP
PLUGin:MODule:VERSion?
// 2.0.0
PLUGin:MODule:COMMeNt?
// This plugin creates a QAM modulation.
// Parameters:
// Rate [Hz]      symbol rate
PLUGin:MODule:DATA?
// 0

```

Example: Using a plugin as a modulation source

```

SCPI
PULSe:CREate "Custom QAM"
PULSe:MOP:ENABLE 1
PULSe:MOP:TYPE PLUGin
PLUGin:CATalog?
// "My_QAM_Plugin","CustomIPM"
PULSe:MOP:PLUGin:NAME "My_QAM_Plugin"
PULSe:MOP:PLUGin:VARiable:CATalog?
// "Rate","Type"
PULSe:MOP:PLUGin:VARiable:SElect "Rate"
PULSe:MOP:PLUGin:VARiable:VALue?
// 100000
PULSe:MOP:PLUGin:VARiable:SElect "Type"
PULSe:MOP:PLUGin:VARiable:VALue?
// 2
// selected is the third value from the list of variables
DSRC:CATalog?
// "DataSource"
PULSe:MOP:DATA:DSRC "DataSource"
PULSe:MOP:DATA:CODing NONE
PULSe:MOP:FILTer:TYPE COS
PULSe:MOP:FILTer:LENGth 128
PULSe:MOP:FILTer:BT 0.3
PULSe:MOP:FILTer:BWIDth 1000

```

Example: Using a plugin as an IPM profile

```

SCPI
IPM:CREate "Custom IPM"
IPM:UNIT PERCent
IPM:TYPE PLUGin
PLUGin:CATalog?
// "My_QAM_Plugin","CustomIPM"
IPM:PLUGin:NAME "CustomIPM"
IPM:PLUGin:VARiable:CATalog?
// "$step","$pw"
IPM:PLUGin:VARiable:SElect "$pw"
IPM:PLUGin:VARiable:VALue 0.00001

```

Example: Generating reports

```

SCPI
SCENario:PDW:ENABle 1
SCENario:PDW:PATH "C:\_My_Report_Files"
SCENario:PDW:TYPE PLUGin
PLUGin:CATalog?
// "My_QAM_Plugin","CustomIPM","Report PDW","Reporting DFS"
SCENario:PDW:PLUGin:NAME "Report PDW"
SCENario:PDW:PLUGin:VARiable:CATalog?
// "File","Header"
SCENario:PDW:PLUGin:VARiable:SElect "File"
SCENario:PDW:PLUGin:VARiable:VALue?
// "pdw_out.txt"

```

PLUGin:LOAD.....	392
PLUGin:MODule:AUTHor?.....	392
PLUGin:MODule:COMMeNt?.....	392
PLUGin:MODule:VERSiOn?.....	392
PLUGin:MODule:TYPE?.....	392
PLUGin:MODule:DATA?.....	392
SCENario:PDW:PLUGin:VARiable:CATalog?.....	393
IPM:PLUGin:VARiable:CATalog?.....	393
PULSe:MOP:PLUGin:VARiable:CATalog?.....	393
SCENario:PDW:PLUGin:VARiable:SElect.....	393
IPM:PLUGin:VARiable:SElect.....	393
PULSe:MOP:PLUGin:VARiable:SElect.....	393
SCENario:PDW:PLUGin:VARiable:VALue.....	393
IPM:PLUGin:VARiable:VALue.....	393
PULSe:MOP:PLUGin:VARiable:VALue.....	393
SCENario:PDW:ENABle.....	393
SCENario:PDW:PATH.....	393
SCENario:PDW:TYPE.....	394
SCENario:PDW:PLUGin:NAME.....	394
SCENario:PDW:TEMPlate.....	394

PLUGin:LOAD <Load>

Loads the selected DLL file, see also [Chapter C, "Plugin Programming API"](#), on page 495.

Setting parameters:

<Load> string
File path incl. file name and extension

Example: See [Example "Loading plugins in the repository"](#) on page 390

Usage: Setting only

PLUGin:MODule:AUTHor?
PLUGin:MODule:COMMeNt?
PLUGin:MODule:VERsion?
PLUGin:MODule:TYPE?

Queries information on the loaded file.

The query returns information as specified in the description of the corresponding function in [Chapter C, "Plugin Programming API"](#), on page 495.

The following are the possible values for the type query.

Return values:

<Type> MOP | REPort | IPM
MOP
 Plugin for MOP
IPM
 Plugin for IPM
REPort
 Plugin for reports created during the waveform generation

Example: See [Example "Loading plugins in the repository"](#) on page 390

Usage: Query only

PLUGin:MODule:DATA?

Queries whether the plugin requires data from a data source.

Return values:

<Data> 0 | 1
0
 Data source is not required
1
 Data source is required
 *RST: 0

Example: See [Example "Loading plugins in the repository"](#) on page 390

Usage: Query only

SCENario:PDW:PLUGin:VARiable:CATalog?**IPM:PLUGin:VARiable:CATalog?****PULSe:MOP:PLUGin:VARiable:CATalog?**

Queries the variables used in the plugin.

Return values:

<Catalog> string

Example: see [Using a plugin as a modulation source](#)**Usage:** Query only**Manual operation:** See "[Plugin Variables](#)" on page 100

SCENario:PDW:PLUGin:VARiable:SElect <Select>**IPM:PLUGin:VARiable:SElect <Select>****PULSe:MOP:PLUGin:VARiable:SElect <Select>**

Selects a plugin variable.

Parameters:

<Select> string

Example: see [Using a plugin as a modulation source](#)**Manual operation:** See "[Plugin Variables](#)" on page 100

SCENario:PDW:PLUGin:VARiable:VALue <Value>**IPM:PLUGin:VARiable:VALue <Value>****PULSe:MOP:PLUGin:VARiable:VALue <Value>**

Sets the values of the selected variable.

Parameters:

<Value> string

Example: see [Using a plugin as a modulation source](#)**Manual operation:** See "[Plugin Variables](#)" on page 100

SCENario:PDW:ENABLE <Enable>

Enables generation of Pulse Descriptor Word (PDW) reports.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: See [Example "Generating reports"](#) on page 391**Manual operation:** See "[Reporting Enable](#)" on page 310

SCENario:PDW:PATH <Path>

Sets the target directory in that the generated report files are stored.

Parameters:

<Path> string

Example: see [Example "Generating reports"](#) on page 391

Manual operation: See ["Target Path, Set Path"](#) on page 310

SCENario:PDW:TYPE <Type>

Sets the template used by the reporting function.

Parameters:

<Type> DEFault | TEMPlate | PLUGin

Manual operation: See ["Type"](#) on page 310

SCENario:PDW:PLUGin:NAME <Name>

Selects and loads a reporting template. This template must exist in the "Plugin" library.

To query a list of available plugins, use the command [PLUGin:CATalog?](#).

Parameters:

<Name> string

Example: See [Example "Generating reports"](#) on page 391

Manual operation: See ["Plugin and Plugin Variables"](#) on page 312

SCENario:PDW:TEMPlate <Template>

Edits the selected template.

Parameters:

<Template> string

Example:

```

SCPI
SCENario:PDW:TEMplate?
# Pulse Sequencer PDW Report
#
Date:      <ISODATE>
Repository: <REPOSITORY>
Scenario:  <SCENARIO>

          TOA |      RF |      PW |      PA | MF | MOP |      BW |
          ns |      GHz |      us |      dBm |  |    |      kHz |
-----
.HDR
<TOA> | <RF> | <PW> | <PA> | <MF> | <MOP> | <BW> |
# END OF REPORT
.OPT <TOA:12.0f:-9>
.OPT <RF:9.6f:9>
.OPT <PW:9.3f:-6>
.OPT <PA:6.1f:0>
.OPT <BW:6.0f:3>

```

Manual operation: See ["Edit Template"](#) on page 311

23.15 Pulse Commands

The following are examples on how to create and configure pulses in remote environment.

Example: Creating an unmodulated pulse

```

SCPI
PULSe:CREate "P_1"
PULSe:CATalog?
// "P_1"
PULSe:SElect "P_1"
PULSe:COMment "PW=100us"
PULSe:TIME:REFerence FULL
PULSe:TIME:WIDTh 100 us
PULSe:TIME:RISE 10 us
PULSe:TIME:FALL 10 us
PULSe:TYPE:RISE LINear
PULSe:TYPE:FALL LINear
PULSe:MOP:ENABle 0
PULSe:MOP:EXCLude:ENABle 0
// remane the pulse
PULSe:NAME "P1"

```

Example: Creating a linear chirp pulse

```

SCPI
PULSe:CREate "LinearChirp"
PULSe:CUSTom 0
PULSe:TIME:REFerence POWer
PULSe:TIME:RISE 1e-05
PULSe:TYPE:RISE RCOSine
PULSe:TIME:WIDTh 0.0001
PULSe:TIME:FALL 1e-05
PULSe:TYPE:FALL RCOSine
PULSe:LEVel:ON 0
PULSe:LEVel:OFF 100
PULSe:LEVel:DRooP 1
PULSe:RIPlle 0
PULSe:RIPlle:FREQuency 50000

PULSe:MOP:ENABle 1
PULSe:MOP:TYPE CHIRP
PULSe:MOP:CHIRp:TYPE UP
PULSe:MOP:CHIRp:DEVIation 5e+06
PULSe:MOP:EXCLude:MODE LEVel
PULSe:MOP:EXCLude:LEVel:STARt 25
PULSe:MOP:EXCLude:LEVel:STOP 25

```

Example: Creating an FM step pulse

```

SCPI
PULSe:CREate "FM_Step"
PULSe:TIME:REFerence FULL
PULSe:TIME:RISE 0
PULSe:TIME:WIDTh 0.000125
PULSe:TIME:FALL 0
PULSe:LEVel:ON 0
PULSe:LEVel:OFF 100
PULSe:MOP:EXCLude:MODE WIDTh

PULSe:MOP:ENABle 1
PULSe:MOP:COMment "fstart = -50 MHz, fend = 50 MHz, Df = 25 MHz (N = 5)"
PULSe:MOP:TYPE FMSTep
PULSe:MOP:FMSTep:ADD
PULSe:MOP:FMSTep:SElect 1
PULSe:MOP:FMSTep:DURation 2.49999993684469e-05
PULSe:MOP:FMSTep:FREQuency -50000000
PULSe:MOP:FMSTep:ADD
PULSe:MOP:FMSTep:SElect 2
PULSe:MOP:FMSTep:DURation 2.49999993684469e-05
PULSe:MOP:FMSTep:FREQuency -25000000
...

```

Example: Creating a BPSK pulse

```

SCPI
PULSe:CREate "Test"
PULSe:MOP:ENABle 1
PULSe:MOP:TYPE BPSK
PULSe:MOP:BPSK:TYPE NORMAl
PULSe:MOP:BPSK:SRATe:AUTO 0
PULSe:MOP:BPSK:SRATe 1e+06
PULSe:MOP:BPSK:PHASe 180
PULSe:MOP:BPSK:TTYpe COSine
PULSe:MOP:BPSK:TTIME 5

```

Example: Creating pulses with custom envelope

```

SCPI
PULSe:SELEct "Test"
PULSe:CUSTom 1
PULSe:ENVELOpe:MODE DATA
PULSe:ENVELOpe:DATA:LOAD "C:\_PS_files\mop shape custom.csv"
PULSe:ENVELOpe:DATA:UNIT VOLTAGE
PULSe:ENVELOpe:DATA:MULTIplier 1
PULSe:ENVELOpe:DATA:OFFSet 0
PULSe:ENVELOpe:DATA:ITEM:SELEct 1
PULSe:ENVELOpe:DATA:ITEM:VALue?
// 0.2
PULSe:ENVELOpe:DATA:ITEM:SELEct 7
PULSe:ENVELOpe:DATA:ITEM:VALue?
// 0.3
PULSe:ENVELOpe:DATA:ITEM:COUNT?
// 7

// PULSe:ENVELOpe:MODE EQUation
// PULSe:ENVELOpe:EQUation "# Sech Pulse
// 1 / ( cosh( ( 2 * (t-Tr-Ton/2) * ln( 1+sqrt(2) ) ) / Ton ) ) ^2"

```

PULSe:CUSTom.....	399
PULSe:ENVELOpe:DATA:ITEM:VALue.....	399
PULSe:ENVELOpe:DATA:LOAD.....	399
PULSe:ENVELOpe:DATA:UNIT.....	399
PULSe:ENVELOpe:DATA:MULTIplier.....	400
PULSe:ENVELOpe:DATA:OFFSet.....	400
PULSe:ENVELOpe:DATA:SAVE.....	400
PULSe:ENVELOpe:EQUation.....	400
PULSe:ENVELOpe:MODE.....	401
PULSe:LEVel:DROop.....	401
PULSe:LEVel:OFF.....	401
PULSe:LEVel:ON.....	401
PULSe:MOP:AM:FREQuency.....	401
PULSe:MOP:AM:MDEPth.....	402
PULSe:MOP:AM:TYPE.....	402

PULSe:MOP:AMSTep:DURation.....	402
PULSe:MOP:FMSTep:DURation.....	402
PULSe:MOP:AMSTep:LEVel.....	402
PULSe:MOP:FMSTep:FREQuency.....	403
PULSe:MOP:ASK:INVert.....	403
PULSe:MOP:ASK:MDEPth.....	403
PULSe:MOP:ASK:SRATe.....	403
PULSe:MOP:BARKEr:CODE.....	404
PULSe:MOP:BARKEr:TTime.....	404
PULSe:MOP:BARKEr:BLANk.....	404
PULSe:MOP:BPSK:PHASe.....	404
PULSe:MOP:BPSK:SRATe.....	405
PULSe:MOP:BPSK:SRATe:AUTO.....	405
PULSe:MOP:BPSK:TTime.....	405
PULSe:MOP:BPSK:TType.....	405
PULSe:MOP:BPSK:TYPE.....	405
PULSe:MOP:CHIRp:DEVIation.....	406
PULSe:MOP:CHIRp:TYPE.....	406
PULSe:MOP:DATA:CODIng.....	406
PULSe:MOP:DATA:DSRC.....	406
PULSe:MOP:ENABle.....	406
PULSe:MOP:EXCLude:ENABle.....	407
PULSe:MOP:EXCLude:LEVel:STARt.....	407
PULSe:MOP:EXCLude:LEVel:STOP.....	407
PULSe:MOP:EXCLude:MODE.....	407
PULSe:MOP:EXCLude:TIME:STARt.....	407
PULSe:MOP:EXCLude:TIME:STOP.....	407
PULSe:MOP:FILTer:BT.....	408
PULSe:MOP:FILTer:BWIDth.....	408
PULSe:MOP:FILTer:LENGth.....	408
PULSe:MOP:FILTer:ROLLoff.....	408
PULSe:MOP:FILTer:TYPE.....	409
PULSe:MOP:FM:DEVIation.....	409
PULSe:MOP:FM:FREQuency.....	409
PULSe:MOP:FSK:DEVIation.....	409
PULSe:MOP:FSK:SRATe.....	409
PULSe:MOP:NLCHirp:EQUation.....	410
PULSe:MOP:NOISe:BWIDth.....	410
PULSe:MOP:PCHirp:COEFFicient.....	410
PULSe:MOP:PCHirp:TERM.....	410
PULSe:MOP:PLISt:VALue.....	410
PULSe:MOP:POLY:LENGth.....	411
PULSe:MOP:POLY:TYPE.....	411
PULSe:MOP:QPSK:SRATe.....	411
PULSe:MOP:QPSK:TYPE.....	411
PULSe:MOP:QPSK:SOQPsk:IRIG.....	411
PULSe:MOP:TYPE.....	412
PULSe:OVERshoot.....	412
PULSe:OVERshoot:DECay.....	412
PULSe:RIPPlE.....	412

PULSe:RIPPlE:FREQUency.....	413
PULSe:TIME:FALL.....	413
PULSe:TIME:RISE.....	413
PULSe:TIME:REFerence.....	413
PULSe:TIME:WIDTh.....	413
PULSe:TYPE:FALL.....	414
PULSe:TYPE:RISE.....	414
PULSe:SETTings.....	414
PULSe:PREView:MODE.....	414
PULSe:PREView:MOP.....	414

PULSe:CUSToM <Custom>

Enables the use of a custom envelope function

Parameters:

<Custom> ON | OFF | 1 | 0

Manual operation: See ["Custom Envelope"](#) on page 75

PULSe:ENVELOPE:DATA:ITEM:VALue <Value>

Sets the value of the selected item.

Parameters:

<Value> float
 Range: -1e+09 to 1e+09

Example: see [Example "Creating pulses with custom envelope"](#)
 on page 397

Manual operation: See ["Envelope Definition as a Function form Imported Data"](#)
 on page 81

PULSe:ENVELOPE:DATA:LOAD <Load>

Loads an envelope description form an ASCII file.

Parameters:

<Load> string
 File path, file name, and file extension

Example: See [Example "Creating pulses with custom envelope"](#)
 on page 397

Manual operation: See ["Envelope Definition as a Function form Imported Data"](#)
 on page 81

PULSe:ENVELOPE:DATA:UNIT <Unit>

Sets the data format.

Parameters:

<Unit> VOLTage | WATTs | DB

Example:

see [Example "Creating pulses with custom envelope"](#)
on page 397

Manual operation:

See ["Envelope Definition as a Function form Imported Data"](#)
on page 81

PULSe:ENVELOpe:DATA:MULTIplier <Multiplier>

Sets a multiplier factor.

Parameters:

<Multiplier> float
Range: -100 to 100

Example:

see [Example "Creating pulses with custom envelope"](#)
on page 397

Manual operation:

See ["Envelope Definition as a Function form Imported Data"](#)
on page 81

PULSe:ENVELOpe:DATA:OFFSet <Offset>

Sets an offset for the envelope.

Parameters:

<Offset> float
Range: -100 to 100

Example:

see [Example "Creating pulses with custom envelope"](#)
on page 397

Manual operation:

See ["Envelope Definition as a Function form Imported Data"](#)
on page 81

PULSe:ENVELOpe:DATA:SAVE <Save>

Stores the custom envelope into file.

Parameters:

<Save> string
File path, file name, and file extension

Example:

```
PULSe:ENVELOpe:DATA:LOAD "C:\_PS_files\mop_shape.csv"
```

PULSe:ENVELOpe:EQUation <Equation>

Determines the envelope mathematically.

Parameters:

<Equation> string

Manual operation: See ["Envelope Definition as an Equation"](#) on page 80**PULSe:ENVELOpe:MODE <Mode>**

Selects the type of the custom envelope function.

Parameters:

<Mode> DATA | EQUation

Example: see [Example "Creating pulses with custom envelope"](#) on page 397**Manual operation:** See ["Custom Envelope"](#) on page 75**PULSe:LEVel:DRooP <Droop>**

Sets the amplitude droop.

Parameters:<Droop> float
Range: 0 to 50**Example:** see [Example "Creating a linear chirp pulse"](#) on page 396**Manual operation:** See ["Attenuation"](#) on page 77**PULSe:LEVel:OFF <Off>****PULSe:LEVel:ON <On>**

Sets the power during the pulse on time or the pulse off time.

Parameters:<On> float
Range: 0 to 100**Example:** see [Example "Creating a linear chirp pulse"](#) on page 396**Manual operation:** See ["Attenuation"](#) on page 77**PULSe:MOP:AM:FREQUency <Frequency>**

Sets modulation frequency.

Parameters:<Frequency> float
Range: 0.001 to 1e+09**Example:** see [PULSe:MOP:AM:TYPE](#) on page 402**Manual operation:** See ["AM"](#) on page 87

PULSe:MOP:AM:MDEPth <Mdepth>

Sets the modulation depth.

Parameters:

<Mdepth> float
Range: 0 to 100
Default unit: percent

Example: see [PULSe:MOP:AM:TYPE](#) on page 402

Manual operation: See ["AM"](#) on page 87

PULSe:MOP:AM:TYPE <Type>

Selects the modulation type.

Parameters:

<Type> STD | LSB | USB | SB

Example:

```
PULSe:MOP:TYPE AM
PULSe:MOP:AM:TYPE STD
PULSe:MOP:AM:FREQuency 50000
PULSe:MOP:AM:MDEPth 50
```

Manual operation: See ["AM"](#) on page 87

PULSe:MOP:AMSTep:DURation <Duration>**PULSe:MOP:FMSTep:DURation** <Duration>

Sets the step time.

Parameters:

<Duration> float
Range: 0 to 3600
Default unit: s

Example: see [Example "Creating an FM step pulse"](#) on page 396

Manual operation: See ["FM Step"](#) on page 90

PULSe:MOP:AMSTep:LEVel <Level>

Sets the step level.

Parameters:

<Level> float
Range: -100 to 0

Example: see [Example "Creating an FM step pulse"](#) on page 396

Manual operation: See ["AM Step"](#) on page 88

PULSe:MOP:FMSTep:FREQuency <Frequency>

Sets the step frequency.

Parameters:

<Frequency> float
Range: -1e+09 to 1e+09

Example: see [Example "Creating an FM step pulse"](#) on page 396

Manual operation: See ["FM Step"](#) on page 90

PULSe:MOP:ASK:INVert <Invert>

Inverts the modulation.

Parameters:

<Invert> ON | OFF | 1 | 0

Example:

```
PULSe:MOP:TYPE ASK
PULSe:MOP:ASK:MDEPth 100
PULSe:MOP:ASK:INVert 1
PULSe:MOP:ASK:SRATe 1e+06
```

Manual operation: See ["ASK"](#) on page 87

PULSe:MOP:ASK:MDEPth <Mdepth>

Sets the modulation depth.

Parameters:

<Mdepth> float
Range: 0 to 100
Default unit: percent

Example: see [PULSe:MOP:ASK:INVert](#) on page 403

Manual operation: See ["ASK"](#) on page 87

PULSe:MOP:ASK:SRATe <Srate>

Sets the symbol rate.

Parameters:

<Srate> float
Range: 1 to 1e+09

Example: see [PULSe:MOP:ASK:INVert](#) on page 403

Manual operation: See ["ASK"](#) on page 87

PULSe:MOP:BARKer:CODE <Code>

Selects the code sequence.

Parameters:

<Code> R3 | R4A | R4B | R5 | R7 | R11 | R13

Example:

```
PULSe:MOP:TYPE BARKer
PULSe:MOP:BARKer:CODE R13
PULSe:MOP:BARKer:TTIME 10
PULSe:MOP:BARKer:BLANK 1
```

Manual operation: See "[Barker](#)" on page 93

PULSe:MOP:BARKer:TTIME <Ttime>

Sets the transition time.

Parameters:

<Ttime> float
Range: 0 to 50
Default unit: percent

Example: See [PULSe:MOP:BARKer:CODE](#) on page 404

Manual operation: See "[Barker](#)" on page 93

PULSe:MOP:BARKer:BLANK <Blank>

Blanks out the signal during the transition time.

Parameters:

<Blank> ON | OFF | 1 | 0

Example: See [PULSe:MOP:BARKer:CODE](#) on page 404

Manual operation: See "[Barker](#)" on page 93

PULSe:MOP:BPSK:PHASe <Phase>

Sets the phase.

Parameters:

<Phase> float
Range: 0.1 to 180
Default unit: degree

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See "[BPSK](#)" on page 96

PULSe:MOP:BPSK:SRATe <Srate>

Sets the symbol rate.

Parameters:

<Srate> float
Range: 1 to 1e+09

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See ["BPSK"](#) on page 96

PULSe:MOP:BPSK:SRATe:AUTO <Auto>

Enables automatic adjusting of the bits in the pulse width.

Parameters:

<Auto> ON | OFF | 1 | 0

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See ["BPSK"](#) on page 96

PULSe:MOP:BPSK:TTIME <Ttime>

Sets the transition time.

Parameters:

<Ttime> float
Range: 0 to 50
Default unit: percent

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See ["BPSK"](#) on page 96

PULSe:MOP:BPSK:TTYPe <Ttype>

Selects the transition type.

Parameters:

<Ttype> LINear | COSine

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See ["BPSK"](#) on page 96

PULSe:MOP:BPSK:TYPE <Type>

Sets the modulation type.

Parameters:

<Type> STANdard | CONStant

Example: see [Example "Creating a BPSK pulse"](#) on page 397

Manual operation: See ["BPSK"](#) on page 96

PULSe:MOP:CHIRp:DEVIation <Deviation>

Sets the modulation deviation.

Parameters:

<Deviation> float
Range: 1 to 1e+09

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Linear Chirp"](#) on page 92

PULSe:MOP:CHIRp:TYPE <Type>

Selects the modulation type.

Parameters:

<Type> UP | DOWN | SINE | TRIangular

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Linear Chirp"](#) on page 92

PULSe:MOP:DATA:CODing <Coding>

Selects the data coding scheme.

Parameters:

<Coding> NONE | DIFFerential | GRAY | DGRay

Example: see [Using a plugin as a modulation source](#)

Manual operation: See ["Coding"](#) on page 100

PULSe:MOP:DATA:DSRC <Dsrc>

Selects the data source for the modulation, see [DSRC:CREate](#).

Parameters:

<Dsrc> string

Example: see [Using a plugin as a modulation source](#)

Manual operation: See ["Data Source"](#) on page 100

PULSe:MOP:ENABLE <Enable>

Defines whether a MOP is applied.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Enable Modulation on Pulse \(MOP\)"](#) on page 84

PULSe:MOP:EXCLude:ENABle <Enable>

Activates the restriction of the modulation.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: [Example "Creating an unmodulated pulse"](#) on page 395

Manual operation: See ["Restrict modulation to a certain area of the pulse"](#) on page 84

PULSe:MOP:EXCLude:LEVel:STARt <Start>

PULSe:MOP:EXCLude:LEVel:STOP <Stop>

Sets the threshold levels at the beginning and the end of a pulse for the modulation to be excluded.

Parameters:

<Stop> float
Range: 0 to 100

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Restrict modulation to a certain area of the pulse"](#) on page 84

PULSe:MOP:EXCLude:MODE <Mode>

Selects the parameter that determines the area on that the MOP is applied.

Parameters:

<Mode> TIME | LEVel | WIDTH

Example: See [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Restrict modulation to a certain area of the pulse"](#) on page 84

PULSe:MOP:EXCLude:TIME:STARt <Start>

PULSe:MOP:EXCLude:TIME:STOP <Stop>

Sets a time span to be excluded at the beginning and at the end of the pulse.

Parameters:

<Stop> float
Range: 0 to 5e-05

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Restrict modulation to a certain area of the pulse"](#) on page 84

PULSe:MOP:FILTer:BT <Bt>

Sets the B x T filter parameter.

Parameters:

<Bt> float
Range: 0.15 to 2.5

Example: see [Example "Using a plugin as a modulation source"](#) on page 390

Manual operation: See ["Filter"](#) on page 101

PULSe:MOP:FILTer:BWIDth <Bwidth>

Sets the transition bandwidth of the filter.

Parameters:

<Bwidth> float
Range: 1 to 1e+09

Example: see [Example "Using a plugin as a modulation source"](#) on page 390

Manual operation: See ["Filter"](#) on page 101

PULSe:MOP:FILTer:LENGth <Length>

Sets the filter length.

Parameters:

<Length> integer
Range: 1 to 64

Example: see [Example "Using a plugin as a modulation source"](#) on page 390

Manual operation: See ["Filter"](#) on page 101

PULSe:MOP:FILTer:ROLLoff <Rolloff>

Sets the roll off factor.

Parameters:

<Rolloff> float
Range: 0.05 to 1

Example: see [Example "Using a plugin as a modulation source"](#) on page 390

Manual operation: See ["Filter"](#) on page 101

PULSe:MOP:FILTer:TYPE <Type>

Selects the filter type.

Parameters:

<Type> NONE | RECTangular | COS | RCOS | GAUSs | LPASs |
FSKGauss | SOQPsk | SMWRect

Example: see [Example "Using a plugin as a modulation source"](#)
on page 390

Manual operation: See ["Filter"](#) on page 101

PULSe:MOP:FM:DEVIation <Deviation>

Sets the modulation deviation.

Parameters:

<Deviation> float
Range: 0.1 to 1e+09
Default unit: Hz

Manual operation: See ["FM"](#) on page 89

PULSe:MOP:FM:FREQuency <Frequency>

Sets the modulation frequency.

Parameters:

<Frequency> float
Range: 0.002 to 1e+09

Manual operation: See ["FM"](#) on page 89

PULSe:MOP:FSK:DEVIation <Deviation>

Sets the modulation deviation.

Parameters:

<Deviation> float
Range: 0.001 to 1e+09
Default unit: Hz

Manual operation: See ["FSK"](#) on page 89

PULSe:MOP:FSK:SRATe <Srate>

Sets the symbol rate of the modulated signal.

Parameters:

<Srate> float
Range: 1 to 1e+09

Manual operation: See ["FSK"](#) on page 89

PULSe:MOP:NLCHirp:EQUation <Equation>

Determines the chirp mathematically.

Parameters:

<Equation> string

Manual operation: See ["Equation"](#) on page 92

PULSe:MOP:NOISe:BWIDth <Bwidth>

Sets the bandwidth.

Parameters:

<Bwidth> float
Range: 1 to 1e+09
Default unit: Hz

Manual operation: see [Chapter 7.2.6.6, "Noise"](#), on page 98

PULSe:MOP:PCHirp:COEFFicient <Coefficient>

Sets the coefficient of the chirp polynomial.

Parameters:

<Coefficient> float
Range: -1e+22 to 1e+22

Manual operation: See ["Polynomial Chirp"](#) on page 92

PULSe:MOP:PCHirp:TERM <Term>

Sets the term of the chirp polynomial.

Parameters:

<Term> float
Range: 0 to 32

Manual operation: See ["Polynomial Chirp"](#) on page 92

PULSe:MOP:PLISt:VALue <Value>

Sets the phase.

Parameters:

<Value> float
 Range: -180 to 180
 Default unit: degree

Manual operation: See ["Custom Phase"](#) on page 96

PULSe:MOP:POLY:LENGth <Length>

Sets the polyphase length (code order).

Parameters:

<Length> integer
 Range: 1 to 100

Manual operation: See ["Poly Phase"](#) on page 95

PULSe:MOP:POLY:TYPE <Type>

Selects the modulation type.

Parameters:

<Type> FRANK | P1 | P2 | P3 | P4

Manual operation: See ["Poly Phase"](#) on page 95

PULSe:MOP:QPSK:SRATe <Srate>

Sets the symbol rate.

Parameters:

<Srate> float
 Range: 1 to 1e+09

Manual operation: See ["Symbol Rate"](#) on page 98

PULSe:MOP:QPSK:TYPE <Type>

Selects the modulation type.

Parameters:

<Type> NORMal | OQPSk | DQPSk | ASOQpsk | BSOQpsk | TGSQpsk

Manual operation: See ["QPSK Type"](#) on page 98

PULSe:MOP:QPSK:SOQPsk:IRIG <Irig>

Enables differential encoding according to the telemetry standard IRIG 106-04.

Parameters:

<Irig> ON | OFF | 1 | 0

Example:

```
PULSe:MOP:QPSK:TYPE TGSOQPSK
PULSe:MOP:QPSK:SRATE 1e+06
PULSe:MOP:QPSK:SOQPSk:IRIG 1
```

Manual operation: See ["QPSK Type"](#) on page 98

PULSe:MOP:TYPE <Type>

Select the modulation scheme.

Parameters:

<Type> AM | ASK | AMSTep | FM | FSK | FMSTep | CHIRp | NLCHirp | PCHirp | BARKer | POLYphase | PLISt | BPSK | QPSK | NOISe | PLUGin

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: see [Chapter 7.2.6, "Built-In Modulation Types and their Settings"](#), on page 86

PULSe:OVERshoot <Overshoot>

Sets the overshoot level value.

Parameters:

<Overshoot> float
Range: 0 to 50

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Overshoot"](#) on page 78

PULSe:OVERshoot:DECay <Decay>

Sets the number of peaks.

Parameters:

<Decay> float
Range: 1 to 100

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Overshoot"](#) on page 78

PULSe:RIPPIe <Ripple>

Sets the ripple level.

Parameters:

<Ripple> float
Range: 0 to 50

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Ripple"](#) on page 79

PULSe:RIPPLe:FREQuency <Frequency>

Sets the ripple frequency.

Parameters:

<Frequency> float
Range: 0 to 3e+08
Default unit: Hz

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Ripple"](#) on page 79

PULSe:TIME:FALL <Fall>

PULSe:TIME:RISE <Rise>

Sets the transition time of the rising and falling edges.

Parameters:

<Rise> float
Range: 0 to 3600

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Rising/Falling Edge, Width, Rising/Falling Slope"](#) on page 76

PULSe:TIME:REFerence <Reference>

Selects a predefined envelope profile.

Parameters:

<Reference> VOLTage | POWer | FULL

Example: see [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Standard Timing Profile"](#) on page 75

PULSe:TIME:WIDTh <Width>

Sets the time during that the pulse is on top power.

Parameters:

<Width> float
Range: 0 to 3600
Default unit: s

Example: See [Example "Creating a linear chirp pulse"](#) on page 396

Manual operation: See ["Rising/Falling Edge, Width, Rising/Falling Slope"](#) on page 76

PULSe:TYPE:FALL <Fall>

PULSe:TYPE:RISE <Rise>

Sets the slope type for the rising and falling edges.

Parameters:

<Rise> LINear | COSine | RCOSine | SQRT

Example: [Example "Creating an unmodulated pulse"](#) on page 395

Manual operation: See ["Rising/Falling Edge, Width, Rising/Falling Slope"](#)
on page 76

PULSe:SETTings <Settings>

Switches between the displayed settings.

Setting parameters:

<Settings> TIMing | LEVel | MOP | MKR | GENeral

Example: PULSe:SETTings MOP
PULSe:PREView:MODE MOP
PULSe:PREView:MOP IQ

Usage: Setting only

PULSe:PREView:MODE <Mode>

Switches between the envelope and modulation graphs.

Setting parameters:

<Mode> ENVelope | MOP

Example: see [PULSe:SETTings](#) on page 414

Usage: Setting only

Manual operation: See ["Envelope Graph"](#) on page 101

PULSe:PREView:MOP <Mop>

Sets the displayed modulation characteristics.

Setting parameters:

<Mop> IQ | PHASe | FREQuency

Example: see [PULSe:SETTings](#) on page 414

Usage: Setting only

Manual operation: See ["Time domain display"](#) on page 103

23.16 Receiver Commands

Example: Creating receivers

```

REceiver:CREate "Receiver DF"
REceiver:CATalog?
// "Receiver1", "Receiver DF"
REceiver:SElect?
// Receiver DF
REceiver:NAME?
// Receiver DF

REceiver:ANTenna:ADD
REceiver:ANTenna:SElect 1
REceiver:ANTenna:ALIAS "Antenna 1"
REceiver:ANTenna:POSition:X 0.14
REceiver:ANTenna:POSition:Y 0
REceiver:ANTenna:POSition:RADius 0.14
REceiver:ANTenna:POSition:ANGLE 0
REceiver:ANTenna:POSition:HEIGHt 0
ANTenna:CATalog?
// "Isotropic", "My_PencilBeam", "My_Cosecant", "My_PlanarAntenna", "CustomAntennPattern"
REceiver:ANTenna:PATtern "My_PlanarAntenna"
REceiver:ANTenna:GAIN 2
REceiver:ANTenna:DIRection:AWAY 1
REceiver:ANTenna:DIRection:ELEVation 0
REceiver:ANTenna:DIRection:AZIMuth?
// 0

REceiver:ANTenna:ADD
REceiver:ANTenna:SElect 2

REceiver:ANTenna:ALIAS "Antenna 2"
REceiver:ANTenna:POSition:X 0
REceiver:ANTenna:POSition:Y 0.14
REceiver:ANTenna:POSition:RADius 0.14
REceiver:ANTenna:POSition:ANGLE 90
REceiver:ANTenna:POSition:HEIGHt 0
REceiver:ANTenna:PATtern "My_PlanarAntenna"
REceiver:ANTenna:GAIN 0
REceiver:ANTenna:DIRection:AWAY 1
REceiver:ANTenna:DIRection:ELEVation 0
REceiver:ANTenna:DIRection:AZIMuth?
// 90

```

REceiver:ANTenna:ALIAS.....	416
REceiver:ANTenna:POSition:X.....	416
REceiver:ANTenna:POSition:Y.....	416
REceiver:ANTenna:POSition:RADius.....	416

REceiver:ANTenna:POStion:ANGLe.....	416
REceiver:ANTenna:POStion:HEIGHt.....	417
REceiver:ANTenna:PATTern.....	417
REceiver:ANTenna:GAIN.....	417
REceiver:ANTenna:DIRection:AWAY.....	417
REceiver:ANTenna:DIRection:AZIMuth.....	417
REceiver:ANTenna:DIRection:ELEVation.....	418

REceiver:ANTenna:ALias <Alias>

Sets an alias name for the selected antenna element.

Parameters:

<Alias> string

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Antenna Configuration"](#) on page 215

REceiver:ANTenna:POStion:X <X>

REceiver:ANTenna:POStion:Y <Y>

Sets the antenna element position as X and Y values, relative to the receiver origin.

Parameters:

<Y> float
 *RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:POStion:RADius <Radius>

Sets the distance from the antenna element to the receiver origin.

Parameters:

<Radius> float
 *RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:POStion:ANGLe <Angle>

Sets the antenna element position as an angle offset from the X-axis.

Parameters:

<Angle> float
 *RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:POSition:HEIGHT <Height>

Sets the antenna element height, relative to the receiver origin.

Parameters:

<Height> float
*RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:PATtern <Pattern>

Assigns an existing antenna pattern, see [ANTenna:CATalog?](#).

Parameters:

<Pattern> string

Example: See [Example "Creating receivers"](#) on page 415.

REceiver:ANTenna:GAIN <Gain>

Sets the gain of the individual antenna element.

Parameters:

<Gain> float
*RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:DIRection:AWAY <Away>

Sets the azimuth automatically, so that the beam axis is radial to the receiver origin.

Parameters:

<Away> ON | OFF | 1 | 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:DIRection:AZIMuth <Azimuth>

Turns the antenna beam axis.

Parameters:

<Azimuth> float
*RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

REceiver:ANTenna:DIRection:ELEVation <Elevation>

Turns the antenna beam axis.

Parameters:

<Elevation> float
*RST: 0

Example: See [Example "Creating receivers"](#) on page 415.

Manual operation: See ["Individual Antenna Element: Antenna x"](#) on page 216

23.17 Repository Commands



Do not change the default folder structure and the repository file names.

Changing file paths and file names may lead to data lost and irreparable faults in the data sources.

Example: Working with repositories

```

SCPI
// create new empty repository
REpository:CREate "Repository for tests"
REpository:CATalog?
//"Repository for tests","C:\Users\Public\Documents\Rohde-Schwarz\Pulse Sequencer\Repositories
//"K300 and K301 Tests","C:\Users\Public\Documents\Rohde-Schwarz\Pulse Sequencer\Repositories
//"Rep","C:\Users\Public\Documents\Rohde-Schwarz\Pulse Sequencer\Repositories"
REpository:SAVE
REpository:SElect "Repository for tests"
REpository:AUTHor "My Company Name"
REpository:DATE?
REpository:SECurity LEV1
REpository:VERSion "1.1"
REpository:COMplicity EMITter
REpository:SAVE
REpository:PATH?
// C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories\20140722_113521
REpository:FILEname?
// C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories
  \20140722_113521\Config.ps_rep
REpository:ACCess?
// RW,Login=no,Pass=no,Uname=testuser
REpository:REMove "Rep"

```

```

REPManager:PATH:LIST?
// "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories"
REPManager:PATH:ADD "C:\_ps_files"
REPManager:PATH:LIST?
// "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories","C:\_ps_files"
REPManager:CATalog?
// "Repository for tests",
   "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories",
// "K300 and K301 Tests",
   "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories",
// "My_Tests","C:\_ps_files"
REPManager:LOAD "K300 and K301 Tests"
REPManager:EXPort "K300 and K301 Tests","c:\_ps_files\my.psarch"
REPManager:DELeTe "My_Tests"
REPManager:CATalog?
// "Repository for tests",
   "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories",
// "K300 and K301 Tests",
   "C:\Users\Public\Documents\Rohde-Schwarz\PulseSequencer\Repositories"

REPository:ACCess?..... 419
REPository:AUTHor..... 420
REPository:COMPLexity..... 420
REPository:DATE..... 420
REPository:FILEname?..... 420
REPository:PATH?..... 421
REPository:SAVE..... 421
REPository:SECurity..... 421
REPository:VERSion..... 421
REPository:UUID?..... 422
REPManager:CATalog?..... 422
REPManager:LOAD..... 422
REPManager:DELeTe..... 423
REPManager:EXPort..... 423
REPManager:PATH:ADD..... 424
REPManager:PATH:DELeTe..... 424
REPManager:PATH:LIST?..... 424

```

REPository:ACCess?

Queries information on the access rights of the current user.

Return values:

<Access> <permission>, <login>, <pass>, <Uname>
<permission>
 Permission of the current user, for example RW (read-write)
<login>, <pass>
 Login/Pass=No: Password not required
 Login/Pass=Yes: Password required
<Uname>
 User name of the current user

Example: See [Example "Working with repositories"](#) on page 418

Usage: Query only

Manual operation: See ["Users Management"](#) on page 44

REPOSITORY:AUTHor <Author>

Enters information on the author.

Parameters:

<Author> string

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["General Repository Settings"](#) on page 43

REPOSITORY:COMPLexity <Complexity>

Sets the complexity level.

Parameters:

<Complexity> PTRain | EMITter | DIRection

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["General Repository Settings"](#) on page 43

REPOSITORY:DATE <Date>

Queries the creation data.

Parameters:

<Date> string

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["General Repository Settings"](#) on page 43

REPOSITORY:FILEname?

Queries the file name of the repository archive.

Return values:

<Filename> string
File path, incl. file name, and extension

Example: See [Example "Working with repositories"](#) on page 418

Usage: Query only

Manual operation: See ["Storage"](#) on page 45

REPOSITORY:PATH?

Queries the directory in that the repository archive is stored.

Return values:

<Path> string

Example: see [Example "Working with repositories"](#) on page 418

Usage: Query only

Manual operation: See ["Storage"](#) on page 45

REPOSITORY:SAVE

Stores the repository archive.

To query the storage location, use the command [REPOSITORY:PATH?](#).

Example: see [Example "Working with repositories"](#) on page 418

Usage: Event

Manual operation: See ["Storage"](#) on page 45

REPOSITORY:SECURITY <Security>

Sets the security level.

Parameters:

<Security> LEV0 | LEV1 | LEV2 | LEV3 | LEV4

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["General Repository Settings"](#) on page 43

REPOSITORY:VERSION <Version>

Sets the repository version.

Parameters:

<Version> string

Example: see [Example "Working with repositories"](#) on page 418

Manual operation: See ["General Repository Settings"](#) on page 43

REPOSITORY:UUID?

Queries the repository's Universally Unique Identifier (UUID).

Return values:

<Uuid> string

Example:

```
REPOSITORY:UUID?
// 89d00568-e9d7-4470-8b8d-2179336da541
```

Usage: Query only

REPMANAGER:CATALOG?

Queries available repository elements in the database.

Return values:

<Catalog> "<RepositoryName>","<path>"

<RepositoryName> is the name of the repository as defined
with the command [REPOSITORY:CREATE](#)
<Path> is the complete file path

Example: see [Example "Working with repositories"](#) on page 418

Usage: Query only

Manual operation: See ["Discovered Repositories on the Mass Storage"](#) on page 48
See ["Add Path, Add Install Path, Add Home Path"](#) on page 49

REPMANAGER:LOAD <RepName>[,<Path>[,<Username>,<Passwd>]]

Loads the selected repository to the workspace.

If more than one repository with the same name exist, loaded is the first repository with a name match.

To query the available repository elements in the database, use the command [REPOSITORY:CATALOG?](#).

Setting parameters:

<RepName> string
Repository name, as configured in the workspace.

<Path> string
Complete file path, as queried with the command [REPMANAGER:PATH:LIST?](#).
The <Path> must be specified, if <Username> and <Passwd> are used.

<Username> string
Required if the repository is password protected

<Passwd> string
Required if the repository is password protected

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["Load"](#) on page 48

REPManager:DELeTe <RepName>[,<Path>[,<Username>,<Passwd>]]

Deletes the entire repository from the permanent mass storage.

Setting parameters:

<RepName> string
Repository name, as configured in the workspace.
If more than one repository with the same name exist, the <Path> must be specified.

<Path> string
Complete file path, as queried with the command [REPManager:PATH:LIST?](#).
The <Path> must be specified, if the <RepName> is not unique and if <Username> and <Passwd> are used.

<Username> string
Required if the repository is password protected

<Passwd> string
Required if the repository is password protected

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["Delete"](#) on page 49

REPManager:EXPort <RepName>[,<Path>],<PSArchiveFile>

Exports the selected repository file to an archive file.

Setting parameters:

<RepName> string
Repository name, as configured in the workspace.

<Path> string
Complete file path, as queried with the command [REPManager:PATH:LIST?](#).

<PSArchiveFile> Complete file path, incl. file name, and extension (*.psarch)

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["Export"](#) on page 48

REPManager:PATH:ADD <Add>

Add the selected directory.

Setting parameters:

<Add> string
 Complete file path

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["Add Path, Add Install Path, Add Home Path"](#) on page 49

REPManager:PATH:DELeTe <Delete>

Removes the selected file path.

Setting parameters:

<Delete> string
 File path

Example: See [Example "Working with repositories"](#) on page 418

Usage: Setting only

Manual operation: See ["Remove Path"](#) on page 49

REPManager:PATH:LIST?

Queries the directory in that the repository files are stored.

Return values:

<List> string
 Complete file path

Example: See [Example "Working with repositories"](#) on page 418

Usage: Query only

23.18 Scenario Commands

Example: Creating simple pulse train scenario

```

SCPI
SCENario:CREate "SimplePulseTrain"
SCENario:TYPE SEquence
SCENario:ID?
// 4
SCENario:OUTPut:RF:ENABle 1
SCENario:OUTPut:FREQuency 6e+09
SCENario:OUTPut:LEVel -10
SCENario:OUTPut:RESet:ENABle 1
SCENario:OUTPut:RUNMode CONTinuous
SEquence:CATalog?
// "My_S1_PT","My_S2_S1-F1_S1-F2","My_PT_PRI-Stagger","My_PT_F-Hops",
// "My_WV_Seq","My_TestSequence"
SCENario:SEQuence "My_S1_PT"
SCENario:OUTPut:TARGet INSTRument
GENerator:CATalog?
// "My_R&S SMW","My_R&S SMBV","R&S SMW"
SCENario:GENerator "R&S SMW"
SCENario:GENerator:PATH?
// 1
SCENario:CACH:REPository:ENABle 1

SCENario:OUTPut:CLOCK:MODE AUTO
SCENario:OUTPut:CLOCK:AUTO:OVERsampling 8
SCENario:OUTPut:CLOCK:AUTO:BORDER 1000000
// SCENario:OUTPut:CLOCK:MODE MAN
// SCENario:OUTPut:CLOCK:USER?
// 1e+07
SCENario:OUTPut:DURation:MODE SEquence
// SCENario:OUTPut:DURation:TIME 0.10
SCENario:OUTPut:FORMat WV
SCENario:OUTPut:THReshold -100
SCENario:OUTPut:INSTRument:QHIG:ENABle 1
SCENario:OUTPut:RECall:ENABle 1

SCENario:CALCulate
SCENario:STATe?
// RUN
SYSTEM:PROGress?
// 74
SYSTEM:PROGress?
// 100
SCENario:CACH:VOLatile:VALid?
// 1
SCENario:CACH:VOLatile:RELease

```

```
// SCENario:CAChE:VOLatile:CLear
// SCENario:STOP
// SCENario:SANitize REPository
```

```
SCENario:START
```

Example: Creating sequence collection scenario

```
SCPI
SCENario:CREate "My_PT_Collection"
SCENario:TYPE CSEquence
SCENario:CSEquence:ADD
SCENario:CSEquence:SElect 1
SCENario:CSEquence:ALias "PT"
SCENario:CSEquence "My_S2_S1-F1_S1-F2"
SCENario:CSEquence:ADD
SCENario:CSEquence:SElect 2
SCENario:CSEquence:ALias "F-Hops"
SCENario:CSEquence "My_PT_F-Hops"
SCENario:CSEquence:CURRent 1
...
SCENario:CALCulate
```

Example: Creating a simple emitter scenario

```
SCPI
SCENario:CREate "SimpleEmitterScenario"
SCENario:TYPE EMITter
EMITter:CATalog?
// "My_EmitterGuidance","TestEmitter"
SCENario:EMITter "My_EmitterGuidance"
SCENario:EMITter:MODE 2
SCENario:EMITter:MODE:BEAM 1
SCENario:EMITter:DIRection:YAW -10
SCENario:EMITter:DIRection:PITCh -5
...
SCENario:OUTPut:TARGet FILE
SCENario:OUTPut:PATH "C:\My_ARB_Files"
SCENario:CALCulate
```

Example: Creating a scenario with multiple emitters, interferes and background emitters

```

SCPI
SCENario:CREate "LocalizedEmitters"
SCENario:TYPE Localized
EMITter:CATalog?
// "My_EmitterGuidance","TestEmitter"
WAVeform:CATalog?
// "My_Waveform","WV_Matlab","CW","AWGN"
ANTenna:CATalog?
// "Isotropic","My_PencilBeam","My_Cosecant","My_PlanarAntenna","Testantenna"
SCAN:CATalog?
// "My_RasterScan","My_Circular","Test Antenna Scan"

SCENario:LOCALized:RECEiver:ANTenna "My_Cosecant"
SCENario:LOCALized:RECEiver:SCAN "My_Circular"
SCENario:LOCALized:RECEiver:GAIN 5
SCENario:LOCALized:RECEiver:HEIGHT 1
SCENario:LOCALized:RECEiver:DIRection:YAW 0
SCENario:LOCALized:RECEiver:DIRection:PITCh -5

SCENario:LOCALized:ADD
SCENario:LOCALized:SElect?
// 4
SCENario:LOCALized:SElect 5
SCENario:LOCALized:TYPE WAVeform
SCENario:LOCALized:ALias "I3"
SCENario:LOCALized:WAVeform "My_Waveform"
SCENario:LOCALized:WAVeform:EIRP 50
SCENario:LOCALized:WAVeform:FREQuency 3e+09
SCENario:LOCALized:WAVeform:ANTenna "Testantenna"
SCENario:LOCALized:WAVeform:SCAN "Test Antenna Scan"
SCENario:LOCALized:DIRection:TRACk 1
SCENario:LOCALized:DIRection:PITCh 0
SCENario:LOCALized:DIRection:YAW -53.3035
SCENario:LOCALized:LOCation:EAST 2860.34
SCENario:LOCALized:LOCation:NORTH -3837.93
SCENario:LOCALized:DISTance?
// 4786.57

SCENario:LOCALized:ADD
SCENario:LOCALized:SElect 6
SCENario:LOCALized:TYPE EMITter
SCENario:LOCALized:ALias "E4"
SCENario:LOCALized:EMITter "My_EmitterGuidance"
SCENario:LOCALized:EMITter:MODE 1
SCENario:LOCALized:EMITter:MODE:BEAM 2
SCENario:LOCALized:DIRection:TRACk 1
SCENario:LOCALized:DIRection:PITCh 0
SCENario:LOCALized:DIRection:YAW -87.8901

```

```

SCENario:LOCalized:LOCation:EAST 253.448
SCENario:LOCalized:LOCation:NORTH -6879.31
SCENario:LOCalized:DIStance?
// 6883.98

SCENario:LOCalized:ADD
SCENario:LOCalized:SElect 7
SCENario:LOCalized:TYPE BKG
SCENario:LOCalized:ALias "BKG"
WAVeform:CATalog?
// "My_Waveform","WV_Matlab","CW","Noise","MC","BKG"
SCENario:LOCalized:WAVeform "BKG"

SCENario:LOCalized:ADD
SCENario:LOCalized:SElect 7
SCENario:LOCalized:TYPE BKG
SCENario:LOCalized:ALias "PT"
SEquence:CATalog?
// "My_S2_S1-F1_S1-F2","My_PT_PRI-Stagger","My_PT_F-Hops",
// "My_TestSequence","My_PulseTrain_Seq"
SCENario:LOCalized:SEquence "My_PulseTrain_Seq"

...
SCENario:CALCulate
    
```

SCENario:TYPE.....	430
SCENario:ID?.....	431
SCENario:CALCulate.....	431
SCENario:STOP.....	431
SCENario:STARt.....	431
SCENario:STATe?.....	431
SCENario:CACHe:REPository:ENABle.....	431
SCENario:CACHe:REPository:VALid?.....	432
SCENario:CACHe:VOLatile:VALid?.....	432
SCENario:CACHe:VOLatile:RELease.....	432
SCENario:CACHe:VOLatile:REStore.....	432
SCENario:CACHe:REPository:CLEar.....	432
SCENario:CACHe:VOLatile:CLEar.....	432
SCENario:GENerator.....	433
SCENario:GENerator:PATH.....	433
SCENario:SEquence.....	433
SCENario:SEquence:CLEar.....	433
SCENario:DIRection:ALias.....	433
SCENario:CEMit:ALias.....	433
SCENario:CSEquence:ALias.....	433
SCENario:LOCalized:ALias.....	433
SCENario:DIRection:CURRent.....	434
SCENario:CEMit:CURRent.....	434
SCENario:LOCalized:CURRent.....	434
SCENario:CSEquence:CURRent.....	434

SCENario:DIRection:DIRection:YAW.....	434
SCENario:EMITter:DIRection:YAW.....	434
SCENario:CEMit:DIRection:YAW.....	434
SCENario:LOCalized:DIRection:YAW.....	434
SCENario:DIRection:RECeiver:DIRection:YAW.....	434
SCENario:LOCalized:RECeiver:DIRection:YAW.....	434
SCENario:DIRection:DIRection:PITCh.....	434
SCENario:DIRection:RECeiver:DIRection:PITCh.....	434
SCENario:EMITter:DIRection:PITCh.....	434
SCENario:CEMit:DIRection:PITCh.....	434
SCENario:LOCalized:RECeiver:DIRection:PITCh.....	435
SCENario:LOCalized:DIRection:PITCh.....	435
SCENario:DIRection:DIRection:ROLL.....	435
SCENario:DIRection:RECeiver:DIRection:ROLL.....	435
SCENario:EMITter:DIRection:ROLL.....	435
SCENario:CEMit:DIRection:ROLL.....	435
SCENario:LOCalized:RECeiver:DIRection:ROLL.....	435
SCENario:LOCalized:DIRection:ROLL.....	435
SCENario:DIRection:EMITter.....	435
SCENario:EMITter.....	435
SCENario:CEMit:EMITter.....	435
SCENario:LOCalized:EMITter.....	435
SCENario:DIRection:EMITter:MODE.....	435
SCENario:EMITter:MODE.....	435
SCENario:CEMit:EMITter:MODE.....	435
SCENario:LOCalized:EMITter:MODE.....	435
SCENario:DIRection:EMITter:MODE:BEAM.....	436
SCENario:EMITter:MODE:BEAM.....	436
SCENario:CEMit:EMITter:MODE:BEAM.....	436
SCENario:LOCalized:EMITter:MODE:BEAM.....	436
SCENario:DIRection:DIRection:TRACk.....	436
SCENario:LOCalized:DIRection:TRACk.....	436
SCENario:DIRection:DISTance?.....	436
SCENario:LOCalized:DISTance?.....	436
SCENario:DIRection:LOCation:EAST.....	437
SCENario:DIRection:LOCation:NORTH.....	437
SCENario:LOCalized:LOCation:EAST.....	437
SCENario:LOCalized:LOCation:NORTH.....	437
SCENario:DIRection:LOCation:AZIMuth.....	437
SCENario:LOCalized:LOCation:AZIMuth.....	437
SCENario:DIRection:LOCation:ELEVation.....	438
SCENario:LOCalized:LOCation:ELEVation.....	438
SCENario:DIRection:RECeiver:HEIGHt.....	438
SCENario:DIRection:LOCation:HEIGHt.....	438
SCENario:LOCalized:RECeiver:HEIGHt.....	438
SCENario:LOCalized:LOCation:HEIGHt.....	438
SCENario:DIRection:WAVeform:ANTenna.....	438
SCENario:LOCalized:RECeiver:ANTenna.....	438
SCENario:LOCalized:WAVeform:ANTenna.....	438
SCENario:DIRection:WAVeform:SCAN.....	438

SCENario:LOCalized:RECeiver:SCAN.....	438
SCENario:LOCalized:WAVeform:SCAN.....	438
SCENario:DIRection:TYPE.....	439
SCENario:LOCalized:TYPE.....	439
SCENario:LOCalized:RECeiver:GAIN.....	439
SCENario:DIRection:SEQuence.....	439
SCENario:LOCalized:SEQuence.....	439
SCENario:DIRection:WAVeform.....	439
SCENario:LOCalized:WAVeform.....	439
SCENario:DIRection:WAVeform:EIRP.....	440
SCENario:LOCalized:WAVeform:EIRP.....	440
SCENario:DIRection:WAVeform:FREQuency.....	440
SCENario:LOCalized:WAVeform:FREQuency.....	440
SCENario:DIRection:SYNChronize:ENABLE.....	440
SCENario:LOCalized:SYNChronize:ENABLE.....	440
SCENario:DIRection:RECeiver.....	440
SCENario:CSEQuence.....	441
SCENario:CSEQuence:VARIable.....	441
SCENario:OUTPut:CLOCK:MODE.....	441
SCENario:OUTPut:CLOCK:AUTO:OVERsampling.....	441
SCENario:OUTPut:CLOCK:AUTO:BORDER.....	442
SCENario:OUTPut:CLOCK:USER.....	442
SCENario:OUTPut:DURation:MODE.....	442
SCENario:OUTPut:DURation:TIME.....	442
SCENario:OUTPut:FORMat.....	442
SCENario:OUTPut:INSTrument:QHIG:ENABLE.....	443
SCENario:OUTPut:RECall:ENABLE.....	443
SCENario:OUTPut:FREQuency.....	443
SCENario:OUTPut:LEVel.....	443
SCENario:OUTPut:PATH.....	443
SCENario:OUTPut:RESet:ENABLE.....	444
SCENario:OUTPut:RF:ENABLE.....	444
SCENario:OUTPut:RUNMode.....	444
SCENario:OUTPut:TARGet.....	444
SCENario:OUTPut:THReshold.....	445
SCENario:VOLatile:VIEW.....	445
SCENario:SANitize.....	445

SCENario:TYPE <Type>

Sets the scenario type.

Parameters:

<Type> SEQuence | CSEQuence | EMITter | CEMitter | LOCalized | WAVeform | DIRection

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Scenario Type"](#) on page 56

SCENario:ID?

Queries the database identifier of the selected scenario.

Return values:

<Id> float
*RST: 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Usage: Query only

SCENario:CALCulate

Starts the signal calculation.

Example: See [Example "Creating simple pulse train scenario"](#) on page 425.

Usage: Event

Manual operation: See ["Calculate + Deploy, Start/Stop, Busy/Idle"](#) on page 59

SCENario:STOP**SCENario:START**

Starts/stops signal generation.

Example: See [Example "Creating simple pulse train scenario"](#) on page 425.

Usage: Event

Manual operation: See ["Calculate + Deploy, Start/Stop, Busy/Idle"](#) on page 59

SCENario:STATe?

Queries the current scenario status.

Return values:

<State> IDLE | RUN

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Usage: Query only

Manual operation: See ["Calculate + Deploy, Start/Stop, Busy/Idle"](#) on page 59

SCENario:CACHe:REPository:ENABle <Enable>

Enables file storage in the repository.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:CAChE:REPository:VALid?

SCENario:CAChE:VOLatile:VALid?

Queries whether the volatile/repository memory contains a valid waveform.

Return values:

<Valid> ON | OFF | 1 | 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Usage: Query only

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:CAChE:VOLatile:RELEase

Exports and stores the waveform files in the repository.

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Usage: Event

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:CAChE:VOLatile:RESTore

Loads waveform from the storage.

Example:

```
SCENario:CAChE:REPository:ENABle 1
SCENario:CAChE:VOLatile:VALid?
// 1
SCENario:CAChE:VOLatile:RELEase
SCENario:CAChE:VOLatile:RESTore
```

Usage: Event

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:CAChE:REPository:CLEar

SCENario:CAChE:VOLatile:CLEar

Deletes the files from the volatile/repository memory.

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Usage: Event

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:GENerator <Generator>

Sets the signal generator.

Parameters:

<Generator> string
Use the command [GENerator:CATalog?](#) to query a list of configured generator profiles.

Example: See [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Upload to VSG \(Vector Signal Generator\)"](#) on page 60

SCENario:GENerator:PATH <Path>

Selects the signal path that will play the generated signal.

Parameters:

<Path> float
Range: 1 to 32

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Path \(RF output connector\)"](#) on page 58
See ["Upload to VSG \(Vector Signal Generator\)"](#) on page 60

SCENario:SEQuence <Sequence>

Assigns a pulse sequence, see [SEQuence:CATalog?](#) on page 329.

Parameters:

<Sequence> string

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Single Sequence and Waveform Scenario Settings"](#) on page 63

SCENario:SEQuence:CLEar

Usage: Event

Manual operation: See ["Single Sequence and Waveform Scenario Settings"](#) on page 63

SCENario:DIRection:ALias <Alias>**SCENario:CEMit:ALias** <Alias>**SCENario:CSEQuence:ALias** <Alias>**SCENario:LOCalized:ALias** <Alias>

Enters an alias name.

Parameters:

<Alias> string

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRectio:n:CURRent <Current>

SCENario:CEMit:CURRent <Current>

SCENario:LOCalized:CURRent <Current>

SCENario:CSEquence:CURRent <Current>

Sets the sequence/emitter that is used by the scenario.

Parameters:

<Current> float

Number of the sequence/emitter in the list with multiple sequences

*RST: 1

Example: See [Example "Creating sequence collection scenario"](#) on page 426

Manual operation: See ["Sequence \(Collection\) Scenario Settings"](#) on page 64
See ["List of sequences"](#) on page 123

SCENario:DIRectio:n:DIRectio:n:YAW <Yaw>

SCENario:EMITter:DIRectio:n:YAW <Yaw>

SCENario:CEMit:DIRectio:n:YAW <Yaw>

SCENario:LOCalized:DIRectio:n:YAW <Yaw>

SCENario:DIRectio:n:RECeiver:DIRectio:n:YAW <Yaw>

SCENario:LOCalized:RECeiver:DIRectio:n:YAW <Yaw>

Sets the yaw.

Parameters:

<Yaw> float

Range: 0 to 360

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Attitude, Position"](#) on page 213

SCENario:DIRectio:n:DIRectio:n:PITCh <Pitch>

SCENario:DIRectio:n:RECeiver:DIRectio:n:PITCh <Pitch>

SCENario:EMITter:DIRectio:n:PITCh <Pitch>

SCENario:CEMit:DIRectio:n:PITCh <Pitch>

SCENario:LOCalized:RECeiver:DIRection:PITCh <Pitch>
SCENario:LOCalized:DIRection:PITCh <Pitch>

Sets the pitch.

Parameters:

<Pitch> float
 Range: -90 to 90
 Default unit: grad

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Attitude"](#) on page 257

SCENario:DIRection:DIRection:ROLL <Roll>
SCENario:DIRection:RECeiver:DIRection:ROLL <Roll>
SCENario:EMITter:DIRection:ROLL <Roll>
SCENario:CEMit:DIRection:ROLL <Roll>
SCENario:LOCalized:RECeiver:DIRection:ROLL <Roll>
SCENario:LOCalized:DIRection:ROLL <Roll>

Sets the roll.

Parameters:

<Roll> float
 Range: 0 to 360

Manual operation: See ["Attitude"](#) on page 257

SCENario:DIRection:EMITter <Emitter>
SCENario:EMITter <Emitter>
SCENario:CEMit:EMITter <Emitter>
SCENario:LOCalized:EMITter <Emitter>

Assigns an existing emitter or an existing waveform, see [WAVEform:CATalog?](#) and [EMITter:CATalog?](#).

Parameters:

<Emitter> string

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:EMITter:MODE <Mode>
SCENario:EMITter:MODE <Mode>
SCENario:CEMit:EMITter:MODE <Mode>
SCENario:LOCalized:EMITter:MODE <Mode>

Set the emitter mode.

Parameters:

<Mode> float
Range: 1 to 16

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:EMITter:MODE:BEAM <Beam>

SCENario:EMITter:MODE:BEAM <Beam>

SCENario:CEMit:EMITter:MODE:BEAM <Beam>

SCENario:LOCalized:EMITter:MODE:BEAM <Beam>

Sets the used beam of the current mode.

Parameters:

<Beam> float
Range: 1 to 32

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:DIRection:TRACk <Track>

SCENario:LOCalized:DIRection:TRACk <Track>

Turns the antenna in the direction of the receiver.

Parameters:

<Track> ON | OFF | 1 | 0

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Attitude"](#) on page 257

SCENario:DIRection:DISTance?

SCENario:LOCalized:DISTance?

Sets the distance to the receiver.

Return values:

<Distance> float
Range: 0 to 1e+09
Default unit: m

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Usage: Query only

Manual operation: See ["Position"](#) on page 255

SCENario:DIRection:LOCation:EAST <East>
SCENario:DIRection:LOCation:NORTH <North>
SCENario:LOCalized:LOCation:EAST <East>
SCENario:LOCalized:LOCation:NORTH <North>

Sets the emitter coordinates.

Parameters:

<North> float
 Range: -1e+06 to 1e+06
 Default unit: m

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Position"](#) on page 255

SCENario:DIRection:LOCation:AZIMuth <Azimuth>
SCENario:LOCalized:LOCation:AZIMuth <Azimuth>

Sets the azimuth.

Parameters:

<Azimuth> float
 Range: 0 to 360

Example:

```

SCENario:LOCalized:LOCation:NORTH 7000
SCENario:LOCalized:LOCation:EAST 0
SCENario:LOCalized:LOCation:ALTitude 0

SCENario:LOCalized:DISTance?
// 7000
SCENario:LOCalized:LOCation:AZIMuth?
// 0
SCENario:LOCalized:LOCation:ELEVation?
// 0

SCENario:LOCalized:LOCation:AZIMuth 90
SCENario:LOCalized:LOCation:NORTH?
// 4.28626e-13
SCENario:LOCalized:LOCation:EAST?
// 7000

SCENario:LOCalized:LOCation:ALTitude 10
SCENario:LOCalized:LOCation:ELEVation?
// 0.0818511
SCENario:LOCalized:DISTance?
// 7000.01

```

Manual operation: See ["Position"](#) on page 255

SCENario:DIRection:LOCation:ELEVation <Elevation>
SCENario:LOCalized:LOCation:ELEVation <Elevation>

Sets the elevation.

Parameters:

<Elevation> float
Range: -90 to 90

Example: see [SCENario:LOCalized:LOCation:AZIMuth](#)
on page 437

Manual operation: See "[Position](#)" on page 255

SCENario:DIRection:RECeiver:HEIGht <Height>
SCENario:DIRection:LOCation:HEIGht <Height>
SCENario:LOCalized:RECeiver:HEIGht <Height>
SCENario:LOCalized:LOCation:HEIGht <Height>

Sets the height of the antenna.

Parameters:

<Height> float
Range: -1e+06 to 1e+06

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See "[Position](#)" on page 255

SCENario:DIRection:WAVEform:ANTenna <Antenna>
SCENario:LOCalized:RECeiver:ANTenna <Antenna>
SCENario:LOCalized:WAVEform:ANTenna <Antenna>

Assigns an existing antenna pattern, see [ANTenna:CATalog?](#) on page 329.

Parameters:

<Antenna> string

Example: See [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See "[Properties of the selected Emitter/Interferer](#)" on page 254

SCENario:DIRection:WAVEform:SCAN <Scan>
SCENario:LOCalized:RECeiver:SCAN <Scan>
SCENario:LOCalized:WAVEform:SCAN <Scan>

Assigns an existing antenna scan, see [SCAN:CATalog?](#) on page 329.

Parameters:

<Scan> string

Example: See [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:TYPE <Type>

SCENario:LOCalized:TYPE <Type>

Defines whether an emitter or an interferer is configured.

Parameters:

<Type> EMITter | WAVeform

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:LOCalized:RECeiver:GAIN <Gain>

Sets the antenna [Gain](#).

Parameters:

<Gain> float
Range: -120 to 120

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Gain"](#) on page 213

SCENario:DIRection:SEQuence <Sequence>

SCENario:LOCalized:SEQuence <Sequence>

Assigns a sequene to the background signal.

Parameters:

<Sequence> string

Example: See [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Background"](#) on page 250

SCENario:DIRection:WAVeform <Waveform>

SCENario:LOCalized:WAVeform <Waveform>

Assigns an existing emitter or an existing waveform, see [WAVeform:CATalog?](#) and [EMITter:CATalog?](#).

Parameters:

<Waveform> string

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:WAVeform:EIRP <Eirp>

SCENario:LOCalized:WAVeform:EIRP <Eirp>

Sets the [EIRP](#) of the interferer.

Parameters:

<Eirp> float
Range: -200 to 200

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:WAVeform:FREQuency <Frequency>

SCENario:LOCalized:WAVeform:FREQuency <Frequency>

Sets the frequency of the interferer.

Parameters:

<Frequency> float
Range: 1000 to 4.4e+10

Example: see [Example "Creating a scenario with multiple emitters, interferes and background emitters"](#) on page 427

Manual operation: See ["Properties of the selected Emitter/Interferer"](#) on page 254

SCENario:DIRection:SYNChronize:ENABle <Enable>

SCENario:LOCalized:SYNChronize:ENABle <Enable>

Enables synchronized setup.

Parameters:

<Enable> ON | OFF | 1 | 0

Manual operation: See ["Synchronized Setup"](#) on page 279

SCENario:DIRection:RECeiver <Receiver>

Selects an existing receiver, see [RECeiver:CATalog?](#) on page 329.

Parameters:

<Receiver> string

Example:

```
RECeiver:CATalog?
// "Receiver DF", "DF"
SCENario:DIRection:RECeiver "Receiver DF"
```


Manual operation: See ["Properties of the selected Receiver"](#) on page 217

SCENario:CSEquence <Csequence>

Select an excising sequence, see [SEquence:CATalog?](#) on page 329.

Parameters:

<Csequence> string

Example: See [Example "Creating sequence collection scenario"](#) on page 426

Manual operation: See ["Sequence \(Collection\) Scenario Settings"](#) on page 64
See ["Sequence"](#) on page 123

SCENario:CSEquence:VARiable <Variable>

Sets the collection variable.

Parameters:

<Variable> string

Example: SCENario:CSEquence:VARiable?

Manual operation: See ["Collection Variables"](#) on page 124

SCENario:OUTPut:CLOCK:MODE <Mode>

Sets the clock mode.

Parameters:

<Mode> AUTO | MANual

AUTO

Clock rate is retrieved from the generated waveform

MANual

Clock rate is user-defined

Example: See [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:CLOCK:AUTO:OVERsampling <Oversampling>

Sets the minimum oversampling factor.

Parameters:

<Oversampling> float

Range: 1 to 1000

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:CLOCK:AUTO:BORDER <Border>

Sets the minimum clock rate.

Parameters:

<Border> float
Range: 1000 to 1e+08

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:CLOCK:USER <User>

Sets a user defined clock rate.

Parameters:

<User> float
Range: 1 to 2.4e+09

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:DURation:MODE <Mode>

Defines the waveform duration.

Parameters:

<Mode> SEquence | MANual | SCAN

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:DURation:TIME <Time>

Sets the duration of the generated waveform.

Parameters:

<Time> float
Range: 1e-06 to 3600

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Clock, Duration"](#) on page 274

SCENario:OUTPut:FORMat <Format>

Sets the type of the generated waveform file.

Parameters:

<Format> WV | MSW

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Output"](#) on page 273

SCENario:OUTPut:INSTrument:QHIG:ENABLE <Enable>

Sets the I/Q modulator of signal generator to work in a high quality mode.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Output"](#) on page 273

SCENario:OUTPut:RECall:ENABLE <Enable>

Stores the current signal generator configuration as a save/recall file.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Output"](#) on page 273

SCENario:OUTPut:FREQuency <Frequency>

Sets the carrier RF frequency of the generated signal.

Parameters:

<Frequency> float
Range: 1000 to 4.4e+10

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Frequency"](#) on page 57

SCENario:OUTPut:LEVel <Level>

Sets the reference level used by the calculation of the pulse envelope.

Parameters:

<Level> float
Range: -200 to 30

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Ref. Level"](#) on page 57

SCENario:OUTPut:PATH <Path>

Sets the directory the generated waveform is stored in.

Parameters:

<Path> string
File path

Example: See [Example "Creating a simple emitter scenario"](#) on page 426

Manual operation: See ["Load ARB File"](#) on page 62

SCENario:OUTPut:RESet:ENABle <Enable>

Restarts the connected instrument on scenario start.

Parameters:

<Enable> ON | OFF | 1 | 0

Example: See [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Start with Reset"](#) on page 58

SCENario:OUTPut:RF:ENABle <Enable>**Parameters:**

<Enable> ON | OFF | 1 | 0

Manual operation: See ["Set RF frequency and level on target instrument"](#) on page 57

SCENario:OUTPut:RUNMode <Runmode>

Defines the way the generated signal is processed.

Parameters:

<Runmode> CONTinuous | SINGLE

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Playback mode"](#) on page 58

SCENario:OUTPut:TARGet <Target>

Defines whether the software creates an ARB file or transfers the generated waveform to a connected physical generator.

To assign a generator profile, use the command [SCENario:GENerator](#) on page 433.

To set the name and the directory the ARB file is stored in, use the command [SCENario:OUTPut:PATH](#).

Parameters:

<Target> INSTrument | FILE

Firmware/Software: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Upload to VSG \(Vector Signal Generator\)"](#) on page 60

SCENario:OUTPut:THReshold <Threshold>

Sets a threshold. Pulses at levels below this threshold are omitted.

Parameters:

<Threshold> float
Range: -100 to 0

Example: see [Example "Creating simple pulse train scenario"](#) on page 425

Manual operation: See ["Output"](#) on page 273

SCENario:VOLatile:VIEW

If a waveform exists in the volatile memory, opens the "Waveform Viewer" and displays this waveform.

Usage: Event

Manual operation: See ["Volatile, Repository"](#) on page 59

SCENario:SANitize <Sanitize>

Removes uploaded waveforms from the hard disk of the signal generator.

Setting parameters:

<Sanitize> SCENario | REPository | ALL

SCENario

Removes the current scenario waveforms

REPository

Removes the waveforms of all scenarios from the current repository

ALL

Removes all waveforms created by the R&S Pulse Sequencer

Example: SCENario:SANitize SCENario

Usage: Setting only

Manual operation: See ["Upload to VSG \(Vector Signal Generator\)"](#) on page 60

23.19 Sequence Commands

In the following examples, we assume that a pulse-base sequence "Test Sequence" and the required Pulses have been created.

See also:

- [Example "Handling items"](#) on page 325
- [Example "Creating an unmodulated pulse"](#) on page 395

Example: Creating a CW segment

```

SCPI
SEquence:CREate "Test Sequence"
SEquence:ITEM:ADD
SEquence:ITEM COUNT?
// 2
SEquence:ITEM:SElect 2
SEquence:ITEM:TYPE FILLer
SEquence:ITEM:FILLer:SIGNAL CW
SEquence:ITEM:FILLer:MODE DURation
SEquence:ITEM:FILLer:TIME FIXed
SEquence:ITEM:FILLer:TIME:FIXed 5e-3

```

Example: Creating a simple sequence with two pulses repeated different number of times

```

SCPI
SEquence:SElect "Test Sequence"
SEquence:TYPE?
SEquence:PHASe:MODE ABSolute

SEquence:ITEM:ADD
SEquence:ITEM:SElect 1; SEquence:ITEM:TYPE PULSe
SEquence:ITEM:PULSe "P1"
SEquence:ITEM:PRI 2.5 ms
SEquence:ITEM:PDELay 2 ms
SEquence:ITEM:REP:TYPE FIXed
SEquence:ITEM:REP:COUNT:FIXed 2

SEquence:ITEM:ADD
SEquence:ITEM:SElect 2; SEquence:ITEM:TYPE LOOP
SEquence:ITEM:LOOP:TYPE VARIable
SEquence:ITEM:LOOP:COUNT:MINimum 2
SEquence:ITEM:LOOP:COUNT:MAXimum 10
SEquence:ITEM:LOOP:COUNT:STEP 2

SEquence:ITEM:ADD
SEquence:ITEM:SElect 3; SEquence:ITEM:TYPE PULSe
SEquence:ITEM:PULSe "P1"
SEquence:ITEM:PRI 5 ms
SEquence:ITEM:FREquency:OFFSet 1000000
SEquence:ITEM:LEVel:OFFSet -10
SEquence:ITEM:REP:TYPE DURation
SEquence:ITEM:REP:COUNT:DURation 0.001
SEquence:ITEM:REP:COUNT:ROUNDing UP
SEquence:ITEM:INDent 1

```

Example: Creating an overlay

```

SCPI
SEquence:SElect "Test Sequence"
SEquence:ITEM COUNT?
// 3
SEquence:ITEM:ADD
SEquence:ITEM:SElect 4; SEquence:ITEM:TYPE PULSe
SEquence:ITEM:PULSe "P1"
SEquence:ITEM:PRI 2.5 ms

SEquence:ITEM:SElect 2
SEquence:ITEM:TYPE OVL
SEquence:ITEM:OVL:WTIME 0.001

SEquence:ITEM:SElect 3; SEquence:ITEM:INDent 1
SEquence:ITEM:SElect 4; SEquence:ITEM:INDent 1
    
```

SEquence:TYPE.....	448
SEquence:PHASe:MODE.....	448
SEquence:ITEM:FILLer:MODE.....	448
SEquence:ITEM:FILLer:SIGNal.....	448
SEquence:ITEM:FILLer:TIME.....	448
SEquence:ITEM:FILLer:TIME:EQUation.....	449
SEquence:ITEM:FILLer:TIME:FIXed.....	449
SEquence:ITEM:FREQuency:OFFSet.....	449
SEquence:ITEM:INDent.....	449
SEquence:ITEM:LEVel:OFFSet.....	450
SEquence:ITEM:LOOP:COUNT:FIXed.....	450
SEquence:ITEM:LOOP:COUNT:MINimum.....	450
SEquence:ITEM:LOOP:COUNT:MAXimum.....	450
SEquence:ITEM:LOOP:COUNT:STEP.....	450
SEquence:ITEM:LOOP:TYPE.....	451
SEquence:ITEM:LOOP:VARiable.....	451
SEquence:ITEM:OVL:VARiable.....	451
SEquence:ITEM:OVL:WTIME.....	451
SEquence:ITEM:PDELay.....	451
SEquence:ITEM:PHASe:OFFSet.....	452
SEquence:ITEM:PRI.....	452
SEquence:ITEM:WAVeform.....	452
SEquence:ITEM:PULSe.....	452
SEquence:ITEM:REP:COUNT:DURation.....	452
SEquence:ITEM:REP:COUNT:FIXed.....	453
SEquence:ITEM:REP:COUNT:MINimum.....	453
SEquence:ITEM:REP:COUNT:MAXimum.....	453
SEquence:ITEM:REP:COUNT:ROUNding.....	453
SEquence:ITEM:REP:COUNT:STEP.....	453
SEquence:ITEM:REP:TYPE.....	454
SEquence:ITEM:REP:VARiable.....	454
SEquence:ITEM:TYPE.....	454

SEquence:TYPE <Type>

Sets the sequence type.

Parameters:

<Type> PULSe | WAVeform

Example: see [Example "Handling items"](#) on page 325

Manual operation: See ["Sequence Name, Comment, Type, Sequence Description Table, Block Diagram"](#) on page 114

SEquence:PHASe:MODE <Mode>

Defines how the phase is set at each pulse start.

Parameters:

<Mode> ABSolute | CONTInuous | MEMory

Example: See [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Phase Mode"](#) on page 114

SEquence:ITEM:FILLer:MODE <Mode>

Sets how the filler duration is determined.

Parameters:

<Mode> DURation | TSYNc

Example: See [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Mode"](#) on page 122

SEquence:ITEM:FILLer:SIGNal <Signal>

Sets the signal type.

Parameters:

<Signal> BLANk | CW | HOLD

Example: see [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Signal"](#) on page 122

SEquence:ITEM:FILLer:TIME <Time>

Defines the way the duration is defined.

Parameters:

<Time> FIXed | EQUation

Example: See [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Time"](#) on page 122

SEquence:ITEM:FILLer:TIME:EQUation <Equation>

Sets the filler duration as an equation.

Parameters:

<Equation> string

Example: SEquence:ITEM:FILLer:TIME:EQUation?

Manual operation: See ["Time"](#) on page 122

SEquence:ITEM:FILLer:TIME:FIXed <Fixed>

Sets the duration of the filler.

Parameters:

<Fixed> float
Range: 0 to 3600
Default unit: sec

Example: see [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Time"](#) on page 122

SEquence:ITEM:FREQUency:OFFSet <Offset>

Enables a frequency offset.

Parameters:

<Offset> float
Range: -1e+09 to 1e+09
Default unit: Hz

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Δ Freq."](#) on page 117

SEquence:ITEM:INDent <Indent>

Indents the selected item rows to include it for example in a loop.

Parameters:

<Indent> float
Range: 0 to 5

Example: See [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Nesting"](#) on page 116

SEquence:ITEM:LEVel:OFFSet <Offset>

Sets a level offset.

Parameters:

<Offset> float
Range: -100 to 0
Default unit: dB

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See "[Δ Level](#)" on page 117

SEquence:ITEM:LOOP:COUNT:FIXed <Fixed>

Sets the repetition number as a numeric value.

Parameters:

<Fixed> float
Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See "[Rep. Cont.](#)" on page 117

SEquence:ITEM:LOOP:COUNT:MINimum <Minimum>**SEquence:ITEM:LOOP:COUNT:MAXimum** <Maximum>

Sets the value range of the loop count.

Parameters:

<Maximum> float
Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See "[Loop Repetition](#)" on page 120

SEquence:ITEM:LOOP:COUNT:STEP <Step>

Sets the loop count granularity.

Parameters:

<Step> float
Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See "[Loop Repetition](#)" on page 120

SEquence:ITEM:LOOP:TYPE <Type>

Sets how the loop repetition is defined.

Parameters:

<Type> FIXed | VARiable

Example: See [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Loop Repetition"](#) on page 120

SEquence:ITEM:LOOP:VARiable <Variable>

Sets a loop variable.

Parameters:

<Variable> string

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Loop Variables"](#) on page 120

SEquence:ITEM:OVL:VARiable <Variable>

Sets a variable.

Parameters:

<Variable> string

Manual operation: See ["Overlay Variables"](#) on page 121

SEquence:ITEM:OVL:WTIME <Wtime>

Sets the duration of the overlay.

Parameters:

<Wtime> float
Range: 0 to 3600
Default unit: sec

Example: see [Example "Creating an overlay"](#) on page 447

Manual operation: See ["Duration"](#) on page 121

SEquence:ITEM:PDELay <Pdelay>

Enables a start delay.

Parameters:

<Pdelay> float
Range: 0 to 1e+09
Default unit: sec

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Delay"](#) on page 118

SEquence:ITEM:PHASe:OFFSet <Offset>

Sets a phase offset.

Parameters:

<Offset> float
Range: -180 to 180

Example: SEquence:ITEM:PHASe:OFFSet 30

Manual operation: See ["Phase"](#) on page 117

SEquence:ITEM:PRI <Pri>

Sets the Pulse Repetition Interval (PRI).

Parameters:

<Pri> float
Range: 0 to 1e+09

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["PRI"](#) on page 118

SEquence:ITEM:WAVEform <Waveform>

SEquence:ITEM:PULSe <Pulse>

Assigns a pulse or a waveform to the selected item.

Use the commands [PULSe:CATalog?](#) and [WAVEform:CATalog?](#) to query the available pulses and waveforms.

Parameters:

<Pulse> string
Pulse name

Example: See [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Pulse/Waveform"](#) on page 116

SEquence:ITEM:REP:COUNT:DURation <Duration>

Sets a time duration.

Parameters:

<Duration> float
 Range: 0 to 1e+09
 Default unit: sec

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Repetition Number"](#) on page 119

SEquence:ITEM:REP:COUNT:FIXed <Fixed>

Sets the repetition number as a numeric value.

Parameters:

<Fixed> float
 Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Rep. Cont."](#) on page 117

SEquence:ITEM:REP:COUNT:MINimum <Minimum>**SEquence:ITEM:REP:COUNT:MAXimum** <Maximum>

Sets the value range of the repetition count.

Parameters:

<Maximum> float
 Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Repetition Number"](#) on page 119

SEquence:ITEM:REP:COUNT:ROUNding <Rounding>

Sets how the repetition number is rounded.

Parameters:

<Rounding> DN | UP

Example: See [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Repetition Number"](#) on page 119

SEquence:ITEM:REP:COUNT:STEP <Step>

Sets the repetition count granularity.

Parameters:

<Step> float
Range: 1 to 65535

Example: see [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Repetition Number"](#) on page 119

SEQUENCE:ITEM:REP:TYPE <Type>

Sets how the repetition number is defined.

Parameters:

<Type> FIXed | VARiable | DURation

Example: See [Example "Creating a simple sequence with two pulses repeated different number of times"](#) on page 446

Manual operation: See ["Repetition Number"](#) on page 119

SEQUENCE:ITEM:REP:VARiable <Variable>

Set a repetition variable.

Parameters:

<Variable> string

Manual operation: See ["Repetition Variables"](#) on page 119

SEQUENCE:ITEM:TYPE <Type>

Sets the content type of the selected item.

Parameters:

<Type> PULSe | FILLer | LOOP | OVL | SUBSequence | WAVeform

Example: see [Example "Creating a CW segment"](#) on page 446

Manual operation: See ["Type"](#) on page 116

23.20 Status Commands

This system contains the commands for the status reporting system.

Refer to the R&S SMW/SMBV/SGT user manual for detailed information.

Value ranges

- **Return parameters**

Queries return the current value of the respective register, which permits a check of the device status.

Range: A decimal value in the range 0 to 32767 ($=2^{15}-1$)

- **Setting parameters**

The configuration commands set the respective register thus determining which status changes of the R&S Pulse Sequencer cause the status registers to be changed.

Range: A decimal value in the range 0 to 32767 ($=2^{15}-1$)

STATus:OPERation:CONDition?

Queries 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 status.

Return values:

<Condition> float

Example: :STATus:OPERation:CONDition?
queries the Status:Operation:Condition register.

Usage: Query only

23.21 System Commands

The SYSTem subsystem contains a series of commands for general functions which do not directly affect signal generation.

SYSTem:ERRor?

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:

<Error> string
Error/event_number,
"Error/event_description>[;Device-dependent
info]"
If the queue is empty, the response is 0, "No error"

Example: see [SYSTem:ERRor:ALL?](#) on page 455

Usage: Query only

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:

SYSTem:ERRor: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

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:

SYSTem:ERRor:COUNT?
 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:PROGress?

Queries the signal generation progress status.

Return values:

<Progress> float

Example:

see [Example "Creating simple pulse train scenario"](#) on page 425

Usage:

Query only

Manual operation:

See ["Calculate + Deploy, Start/Stop, Busy/Idle"](#) on page 59

23.22 Waveform Commands

Example: Creating waveforms

```

SCPI
WAVeform:CREate "NOISe"
WAVeform:SElect "NOISe"
WAVeform:TYPE NOISe
WAVeform:NOISe:BWIDth 2e+6

WAVeform:CREate "WV"
WAVeform:SElect "WV"
WAVeform:TYPE WAVeform
WAVeform:SIGCont COMM
WAVeform:WAVeform:LOAD "C:\_PS_files\dl_IQ_10MBW.wv"
WAVeform:LEVel:REFerence?
// 14.4695
WAVeform:WAVeform:CLEar

WAVeform:CREate "BKG"
WAVeform:SElect "BKG"
WAVeform:TYPE BEMitter
WAVeform:BEMitter:COUNT 64
WAVeform:BEMitter:BWIDth 2e+07
WAVeform:BEMitter:DURation 0.05
WAVeform:BEMitter:PW:MIN 5e-05
WAVeform:BEMitter:PW:MAX 0.00025
WAVeform:BEMitter:PRI:RATio:MIN 10
WAVeform:BEMitter:PRI:RATio:MAX 20
WAVeform:BEMitter:LEVel:RANGe 20

```

WAVeform:TYPE.....	458
WAVeform:NOISe:BWIDth.....	458
WAVeform:SIGCont.....	458
WAVeform:WAVeform:LOAD.....	458
WAVeform:LEVel:REFerence.....	458
WAVeform:IQ:CLEar.....	459
WAVeform:WAVeform:CLEar.....	459
WAVeform:MT:COUNT.....	459
WAVeform:MT:SPACing.....	459
WAVeform:BEMitter:COUNT.....	459
WAVeform:BEMitter:BWIDth.....	460
WAVeform:BEMitter:DURation.....	460
WAVeform:BEMitter:PRI:RATio:MINimum.....	460
WAVeform:BEMitter:PRI:RATio:MAXimum.....	460
WAVeform:BEMitter:PW:MINimum.....	460
WAVeform:BEMitter:PW:MAXimum.....	460
WAVeform:BEMitter:LEVel:RANGe.....	461

WAVeform:TYPE <Type>

Sets the type of the waveform.

Parameters:

<Type> CW | NOISe | WAVeform | USER | BEMitter | MT

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Type"](#) on page 224

WAVeform:NOISe:BWIDth <Bwidth>

Sets the bandwidth of the generated AWGN waveform.

Parameters:

<Bwidth> float
Range: 0 to 3e+08
Default unit: Hz

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["White Noise settings"](#) on page 227

WAVeform:SIGCont <Sigcont>

Defines the waveform signal type.

Parameters:

<Sigcont> PULSe | COMM

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Settings for importing externally created R&S waveforms or files with custom I/Q data"](#) on page 227

WAVeform:WAVeform:LOAD <Load>

Load the selected waveform file (* .wv), see [Table A-1](#).

Setting parameters:

<Load> string
Complete file path with file name and extension

Example: See [Example "Creating waveforms"](#) on page 457

Usage: Setting only

Manual operation: See ["Settings for importing externally created R&S waveforms or files with custom I/Q data"](#) on page 227

WAVeform:LEVel:REFerence <Reference>

Queries the reference level.

Parameters:

<Reference> float
Range: 0 to 100

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Settings for importing externally created R&S waveforms or files with custom I/Q data"](#) on page 227
See ["Measure Ref. Level"](#) on page 269

WAVeform:IQ:CLEAr**WAVeform:WAVeform:CLEAr**

Removes the imported waveform or file with I/Q data.

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Settings for importing externally created R&S waveforms or files with custom I/Q data"](#) on page 227

WAVeform:MT:COUNT <Count>

Sets the number of tones.

Parameters:

<Count> integer
Range: 2 to 1024

Example:
`WAVeform:CREate "MT"`
`WAVeform:SElect "MT"`
`WAVeform:TYPE MT`
`WAVeform:MT:COUNT 64`
`WAVeform:MT:SPACing 100000`

Manual operation: See ["CW and Multitone waveform settings"](#) on page 224

WAVeform:MT:SPACing <Spacing>

Sets the tone spacing.

Parameters:

<Spacing> float
Range: 100 to 1e+07

Example: see [WAVeform:MT:COUNT](#) on page 459

Manual operation: See ["CW and Multitone waveform settings"](#) on page 224

WAVeform:BEMitter:COUNT <Count>

Sets the number of background emitters.

Parameters:

<Count> integer
Range: 1 to 255

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

WAVeform:BEMitter:BWIDth <Bwidth>

Sets the maximum permissible bandwidth of the resulting signal.

Parameters:

<Bwidth> float
Range: 1000 to 2.4e+08

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

WAVeform:BEMitter:DURation <Duration>

Sets the signal duration.

Parameters:

<Duration> float
Range: 0.0001 to 1

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

WAVeform:BEMitter:PRI:RATio:MINimum <Minimum>**WAVeform:BEMitter:PRI:RATio:MAXimum <Maximum>**

Sets the value range for the PRI/PW ratio.

Parameters:

<Maximum> float
Range: 10 to 1000

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

WAVeform:BEMitter:PW:MINimum <Minimum>**WAVeform:BEMitter:PW:MAXimum <Maximum>**

Sets the value range for the pulse width values.

Parameters:

<Maximum> float
Range: 5e-05 to 1

Example: see [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

WAVeform:BEMitter:LEVel:RANGe <Range>

Sets the maximum level difference between the strongest and the weakest emitter.

Parameters:

<Range> float
Range: 0 to 90

Example: See [Example "Creating waveforms"](#) on page 457

Manual operation: See ["Background Emitters settings"](#) on page 225

23.23 Waveform Viewer Commands

WAVeform:VIEW:XMODE	461
SCENario:VOLatile:VIEW:XMODE	461
WAVeform:VIEW:YMODE	461
SCENario:VOLatile:VIEW:YMODE	461
WAVeform:VIEW:ZOOM:POINT	462
SCENario:VOLatile:VIEW:ZOOM:POINT	462
WAVeform:VIEW:ZOOM:RANGe	462
SCENario:VOLatile:VIEW:ZOOM:RANGe	462

WAVeform:VIEW:XMODE <Xmode>

SCENario:VOLatile:VIEW:XMODE <Xmode>

Sets the units (time or samples) used on the x axis.

Setting parameters:

<Xmode> SAMPLES | TIME

Usage: Setting only

Manual operation: See ["Units"](#) on page 265

WAVeform:VIEW:YMODE <Ymode>

SCENario:VOLatile:VIEW:YMODE <Ymode>

Sets the view mode.

Setting parameters:

<Ymode> IQ | MAGDb | MAGW | MAGV | PHASe | FREQuency | PAV

Usage: Setting only

Manual operation: See ["View mode"](#) on page 265

WAVeform:VIEW:ZOOM:POINT <Point>

SCENario:VOLatile:VIEW:ZOOM:POINT <Point>

Sets center point of the displayed area.

Setting parameters:

<Point> float
 Always related to time
 Default unit: s

Example:

```
SCENario:VOLatile:VIEW:ZOOM:POINT 300 us
SCENario:VOLatile:VIEW:ZOOM:RANGe 100u
// 100 us around the 300 us point, i.e.
// displayed is the time span of 200 us to 400 us

SCENario:VOLatile:VIEW:ZOOM:POINT 0.5 ms
SCENario:VOLatile:VIEW:ZOOM:RANGe 2 m
// displayed is the time span of 0 s to 2 ms
```

Usage: Setting only

Manual operation: See "[Zoom in/out](#)" on page 266

WAVeform:VIEW:ZOOM:RANGe <Range>

SCENario:VOLatile:VIEW:ZOOM:RANGe <Range>

Sets the displayed waveform part as a range around the selected center point, set with the command `SCENario:VOLatile:VIEW:ZOOM:POINT`.

Setting parameters:

<Range> float
 Expressed as a time span (units can be omitted) or as number of samples

Example: See `SCENario:VOLatile:VIEW:ZOOM:POINT`

Usage: Setting only

Manual operation: See "[Zoom in/out](#)" on page 266

23.24 List of Remote Commands

*IDN?.....	323
*OPT?.....	323
*RST.....	323
ANTenna:CATalog?.....	329
ANTenna:COMMeNt.....	331
ANTenna:CREate.....	329
ANTenna:MODEl:ARRay:COsN.....	340
ANTenna:MODEl:ARRay:DISTRibution.....	340
ANTenna:MODEl:ARRay:ELEMent:COsine.....	340

List of Remote Commands

ANTenna:MODel:ARRay:NX.....	340
ANTenna:MODel:ARRay:NZ.....	340
ANTenna:MODel:ARRay:PEDestal.....	341
ANTenna:MODel:ARRay:XDIStance.....	341
ANTenna:MODel:ARRay:ZDIStance.....	341
ANTenna:MODel:BACKlobe:ATTenuation.....	342
ANTenna:MODel:BACKlobe:ENABle.....	341
ANTenna:MODel:BACKlobe:TYPE.....	342
ANTenna:MODel:BANDwidth.....	344
ANTenna:MODel:CARRay:CIRCular:DIStance.....	341
ANTenna:MODel:CARRay:CIRCular:LATTice.....	342
ANTenna:MODel:CARRay:CIRCular:RADIus.....	342
ANTenna:MODel:CARRay:COsN.....	340
ANTenna:MODel:CARRay:DIStRiBution.....	340
ANTenna:MODel:CARRay:ELEMeNt.....	343
ANTenna:MODel:CARRay:ELEMeNt:COsine.....	340
ANTenna:MODel:CARRay:GEOMetry.....	343
ANTenna:MODel:CARRay:HEXagonal:DIStance.....	341
ANTenna:MODel:CARRay:HEXagonal:N.....	340
ANTenna:MODel:CARRay:LINear:DIStance.....	341
ANTenna:MODel:CARRay:LINear:N.....	340
ANTenna:MODel:CARRay:PEDestal.....	341
ANTenna:MODel:CARRay:RECTangular:LATTice.....	342
ANTenna:MODel:CARRay:RECTangular:NX.....	340
ANTenna:MODel:CARRay:RECTangular:NZ.....	340
ANTenna:MODel:CARRay:RECTangular:XDIStance.....	341
ANTenna:MODel:CARRay:RECTangular:ZDIStance.....	341
ANTenna:MODel:COsEcant:HPBW.....	345
ANTenna:MODel:COsEcant:T1.....	342
ANTenna:MODel:COsEcant:T2.....	342
ANTenna:MODel:CUSTom:HPBW:XY.....	343
ANTenna:MODel:CUSTom:HPBW:YZ.....	343
ANTenna:MODel:CUSTom:SLRolloff.....	344
ANTenna:MODel:CUSTom:SLSCale.....	344
ANTenna:MODel:CUSTom:SLStart.....	344
ANTenna:MODel:FREQuency.....	344
ANTenna:MODel:GAUSSian:HPBW.....	345
ANTenna:MODel:HORN:LX.....	345
ANTenna:MODel:HORN:LZ.....	345
ANTenna:MODel:PARAbolic:DIAMeter.....	345
ANTenna:MODel:POLarization.....	339
ANTenna:MODel:ROTation:X.....	346
ANTenna:MODel:ROTation:Z.....	346
ANTenna:MODel:SINC:HPBW.....	345
ANTenna:MODel:TYPE.....	339
ANTenna:MODel:USER:CLEar.....	335
ANTenna:MODel:USER:CSV:FORMat.....	346
ANTenna:MODel:USER:LOAD.....	346
ANTenna:NAME.....	330
ANTenna:REMOve.....	331

ANTenna:SElect.....	330
ASSignment:ANTennas:LIST?.....	363
ASSignment:ANTennas:SElect.....	334
ASSignment:EMITters:LIST?.....	363
ASSignment:EMITters:SElect.....	334
ASSignment:GENerator:CAPabilities?.....	364
ASSignment:GENerator:LIST?.....	364
ASSignment:GENerator:MASTer.....	364
ASSignment:GENerator:PATH:ANTenna:ADD.....	332
ASSignment:GENerator:PATH:ANTenna:CLEar.....	335
ASSignment:GENerator:PATH:ANTenna:DELeTe.....	334
ASSignment:GENerator:PATH:ANTenna:LIST?.....	365
ASSignment:GENerator:PATH:ANTenna:SElect.....	334
ASSignment:GENerator:PATH:EMITter:ADD.....	332
ASSignment:GENerator:PATH:EMITter:CLEar.....	335
ASSignment:GENerator:PATH:EMITter:DELeTe.....	334
ASSignment:GENerator:PATH:EMITter:LIST?.....	365
ASSignment:GENerator:PATH:EMITter:SElect.....	334
ASSignment:GENerator:PATH:LIST?.....	365
ASSignment:GENerator:PATH:SElect.....	334
ASSignment:GENerator:SElect.....	334
DASHboard:MUTE.....	357
DASHboard:REFResh.....	357
DASHboard:SCENario:CURRent:NAME?.....	357
DASHboard:SCENario:CURRent:STATus?.....	357
DASHboard:SCENario:DEPLoyed:NAME?.....	357
DASHboard:SCENario:DEPLoyed:TIME?.....	358
DASHboard:UNUSed:ALL.....	358
DASHboard:UNUSed:FREQUency.....	359
DASHboard:UNUSed:LEVel.....	359
DASHboard:UNUSed:LIST?.....	358
DASHboard:UNUSed:PATH:LIST.....	358
DASHboard:UNUSed:PATH:SElect.....	334
DASHboard:UNUSed:SElect.....	334
DASHboard:UNUSed:STATe.....	359
DASHboard:USED:FREQUency.....	359
DASHboard:USED:LEVel.....	359
DASHboard:USED:LIST?.....	358
DASHboard:USED:PATH:LIST?.....	358
DASHboard:USED:PATH:SElect.....	334
DASHboard:USED:SElect.....	334
DASHboard:USED:STATe.....	359
DSRC:CATalog?.....	329
DSRC:CLEar.....	335
DSRC:COMMeNt.....	331
DSRC:CREate.....	329
DSRC:ITEM:ADD.....	332
DSRC:ITEM:BITS.....	360
DSRC:ITEM:DATA.....	360
DSRC:ITEM:DELeTe.....	334

DSRC:ITEM:PATtern.....	361
DSRC:ITEM:PRBS:INIT.....	361
DSRC:ITEM:PRBS:INIT:VALue.....	361
DSRC:ITEM:PRBS:TYPE.....	361
DSRC:ITEM:SElect.....	333
DSRC:ITEM:TYPE.....	362
DSRC:NAME.....	330
DSRC:REMove.....	331
DSRC:SElect.....	330
EMITter:CATalog?.....	329
EMITter:COMMeNt.....	331
EMITter:CREate.....	329
EMITter:EIRP.....	366
EMITter:FREQuency.....	367
EMITter:MODE:ADD.....	332
EMITter:MODE:ANTenna.....	367
EMITter:MODE:ANTenna:CLEar.....	335
EMITter:MODE:BEAM:ADD.....	332
EMITter:MODE:BEAM:CLEar.....	335
EMITter:MODE:BEAM:COUNT?.....	333
EMITter:MODE:BEAM:DELeTe.....	334
EMITter:MODE:BEAM:OFFSet:AZIMuth.....	367
EMITter:MODE:BEAM:OFFSet:ELEVation.....	367
EMITter:MODE:BEAM:OFFSet:FREQuency.....	367
EMITter:MODE:BEAM:SElect.....	333
EMITter:MODE:BEAM:SEQuence.....	368
EMITter:MODE:BEAM:STATe.....	368
EMITter:MODE:CLEar.....	335
EMITter:MODE:COUNT?.....	333
EMITter:MODE:DELeTe.....	334
EMITter:MODE:NAME.....	330
EMITter:MODE:SCAN.....	368
EMITter:MODE:SCAN:CLEar.....	335
EMITter:MODE:SElect.....	333
EMITter:NAME.....	330
EMITter:REMove.....	331
EMITter:SElect.....	330
GENerator:CAPabilities.....	370
GENerator:CATalog?.....	329
GENerator:COMMeNt.....	331
GENerator:CREate.....	329
GENerator:NAME.....	330
GENerator:REMove.....	331
GENerator:SElect.....	330
GENerator:TYPE.....	370
INSTrument:ADD.....	371
INSTrument:CAPabilities?.....	370
INSTrument:CLEar.....	335
INSTrument:COUNT?.....	333
INSTrument:CREate.....	371

INSTrument:DELeTe.....	334
INSTrument:FIND?.....	371
INSTrument:FIRMWare?.....	371
INSTrument:LINK:STATus?.....	372
INSTrument:LIST?.....	370
INSTrument:MAP.....	372
INSTrument:RESource?.....	372
INSTrument:SElect.....	334
IPM:CATalog?.....	329
IPM:COMMeNt.....	331
IPM:CREate.....	329
IPM:EQUation.....	377
IPM:LIST:CLEar.....	335
IPM:LIST:ITEM:ADD.....	332
IPM:LIST:ITEM:COUNT?.....	333
IPM:LIST:ITEM:DELeTe.....	334
IPM:LIST:ITEM:REPetition.....	377
IPM:LIST:ITEM:SElect.....	333
IPM:LIST:ITEM:VALue.....	377
IPM:LIST:LOAD.....	377
IPM:LIST:SAVE.....	377
IPM:NAME.....	330
IPM:PLUGin:NAME.....	330
IPM:PLUGin:VARiable:CATalog?.....	393
IPM:PLUGin:VARiable:SElect.....	393
IPM:PLUGin:VARiable:VALue.....	393
IPM:RANDom:DISTRibution.....	378
IPM:RANDom:NORMal:LIMit.....	378
IPM:RANDom:NORMal:MEAN.....	378
IPM:RANDom:NORMal:STD.....	378
IPM:RANDom:U:CENTer.....	379
IPM:RANDom:U:RANGe.....	379
IPM:RANDom:UNIForm:MAXimum.....	379
IPM:RANDom:UNIForm:MINimum.....	379
IPM:RANDom:UNIForm:STEP.....	379
IPM:REMOve.....	331
IPM:RLISt:BURSt.....	379
IPM:RLISt:REUSe.....	380
IPM:RSTep:MAXimum.....	382
IPM:RSTep:MINimum.....	382
IPM:RSTep:PERiod.....	382
IPM:RSTep:STEP:MAXimum.....	382
IPM:RSTep:STEP:MINimum.....	382
IPM:SElect.....	330
IPM:SHAPE:BASE.....	380
IPM:SHAPE:COUNT.....	380
IPM:SHAPE:INTerpol.....	380
IPM:SHAPE:PERiod.....	381
IPM:STEP:BURSt.....	381
IPM:STEP:INCRement.....	381

List of Remote Commands

IPM:STEP:START.....	381
IPM:STEP:STEPS.....	381
IPM:TYPE.....	375
IPM:UNIT.....	376
IPM:WAVEform:BASE.....	383
IPM:WAVEform:COUNT.....	383
IPM:WAVEform:OFFSet.....	383
IPM:WAVEform:PERiod.....	384
IPM:WAVEform:PHASe.....	383
IPM:WAVEform:PKPK.....	384
IPM:WAVEform:TYPE.....	382
PLOT:POLar:CUT.....	347
PLOT:POLar:LOG:MIN.....	347
PLOT:POLar:TYPE.....	347
PLUGin:CATalog?.....	329
PLUGin:COMMeNt.....	331
PLUGin:CREate.....	329
PLUGin:LOAD.....	392
PLUGin:MODule:AUTHor?.....	392
PLUGin:MODule:COMMeNt?.....	392
PLUGin:MODule:DATA?.....	392
PLUGin:MODule:NAME?.....	330
PLUGin:MODule:TYPE?.....	392
PLUGin:MODule:VERSion?.....	392
PLUGin:NAME.....	330
PLUGin:REMOve.....	331
PLUGin:SElect.....	330
PULSe:CATalog?.....	329
PULSe:COMMeNt.....	331
PULSe:CREate.....	329
PULSe:CUSTom.....	399
PULSe:ENVELOpe:DATA:CLEar.....	335
PULSe:ENVELOpe:DATA:ITEM:ADD.....	332
PULSe:ENVELOpe:DATA:ITEM:COUNT?.....	333
PULSe:ENVELOpe:DATA:ITEM:DELEte.....	334
PULSe:ENVELOpe:DATA:ITEM:SElect.....	333
PULSe:ENVELOpe:DATA:ITEM:VALue.....	399
PULSe:ENVELOpe:DATA:LOAD.....	399
PULSe:ENVELOpe:DATA:MULTiplier.....	400
PULSe:ENVELOpe:DATA:OFFSet.....	400
PULSe:ENVELOpe:DATA:SAVE.....	400
PULSe:ENVELOpe:DATA:UNIT.....	399
PULSe:ENVELOpe:EQUation.....	400
PULSe:ENVELOpe:MODE.....	401
PULSe:LEVel:DROop.....	401
PULSe:LEVel:OFF.....	401
PULSe:LEVel:ON.....	401
PULSe:MARKer:AUTO.....	387
PULSe:MARKer:FALL.....	387
PULSe:MARKer:GATE.....	387

List of Remote Commands

PULSe:MARKer:RISE.....	387
PULSe:MARKer:WIDTh.....	387
PULSe:MOP:AM:FREQuency.....	401
PULSe:MOP:AM:MDEPth.....	402
PULSe:MOP:AM:TYPE.....	402
PULSe:MOP:AMSTep:ADD.....	332
PULSe:MOP:AMSTep:CLEar.....	335
PULSe:MOP:AMSTep:COUnT?.....	333
PULSe:MOP:AMSTep:DELeTe.....	334
PULSe:MOP:AMSTep:DURation.....	402
PULSe:MOP:AMSTep:INSert.....	334
PULSe:MOP:AMSTep:LEVel.....	402
PULSe:MOP:AMSTep:SELeCt.....	333
PULSe:MOP:ASK:INVert.....	403
PULSe:MOP:ASK:MDEPth.....	403
PULSe:MOP:ASK:SRATe.....	403
PULSe:MOP:BARKeR:BLANK.....	404
PULSe:MOP:BARKeR:CODe.....	404
PULSe:MOP:BARKeR:TTIME.....	404
PULSe:MOP:BPSK:PHASe.....	404
PULSe:MOP:BPSK:SRATe.....	405
PULSe:MOP:BPSK:SRATe:AUTO.....	405
PULSe:MOP:BPSK:TTIME.....	405
PULSe:MOP:BPSK:TTYPE.....	405
PULSe:MOP:BPSK:TYPE.....	405
PULSe:MOP:CHIRp:DEVIation.....	406
PULSe:MOP:CHIRp:TYPE.....	406
PULSe:MOP:COMMeNt.....	331
PULSe:MOP:DATA:CODeIng.....	406
PULSe:MOP:DATA:DSRC.....	406
PULSe:MOP:ENABle.....	406
PULSe:MOP:EXCLUde:ENABle.....	407
PULSe:MOP:EXCLUde:LEVel:STARt.....	407
PULSe:MOP:EXCLUde:LEVel:STOP.....	407
PULSe:MOP:EXCLUde:MODe.....	407
PULSe:MOP:EXCLUde:TIME:STARt.....	407
PULSe:MOP:EXCLUde:TIME:STOP.....	407
PULSe:MOP:FILeTer:BT.....	408
PULSe:MOP:FILeTer:BWIDth.....	408
PULSe:MOP:FILeTer:LENGth.....	408
PULSe:MOP:FILeTer:ROLLoff.....	408
PULSe:MOP:FILeTer:TYPE.....	409
PULSe:MOP:FM:DEVIation.....	409
PULSe:MOP:FM:FREQuency.....	409
PULSe:MOP:FMSTep:ADD.....	332
PULSe:MOP:FMSTep:CLEar.....	335
PULSe:MOP:FMSTep:COUnT?.....	333
PULSe:MOP:FMSTep:DELeTe.....	335
PULSe:MOP:FMSTep:DURation.....	402
PULSe:MOP:FMSTep:FREQuency.....	403

PULSe:MOP:FMSTep:INSert.....	334
PULSe:MOP:FMSTep:SElect.....	333
PULSe:MOP:FSK:DEVIation.....	409
PULSe:MOP:FSK:SRATe.....	409
PULSe:MOP:NLCHirp:EQUation.....	410
PULSe:MOP:NOISe:BWIDth.....	410
PULSe:MOP:PCHirp:ADD.....	332
PULSe:MOP:PCHirp:CLear.....	335
PULSe:MOP:PCHirp:COEFFicient.....	410
PULSe:MOP:PCHirp:COUNt?.....	333
PULSe:MOP:PCHirp:DELeTe.....	335
PULSe:MOP:PCHirp:INSert.....	334
PULSe:MOP:PCHirp:SElect.....	333
PULSe:MOP:PCHirp:TERM.....	410
PULSe:MOP:PLISt:ADD.....	332
PULSe:MOP:PLISt:CLear.....	335
PULSe:MOP:PLISt:COUNt?.....	333
PULSe:MOP:PLISt:DELeTe.....	335
PULSe:MOP:PLISt:INSert.....	334
PULSe:MOP:PLISt:SElect.....	333
PULSe:MOP:PLISt:VALue.....	410
PULSe:MOP:PLUGin:NAME.....	330
PULSe:MOP:PLUGin:VARiable:CATalog?.....	393
PULSe:MOP:PLUGin:VARiable:SElect.....	393
PULSe:MOP:PLUGin:VARiable:VALue.....	393
PULSe:MOP:POLY:LENGth.....	411
PULSe:MOP:POLY:TYPE.....	411
PULSe:MOP:QPSK:SOQPsk:IRIG.....	411
PULSe:MOP:QPSK:SRATe.....	411
PULSe:MOP:QPSK:TYPE.....	411
PULSe:MOP:TYPE.....	412
PULSe:NAME.....	330
PULSe:OVERshoot.....	412
PULSe:OVERshoot:DECay.....	412
PULSe:PREView:MODE.....	414
PULSe:PREView:MOP.....	414
PULSe:REMOve.....	331
PULSe:RIPPlE.....	412
PULSe:RIPPlE:FREQUency.....	413
PULSe:SElect.....	330
PULSe:SETTings.....	414
PULSe:TIME:FALL.....	413
PULSe:TIME:REFerence.....	413
PULSe:TIME:RISE.....	413
PULSe:TIME:WIDTh.....	413
PULSe:TYPE:FALL.....	414
PULSe:TYPE:RISE.....	414
RECEiver:ANTenna:ADD.....	332
RECEiver:ANTenna:ALlas.....	416
RECEiver:ANTenna:CLear.....	335

RECeiver:ANTenna:DELeTe.....	334
RECeiver:ANTenna:DIRection:AWAY.....	417
RECeiver:ANTenna:DIRection:AZIMuth.....	417
RECeiver:ANTenna:DIRection:ELEVation.....	418
RECeiver:ANTenna:GAIN.....	417
RECeiver:ANTenna:PATTeRn.....	417
RECeiver:ANTenna:POSiTion:ANGLe.....	416
RECeiver:ANTenna:POSiTion:HEIGHt.....	417
RECeiver:ANTenna:POSiTion:RADIus.....	416
RECeiver:ANTenna:POSiTion:X.....	416
RECeiver:ANTenna:POSiTion:Y.....	416
RECeiver:ANTenna:SELeCt.....	333
RECeiver:CATalog?.....	329
RECeiver:COMMeNt.....	331
RECeiver:CREate.....	329
RECeiver:NAME.....	330
RECeiver:REMOve.....	331
RECeiver:SELeCt.....	330
REPManager:CATalog?.....	422
REPManager:DELeTe.....	423
REPManager:EXPort.....	423
REPManager:LOAD.....	422
REPManager:PATH:ADD.....	424
REPManager:PATH:DELeTe.....	424
REPManager:PATH:LIST?.....	424
REPository:ACCess?.....	419
REPository:AUTHor.....	420
REPository:CATalog?.....	329
REPository:COMMeNt.....	331
REPository:COMPLexity.....	420
REPository:CREate.....	329
REPository:DATE.....	420
REPository:FILEname?.....	420
REPository:PATH?.....	421
REPository:REMOve.....	332
REPository:SAVE.....	421
REPository:SECurity.....	421
REPository:SELeCt.....	330
REPository:UUId?.....	422
REPository:VERSiOn.....	421
SCAN:CATalog?.....	329
SCAN:CIRCUlar:NELevation.....	354
SCAN:CIRCUlar:NODDing.....	354
SCAN:CIRCUlar:NRATE.....	354
SCAN:CIRCUlar:PALMer.....	355
SCAN:CIRCUlar:PRATE.....	355
SCAN:CIRCUlar:PSQuint.....	355
SCAN:CIRCUlar:ROtation.....	349
SCAN:CIRCUlar:RPM.....	349
SCAN:COMMeNt.....	331

SCAN:CONical:RATE.....	350
SCAN:CONical:ROTation.....	349
SCAN:CONical:SQUint.....	351
SCAN:CREate.....	329
SCAN:HELical:ELEVation:STEP.....	350
SCAN:HELical:RETRace.....	352
SCAN:HELical:ROTation.....	349
SCAN:HELical:RPM.....	349
SCAN:HELical:TURNs.....	350
SCAN:LSW:DIRection.....	350
SCAN:LSW:DWELL.....	351
SCAN:LSW:LOBes.....	351
SCAN:LSW:ROTation.....	349
SCAN:LSW:SQUint.....	351
SCAN:NAME.....	331
SCAN:RASTer:BARs.....	351
SCAN:RASTer:BARWidth.....	351
SCAN:RASTer:DIRection.....	353
SCAN:RASTer:FLYBack.....	352
SCAN:RASTer:PALMer.....	355
SCAN:RASTer:PRATe.....	355
SCAN:RASTer:PSQUint.....	355
SCAN:RASTer:RATE.....	350
SCAN:RASTer:RETRace.....	352
SCAN:RASTer:UNIDirection.....	352
SCAN:RASTer:WIDTh.....	353
SCAN:REMOve.....	331
SCAN:SECTor:FLYBack.....	352
SCAN:SECTor:NELevation.....	354
SCAN:SECTor:NODDing.....	354
SCAN:SECTor:NRATe.....	354
SCAN:SECTor:PALMer.....	355
SCAN:SECTor:PRATe.....	355
SCAN:SECTor:PSQUint.....	355
SCAN:SECTor:RATE.....	350
SCAN:SECTor:UNIDirection.....	352
SCAN:SECTor:WIDTh.....	353
SCAN:SElect.....	330
SCAN:SPIRal:RETRace.....	352
SCAN:SPIRal:ROTation.....	349
SCAN:SPIRal:ROUNds.....	353
SCAN:SPIRal:RTIME.....	353
SCAN:SPIRal:STEP.....	353
SCAN:SPIRal:UNIDirection.....	352
SCAN:TYPE.....	354
SCENario:CACHe:REPository:CLEar.....	432
SCENario:CACHe:REPository:ENABle.....	431
SCENario:CACHe:REPository:VALid?.....	432
SCENario:CACHe:VOLatile:CLEar.....	432
SCENario:CACHe:VOLatile:RELease.....	432

List of Remote Commands

SCENario:CAChE:VOLatile:REStore.....	432
SCENario:CAChE:VOLatile:VALid?.....	432
SCENario:CALCulate.....	431
SCENario:CATalog?.....	329
SCENario:CEMit:ADD.....	332
SCENario:CEMit:ALias.....	433
SCENario:CEMit:CLEar.....	335
SCENario:CEMit:CURRent.....	434
SCENario:CEMit:DELeTe.....	335
SCENario:CEMit:DIRection:PItCh.....	434
SCENario:CEMit:DIRection:ROll.....	435
SCENario:CEMit:DIRection:YAW.....	434
SCENario:CEMit:EMITter.....	435
SCENario:CEMit:EMITter:MODE.....	435
SCENario:CEMit:EMITter:MODE:BEAM.....	436
SCENario:CEMit:SELEct.....	333
SCENario:COMMent.....	331
SCENario:CREate.....	329
SCENario:CSEquence.....	441
SCENario:CSEquence:ADD.....	332
SCENario:CSEquence:ALias.....	433
SCENario:CSEquence:CLEar.....	335
SCENario:CSEquence:CURRent.....	434
SCENario:CSEquence:DELeTe.....	335
SCENario:CSEquence:SELEct.....	333
SCENario:CSEquence:VARiable.....	441
SCENario:DIRection:ADD.....	332
SCENario:DIRection:ALias.....	433
SCENario:DIRection:CLEar.....	335
SCENario:DIRection:CURRent.....	434
SCENario:DIRection:DELeTe.....	335
SCENario:DIRection:DIRection:PItCh.....	434
SCENario:DIRection:DIRection:ROll.....	435
SCENario:DIRection:DIRection:TRACk.....	436
SCENario:DIRection:DIRection:YAW.....	434
SCENario:DIRection:DIStance?.....	436
SCENario:DIRection:EMITter.....	435
SCENario:DIRection:EMITter:MODE.....	435
SCENario:DIRection:EMITter:MODE:BEAM.....	436
SCENario:DIRection:LOCation:AZIMuth.....	437
SCENario:DIRection:LOCation:EASt.....	437
SCENario:DIRection:LOCation:ELEVation.....	438
SCENario:DIRection:LOCation:HEIGHt.....	438
SCENario:DIRection:LOCation:NORTh.....	437
SCENario:DIRection:RECEiver.....	440
SCENario:DIRection:RECEiver:DIRection:PItCh.....	434
SCENario:DIRection:RECEiver:DIRection:ROll.....	435
SCENario:DIRection:RECEiver:DIRection:YAW.....	434
SCENario:DIRection:RECEiver:HEIGHt.....	438
SCENario:DIRection:SELEct.....	333

List of Remote Commands

SCENario:DIRection:SEquence.....	439
SCENario:DIRection:SYNChronize:ENABLE.....	440
SCENario:DIRection:TYPE.....	439
SCENario:DIRection:WAVEform.....	439
SCENario:DIRection:WAVEform:ANTenna.....	438
SCENario:DIRection:WAVEform:EIRP.....	440
SCENario:DIRection:WAVEform:FREQuency.....	440
SCENario:DIRection:WAVEform:SCAN.....	438
SCENario:EMITter.....	435
SCENario:EMITter:CLEar.....	335
SCENario:EMITter:DIRection:PITCh.....	434
SCENario:EMITter:DIRection:ROLL.....	435
SCENario:EMITter:DIRection:YAW.....	434
SCENario:EMITter:MODE.....	435
SCENario:EMITter:MODE:BEAM.....	436
SCENario:GENerator.....	433
SCENario:GENerator:CLEar.....	335
SCENario:GENerator:PATH.....	433
SCENario:ID?.....	431
SCENario:LOCalized:ADD.....	332
SCENario:LOCalized:ALias.....	433
SCENario:LOCalized:CLEar.....	335
SCENario:LOCalized:CURRent.....	434
SCENario:LOCalized:DELeTe.....	335
SCENario:LOCalized:DIRection:PITCh.....	435
SCENario:LOCalized:DIRection:ROLL.....	435
SCENario:LOCalized:DIRection:TRACk.....	436
SCENario:LOCalized:DIRection:YAW.....	434
SCENario:LOCalized:DIStance?.....	436
SCENario:LOCalized:EMITter.....	435
SCENario:LOCalized:EMITter:MODE.....	435
SCENario:LOCalized:EMITter:MODE:BEAM.....	436
SCENario:LOCalized:LOCation:AZIMuth.....	437
SCENario:LOCalized:LOCation:EAST.....	437
SCENario:LOCalized:LOCation:ELEVation.....	438
SCENario:LOCalized:LOCation:HEIGht.....	438
SCENario:LOCalized:LOCation:NORTH.....	437
SCENario:LOCalized:RECeiver:ANTenna.....	438
SCENario:LOCalized:RECeiver:DIRection:PITCh.....	435
SCENario:LOCalized:RECeiver:DIRection:ROLL.....	435
SCENario:LOCalized:RECeiver:DIRection:YAW.....	434
SCENario:LOCalized:RECeiver:GAIN.....	439
SCENario:LOCalized:RECeiver:HEIGht.....	438
SCENario:LOCalized:RECeiver:SCAN.....	438
SCENario:LOCalized:SElect.....	333
SCENario:LOCalized:SEQuence.....	439
SCENario:LOCalized:SYNChronize:ENABLE.....	440
SCENario:LOCalized:TYPE.....	439
SCENario:LOCalized:WAVEform.....	439
SCENario:LOCalized:WAVEform:ANTenna.....	438

SCENario:LOCalized:WAVeform:EIRP.....	440
SCENario:LOCalized:WAVeform:FREQuency.....	440
SCENario:LOCalized:WAVeform:SCAN.....	438
SCENario:NAME.....	331
SCENario:OUTPut:CLOCK:AUTO:BORDER.....	442
SCENario:OUTPut:CLOCK:AUTO:OVERsampling.....	441
SCENario:OUTPut:CLOCK:MODE.....	441
SCENario:OUTPut:CLOCK:USER.....	442
SCENario:OUTPut:DURation:MODE.....	442
SCENario:OUTPut:DURation:TIME.....	442
SCENario:OUTPut:FORMat.....	442
SCENario:OUTPut:FREQuency.....	443
SCENario:OUTPut:INSTrument:QHIG:ENABle.....	443
SCENario:OUTPut:LEVel.....	443
SCENario:OUTPut:MARKer:ENABle.....	389
SCENario:OUTPut:MARKer:SCENario:DURation.....	389
SCENario:OUTPut:MARKer:SCENario:ENABle.....	389
SCENario:OUTPut:PATH.....	443
SCENario:OUTPut:RECall:ENABle.....	443
SCENario:OUTPut:RESet:ENABle.....	444
SCENario:OUTPut:RF:ENABle.....	444
SCENario:OUTPut:RUNMode.....	444
SCENario:OUTPut:TARGet.....	444
SCENario:OUTPut:THReshold.....	445
SCENario:PDW:ENABle.....	393
SCENario:PDW:PATH.....	393
SCENario:PDW:PLUGin:NAME.....	394
SCENario:PDW:PLUGin:VARiable:CATalog?.....	393
SCENario:PDW:PLUGin:VARiable:SElect.....	393
SCENario:PDW:PLUGin:VARiable:VALue.....	393
SCENario:PDW:TEMPlate.....	394
SCENario:PDW:TYPE.....	394
SCENario:REMove.....	331
SCENario:SANitize.....	445
SCENario:SElect.....	330
SCENario:SEQuence.....	433
SCENario:SEQuence:CLEar.....	433
SCENario:STARt.....	431
SCENario:STATe?.....	431
SCENario:STOP.....	431
SCENario:TYPE.....	430
SCENario:VOLatile:VIEW.....	445
SCENario:VOLatile:VIEW:XMODE.....	461
SCENario:VOLatile:VIEW:YMODE.....	461
SCENario:VOLatile:VIEW:ZOOM:POINt.....	462
SCENario:VOLatile:VIEW:ZOOM:RANGE.....	462
SEQuence:CATalog?.....	329
SEQuence:COMMENT.....	331
SEQuence:CREate.....	329
SEQuence:ITEM:ADD.....	332

SEquence:ITEM:CLEar.....	335
SEquence:ITEM:COUNT?.....	333
SEquence:ITEM:DELeTe.....	335
SEquence:ITEM:FILLer:MODE.....	448
SEquence:ITEM:FILLer:SIGNAL.....	448
SEquence:ITEM:FILLer:TIME.....	448
SEquence:ITEM:FILLer:TIME:EQUation.....	449
SEquence:ITEM:FILLer:TIME:FIXed.....	449
SEquence:ITEM:FREQuency:OFFSet.....	449
SEquence:ITEM:INDent.....	449
SEquence:ITEM:IPM:ADD.....	332
SEquence:ITEM:IPM:COUNT?.....	333
SEquence:ITEM:IPM:DELeTe.....	335
SEquence:ITEM:IPM:EQUation.....	384
SEquence:ITEM:IPM:MODE.....	384
SEquence:ITEM:IPM:RANDom:RESet.....	384
SEquence:ITEM:IPM:REStArt.....	385
SEquence:ITEM:IPM:SELeCt.....	333
SEquence:ITEM:IPM:SOURce.....	385
SEquence:ITEM:IPM:TARGet:PARAmeter.....	376
SEquence:ITEM:IPM:TARGet:TYPE.....	376
SEquence:ITEM:IPM:TARGet:VARiAbLe.....	376
SEquence:ITEM:LEVel:OFFSet.....	450
SEquence:ITEM:LOOP:COUNT:FIXed.....	450
SEquence:ITEM:LOOP:COUNT:MAXimum.....	450
SEquence:ITEM:LOOP:COUNT:MINimum.....	450
SEquence:ITEM:LOOP:COUNT:STEP.....	450
SEquence:ITEM:LOOP:TYPE.....	451
SEquence:ITEM:LOOP:VARiAbLe.....	451
SEquence:ITEM:MARKer:ALL.....	387
SEquence:ITEM:MARKer:CONDition.....	387
SEquence:ITEM:MARKer:CONDition.....	388
SEquence:ITEM:MARKer:CONDition:TYPE.....	388
SEquence:ITEM:MARKer:CONDition:VALue.....	388
SEquence:ITEM:MARKer:CONDition:VARiAbLe.....	388
SEquence:ITEM:MARKer:FIRSt.....	387
SEquence:ITEM:MARKer:LAST.....	387
SEquence:ITEM:OVL:VARiAbLe.....	451
SEquence:ITEM:OVL:WTIME.....	451
SEquence:ITEM:PDELay.....	451
SEquence:ITEM:PHASe:OFFSet.....	452
SEquence:ITEM:PRI.....	452
SEquence:ITEM:PULSe.....	452
SEquence:ITEM:REP:COUNT:DURation.....	452
SEquence:ITEM:REP:COUNT:FIXed.....	453
SEquence:ITEM:REP:COUNT:MAXimum.....	453
SEquence:ITEM:REP:COUNT:MINimum.....	453
SEquence:ITEM:REP:COUNT:ROUNding.....	453
SEquence:ITEM:REP:COUNT:STEP.....	453
SEquence:ITEM:REP:TYPE.....	454

SEquence:ITEM:REP:VARiable.....	454
SEquence:ITEM:SElect.....	333
SEquence:ITEM:TYPE.....	454
SEquence:ITEM:WAVeform.....	452
SEquence:NAME.....	331
SEquence:PHASe:MODE.....	448
SEquence:REMove.....	331
SEquence:SElect.....	330
SEquence:TYPE.....	448
STATus:OPERation:CONDition?.....	455
SYSTem:ERRor:ALL?.....	455
SYSTem:ERRor:COUNT?.....	456
SYSTem:ERRor?.....	455
SYSTem:PROGress?.....	456
WAVeform:BEMitter:BWIDth.....	460
WAVeform:BEMitter:COUNT.....	459
WAVeform:BEMitter:DURation.....	460
WAVeform:BEMitter:LEVel:RANGe.....	461
WAVeform:BEMitter:PRI:RATio:MAXimum.....	460
WAVeform:BEMitter:PRI:RATio:MINimum.....	460
WAVeform:BEMitter:PW:MAXimum.....	460
WAVeform:BEMitter:PW:MINimum.....	460
WAVeform:CATalog?.....	329
WAVeform:COMMeNt.....	331
WAVeform:CREate.....	329
WAVeform:IQ:CLEar.....	459
WAVeform:LEVel:REFerence.....	458
WAVeform:MT:COUNT.....	459
WAVeform:MT:SPACing.....	459
WAVeform:NAME.....	331
WAVeform:NOISe:BWIDth.....	458
WAVeform:REMove.....	332
WAVeform:SElect.....	330
WAVeform:SIGCont.....	458
WAVeform:TYPE.....	458
WAVeform:VIEW:XMODE.....	461
WAVeform:VIEW:YMODE.....	461
WAVeform:VIEW:ZOOM:POINt.....	462
WAVeform:VIEW:ZOOM:RANGe.....	462
WAVeform:WAVeform:CLEar.....	459
WAVeform:WAVeform:LOAD.....	458

24 Querying Error Messages and Troubleshooting

The R&S Pulse Sequencer monitors the functions performed and automatically detects errors and irregularities. The software logs status and error messages, warnings, SCPI commands sent to the software, or additional information in a log file. Some of the messages are also entered in the error/event queue of the status reporting system.

The "Message Log" dialog lists information about all messages in a history list. If the software detects an error, the "Message Log" dialog pops up automatically and displays the error message.

The log information may help you determine the cause for an error or an unexpected program behavior.

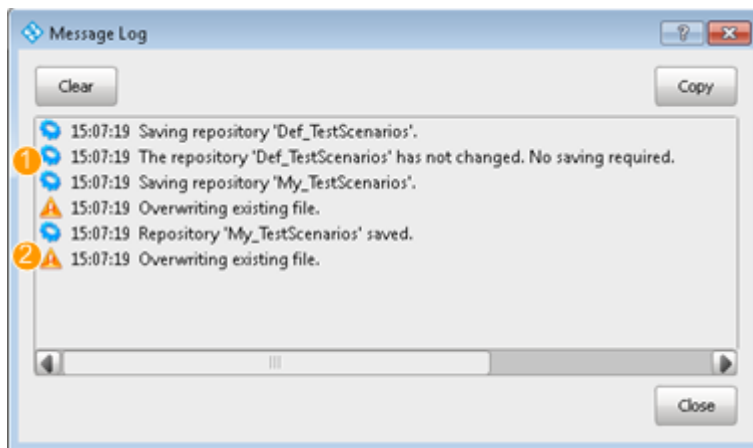
Additionally to the log information, the software provides a dialog with debug information, like the version of the used VISA library, the list of all loaded modules, or the report and data paths.

See:

- ["To access the Message Log dialog"](#) on page 477
- ["To open the Message Log dialog in debug mode"](#) on page 478
- ["To access the Debug Information dialog"](#) on page 478
- ["To remove write lock that results from a previously crashed session"](#) on page 479

To access the Message Log dialog

- ▶ Perform one of the following:
 - a) In the tool bar, select the "Log" icon.
 - b) In the menu bar, select "Window > Message Log".

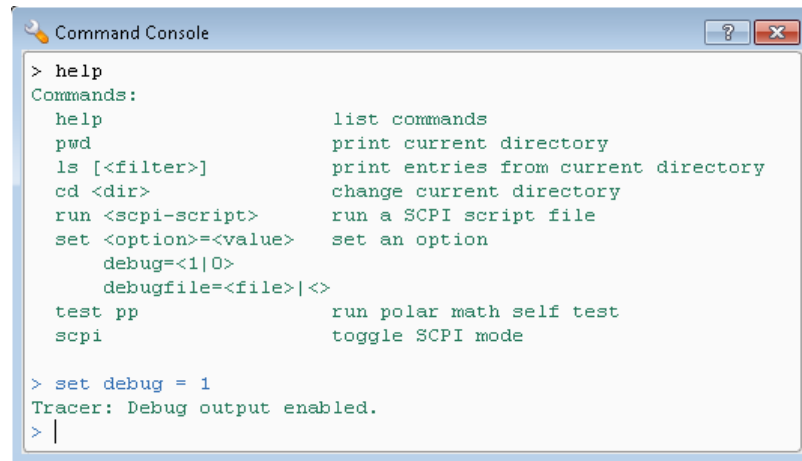


1 = Info message
2 = Warning

The displayed information is read-only but you can mark and copy it.

To open the Message Log dialog in debug mode

1. In the tool bar, select the "Console Window" icon.
2. In the "Command Console" window, type *help*.
The dialog lists all available commands.
3. Enter *set debug = 1* to enable the debug messages to be output in the "Message Log" view.

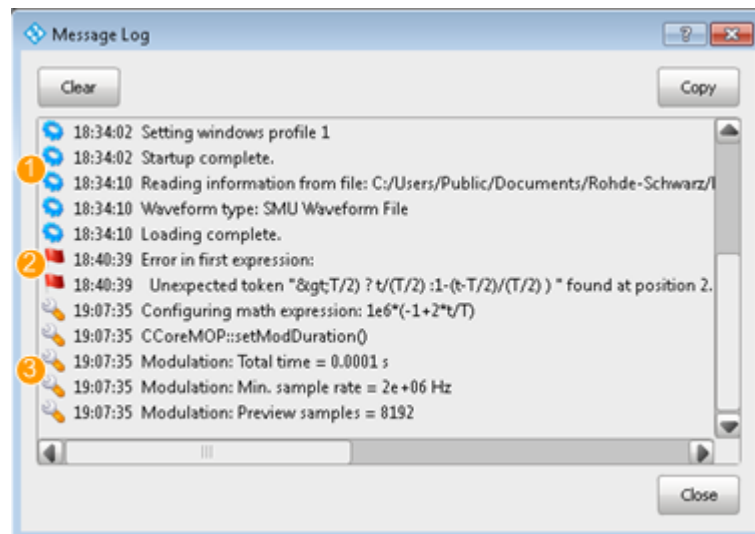


```

Command Console
> help
Commands:
  help          list commands
  pwd           print current directory
  ls [<filter>] print entries from current directory
  cd <dir>      change current directory
  run <scpi-script> run a SCPI script file
  set <option>=<value> set an option
                debug=<1|0>
                debugfile=<file>|<>
  test pp      run polar math self test
  scpi        toggle SCPI mode

> set debug = 1
Tracer: Debug output enabled.
> |

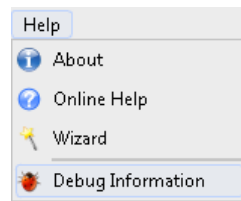
```



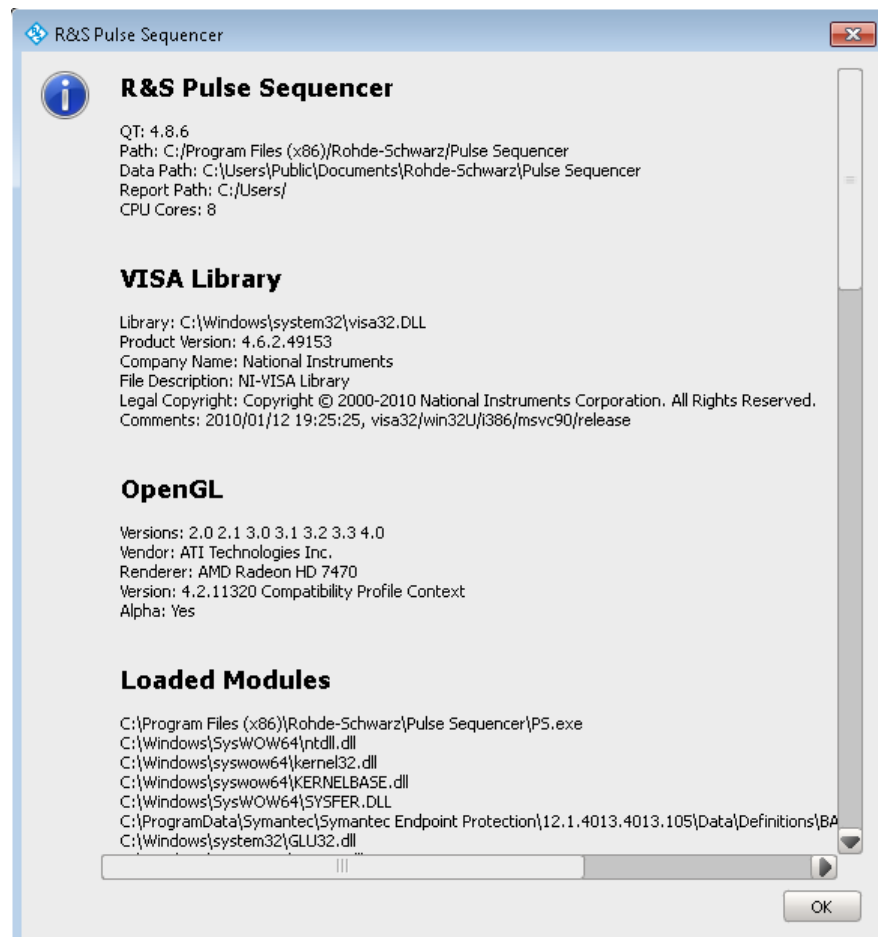
- 1 = Info message
- 2 = Error message
- 3 = Debug message

To access the Debug Information dialog


- ▶ In the menu bar, select "Help > Debug Information".



A dialog opens and displays information that may help the debugging.



To remove write lock that results from a previously crashed session

A red lock symbol  on the data base icon in the "Repository Tree" indicates that a session was terminated improperly. A locked repository can be unlocked only from the same PC and by the same user with write permission that had opened the repository before.

1. In the project tree, double click on the repository name.
2. In the "Repository" dialog, select "Storage > Remove Write Lock".

The repository is unlocked; you can change or store repository settings. See also [Chapter 5.1, "Repository Settings"](#), on page 43.

Annex

A Supported File Types and File Formats

This section summarizes the file types supported by the application and provides information about the default file locations.

List of supported file types and file extensions

Table A-1: Supported file types

Extension	Description
*.ant_pat *.ffe *.tsv *.ffd *.csv	Antenna pattern See Chapter A.2, "Antenna Pattern File Formats" , on page 485.
*.dll	Plugin, for example with user modulation, IPM profile
*.ini	Initialization file. Contains information on default location for temporary files
*.wv	Waveform file in a Rohde & Schwarz proprietary file format. Refer to the user manual of the signal generator for description of this file format
*.riq	Rohde & Schwarz proprietary I/Q data file format used by the R&S®PR100 portable receiver.
*.wav	Rohde & Schwarz proprietary audio-like file format that contains I/Q samples instead of the usual FM stereo signal. Although the file extension suggests that this file is a standard auto file, stereo players do not recognize it.
*.iq.tar	Rohde & Schwarz proprietary I/Q data file format used by some signal and spectrum analyzers, e.g. R&S®FSW. The *.iq.tar file contains I/Q data in binary format together with meta information that describes the nature and the source of data, e.g. the sample rate.
*.mat	MATLAB file that contains complex 1xN or Mx1 vectors.
*.txt	Standard ASCII files with comma-separated values. The *.txt files can be used to: <ul style="list-style-type: none"> Describe settings that are based on lists, like for example the custom pulse envelope shapes, the list for IPM profiles, or data lists. Describe custom I/Q data, with I and Q vectors in one file or in separated *_i.txt and *_q.txt files; If file pairs are stored in the same directory, they are recognized automatically by the "Import Wizard"
*.bin	Files with custom I/Q data in binary format

A.1 File Format of the Reporting Template

The template uses an ASCII text. Text that is not a token, a format (.FMT), or an option (.OPT) description is directly transferred to the final report.

Example: Example of a template that generates report in ASCII format

```
# R&S Pulse Sequencer, PDW Report      ← Text
#
Date:          <DATE>
Repository:    <REPOSITORY>           ← <Token>
Scenario:      <SCENARIO>

TOA           RF           PW           PA           MOP
              GHz          us           dBm

-----
.HDR
<TOA> ; <RF> ; <PW> ; <PA> ; <MOP>     ← .HDR
                                         <Col#1> ; <Col#2> ; ...
# END OF REPORT
.FMT TOA:12.0f:-9
.FMT RF:9.6f:9                          ← .FMT Token:Format:Exponent
.FMT PW:7.3f:-6
.FMT PA:4.0f:0
```

Figure A-1: PDW template explanation

General template text and available tokens

The general template text is copied to the final report. All tokens are replaced with values or text provided by the software.

See [Table A-2](#) for an overview of the available tokens.

Table A-2: List of available tokens

Token name	Description
<USER>	Current user name
<TPATH>	Target file path the generated report file is stored in
<ISODATE>	Date & time in ISO format
<DATE>	Date & time in local format
<REPOSITORY>	Repository name
<RCOMM>	Repository comment
<VERSION>	Repository version
<AUTHOR>	Repository author
<RCDATE>	Repository creation data
<RPATH>	Root path of repository
<SCENARIO>	Scenario name
<SCOMM>	Scenario comment

Adding and formatting data with the .HDR keyword

Each report contains one or more data lines. The position of these data lines in the report is indicated with the keyword `.HDR`. Usually, the reported data is formatted in table form, where the column names are defined with the information right after the `.HDR` line.

Each report template must have a `.HDR` line and subsequent second line defining the individual columns information and following the syntax:

```
.HDR
<Col#1>;<Col#2>;...
```

See [Table A-3](#) for an overview of the available tokens.

Table A-3: List of available .HDR tokens

Token name	Description
Values related to the generated pulse data	
<TOA>	Time of arrival
<PRI>	PRI of current pulse (0% - 0%)
<PRF>	PRF of current pulse (1/PRI or 0)
<PW>	Pulse width (0% - 0%)
<RT>	Rise time
<FT>	Fall time
<RF>	Center frequency of pulse (absolute)
<BW>	RF bandwidth (FMOP) incl. BB filter
<PA>	Pulse amplitude (at pulse start)
<MF>	MOP flag, where 1 indicates that MOP is active
<MOP>	MOP type, AMOP, PMOD, FMOP
<DF>	Frequency offset from carrier
Global variables from IPM profiles or loops, etc.	
<\$<VarName>>	Variable from the internal variable pool
Automatically generated values related to the sequencing	
<@ITEM>	Sequence Line Item, 1... N
<@REP>	Repetition, 1...M
<@COLL>	Collection Line Item, 1...L
<@RFC>	Center Frequency of Emitter / Sequence

Defining format with the .FMT keyword

Each format line must begin with `.FMT` and follow the syntax:

`.FMT Token:Format:Exponent`

The `.FMT` keyword formats the numeric data output, see [Table A-4](#).

Table A-4: Format keyword syntax description

Parameter	Description
Token	Name of the token
Format	The format of the numeric data is compared to a regular expression: <code>[0-9]*.[0-9]*[+diufFeExX]</code> To prevent e.g. application crash, data that does not match this expression is ignored.
Exponent	Sets an exponent to convert the numeric value; e.g. -9 converts the reported data into nano.

See [Figure A-1](#).

Defining options with the `.OPT` keyword

This keyword defines one or multiple options, that further control the output of the PDW report generator. Each option line must begin with `.OPT` and follow the syntax:

`.OPT TAG=<CURLY>,EXT=<html>`

Table A-5: Option keyword syntax description

Parameter	Description
TAG	TAG=CURLY ANGLE Sets the brackets type
EXT	Sets the extension of the reporting file

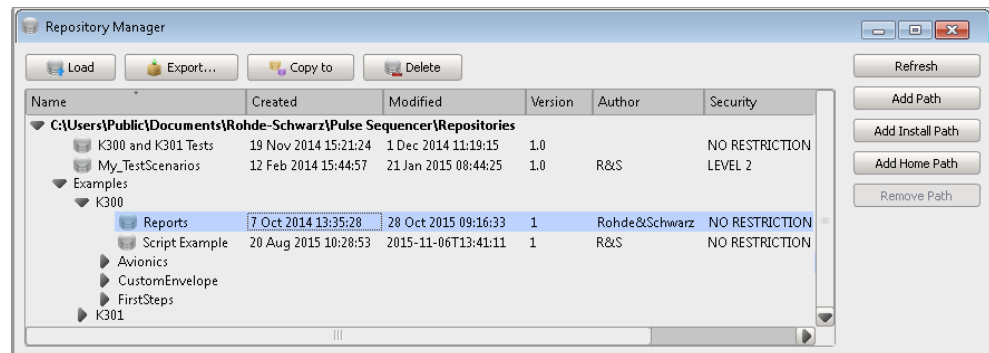
Template examples

Template examples are provided in the repositories supplied with the software.

You can load the example repository or copy&paste the templates content in your own repository, try out them, and further develop them.

Access:

1. Select "File > Load and Manager Repositories".
2. In the "Repository Manager", double click on the "Examples > K300 > Reports".



Created is a new repository with several scenarios using different kind of reporting templates.

- Open a scenario and in the "Scenario" dialog, select "Waveform Generation > Config > Reporting > Template > Edit".

Example: Example of a template that generates report in HTML format

The following is the content of the template that creates a report in HTML format. The report is the same as in the ASCII file example shown on [Figure 21-3](#), but the output is HTML formatted.

```
# Pulse Sequencer Report Template
# The following section defines the HTML report. All values
# in curly brackets are replaced by the actual values.
.OPT TAG=CURLY,EXT=html

<HTML> <HEAD> <TITLE>PDW Report</TITLE> <HEAD>
<BODY> <H2>PDW Report Generated with PS Example</H2>
  <P><TABLE style="width:100%;text-align:left;background-color:gold;">
    <colgroup> <col style="width:30%"> <col style="width:70%"> </colgroup>
    <TR style="background-color:blue;color:white;" >
      <TD>User</TD> <TD>{USER}</TD> </TR>
      <TR> <TD>Repository Path</TD> <TD>{TPATH}</TD> </TR>
      <TR> <TD>Date</TD> <TD>{DATE}</TD> </TR>
      <TR> <TD>ISO Date</TD> <TD>{ISODATE}</TD> </TR>
    </TABLE></P>
  <P><TABLE style="width:100%;text-align:left;background-color:gold;">
    <colgroup> <col style="width:30%"> <col style="width:70%"> </colgroup>
    <TR style="background-color:blue;color:white;" >
      <TD>Repository</TD> <TD>{REPOSITORY}</TD> </TR>
      <TR> <TD>Comment</TD> <TD>{RCOMM}</TD> </TR>
      <TR> <TD>Version</TD> <TD>{VERSION}</TD> </TR>
      <TR> <TD>Author</TD> <TD>{AUTHOR}</TD> </TR>
      <TR> <TD>Created</TD> <TD>{RCDATE}</TD> </TR>
      <TR> <TD>Path</TD> <TD>{RPATH}</TD> </TR>
    </TABLE></P>
  <H3>{SCENARIO}</H3>
  <P>{SCOMM}</P>
```

```

<P><TABLE style="width:100%;text-align:left;background-color:lightgrey;">
  <TR style="background-color:blue;color:white;" >
    <TD>TOA</TD> <TD>RF</TD> <TD>PW</TD> <TD>PA</TD> <TD>MF</TD> <TD>MOP</TD>
    <TD>BW</TD> <TD>Rep</TD> </TR>
  <TR style="background-color:blue;color:white;" >
    <TD>ns</TD> <TD>GHz</TD> <TD>us</TD> <TD>dBm</TD> <TD> </TD> <TD> </TD>
    <TD>kHz</TD> <TD> </TD> </TR>
.HDR
  <TR> <TD>{TOA}</TD> <TD>{RF}</TD> <TD>{PW}</TD> <TD>{PA}</TD> <TD>{MF}</TD>
  <TD>{MOP}</TD> <TD>{BW}</TD> <TD>{@REP}</TD> </TR>
</TABLE>
</BODY>
# The following .FMT lines define the data format
#
.FMT TOA:12.0f:-9
.FMT RF:9.1f:9
.FMT PW:9.3f:-6
.FMT PA:6.1f:0
.FMT BW:6.0f:3
.FMT @REP:3i:0

```

A.2 Antenna Pattern File Formats

R&S Pulse Sequencer supports:

- ["FEKO far field file format *.ffe"](#) on page 486
- ["Antenna Magus file format *.tsv"](#) on page 486
- ["ANSYS HFSS file format *.ffd"](#) on page 486
- ["Antenna patterns in *.csv file format"](#) on page 486
- ["Rohde&Schwarz proprietary antenna pattern *.ant_pat file format"](#) on page 487

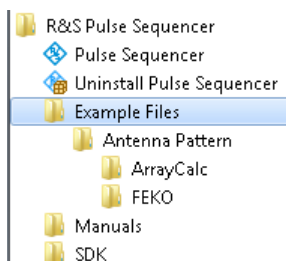
Examples of antenna pattern files

A subset of antenna pattern examples is included in the software.

Access:

- ▶ On your PC, open "Start > All Programs > R&S Pulse Sequencer > Example Files > Antenna Pattern".

You can load and import the patterns, try out them, and further develop them. See ["To import a user-defined antenna pattern from file in one of the predefined file formats"](#) on page 191.



FEKO far field file format *.ffe

FEKO is an electromagnetic simulation software tool distributed by the EM Software & Systems-S.A. (Pty) Ltd.

The R&S Pulse Sequencer supports the *.ffe files with version V1, V2, and V3.

For information on the file format, see the official product page <http://www.feko.info/>.

Antenna Magus file format *.tsv

Antenna Magus is an antenna design software tool.

The *.tsv files are text files that describe the antenna pattern in a tabular structure. These files contain a header section and data part with tab-separated values.

For information, see <http://www.antennamagus.com/>.

Example: Antenna Magus *.tsv file (extract)

```
# Far-field exported from Antenna Magus 5.3.0.1686
# On Friday, March 06, 2015 at 1:18:13 PM
# Frequency: 3.0E+9 Hz
# Power: 0.171146799151093 W
# Theta Samples: 181
# Phi Samples: 361
#Theta   Phi   Re(E_Theta)   Im(E_Theta)   Re(E_Phi)   Im(E_Phi)
0.000000000000E+000  0.000000000000E+000  4.303687774512E-016  0.000000000000E+000  -7.148424634567E+000  -1.436268526906E+000
1.000000000000E+000  0.000000000000E+000  4.425352450194E-016  0.000000000000E+000  -7.363209895869E+000  -1.542584165100E+000
...
```

ANSYS HFSS file format *.ffd

ANSYS HFSS software is the industry standard for simulating 3-D full-wave electromagnetic fields.

The R&S Pulse Sequencer supports the *.ffd multi-frequency far field pattern files but imports only the first frequency.

For information, see <http://www.ansys.com/>.

Example: ANSYS HFSS *.ffd file (extract)

The values in the columns are the complex values of the electromagnetic field in Theta and Phi direction.

```
0 180 19
0 360 37
Frequencies 1
Frequency 8.680000000000000e+008
-3.029304169961667e+000 -3.592562203897908e+000 6.108290737961166e-001 1.376043194687679e+000
-2.877212877279787e+000 -3.299035718428788e+000 1.127582356352358e+000 1.978979886470077e+000
```

Antenna patterns in *.csv file format

Several of the popular antenna design software tools export antenna patterns also as *.csv files.

The *.csv files are text files that describe the antenna pattern in a tabular structure. These files contain a header section and data part with up to six columns. The values can be separated by commas, semicolons, or with spaces.

When a file is selected, the "Import Wizard" tries to determine the used column separator, decimal delimiter and to data content of the columns. You can change the values afterwards.

Example: Antenna Magus file in *.csv format (extract)

```
FRQ=3.0E+9
PIN=0.171146799151093
THETA_SYM_0=0
THETA_SYM_90=0
PHI_SYM_0=0
PHI_SYM_90=0
INTERP_DBI=0
Theta(,deg) Phi(,deg) E_Theta(Re,) E_Theta(Im,) E_Phi(Re,) E_Phi(Im,)

# Far-field exported from Antenna Magus 5.3.0.1686
# On Friday, March 06, 2015 at 1:17:55 PM
# Theta Samples: 181
# Phi Samples: 361
0.000000000000E+000,0.000000000000E+000,4.303687774512E-016,0.000000000000E+000,-7.148424634567E+000,-1.436268526906E+000
1.000000000000E+000,0.000000000000E+000,4.425352450194E-016,0.000000000000E+000,-7.363209895869E+000,-1.542584165100E+000
...
```

Example: ANSYS HFSS files in *.csv format (extract)

The following are two examples of the ANSYS HFSS file in *.csv file format. The first file describes the electromagnetic fields, whereas the second one the gain in dBi.

```
Phi[deg],Theta[deg],re(rEPhi)[mV],im(rEPhi)[mV]
0,0,610.829,1376.04
10,0,1127.58,1978.98
...

Phi[deg],Theta[deg],dB(GainTotal)
0,0,-2.79687
10,0,-2.79687
...
```

Rohde&Schwarz proprietary antenna pattern *.ant_pat file format

The [Table A-6](#) describes the used tags and parameters.

Table A-6: Format of *.ant_pat file

Element	Description
<antenna_pattern>	Root element of the antenna pattern file
<az_res>	Resolution of the columns in the <data> section Value in degrees integer divider of 360

Element	Description
<elev_res>	Resolution of the rows in the <data> section Value in degrees integer divider of 180
<data>	The file has to contain up to: <ul style="list-style-type: none"> • [1 + 360/<az_res>] columns • [1 + 180/<elev_res>] rows If the column resolution exceeds the resolution specified with the <az_res> element, an interpolation is applied. You can also define a subset of values, for example to define a beam antenna pattern. Missing values are internally set to zero, see Example "Beam antenna pattern" on page 488.

The following are two examples of the file format: the description of a beam-like antenna pattern and the description of an antenna pattern with back lobes and medium resolution.

Example: Beam antenna pattern

This example shows that partially defined antenna patterns are also allowed. In

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<antenna_pattern>
  <az_res> 5.00000000e+00 </az_res>
  <elev_res> 5.00000000e+00 </elev_res>
  <data>
    -30,-20,-10,0, 10, 20, 30
    -30,100,100,100,100,100,100,100,
    -20,100,100,100,100,100,100,100,
    -10,100,100,10, 100,10, 100,100,
    0,100,100,100,0, 100,100,100,
    10,100,100,10, 100,10,1 00,100,
    20,100,100,100,100,100,100,100,
    30,100,100,100,100,100,100,100,
  </data>
</antenna_pattern>
```

The example, defined are only the subset of values that describe the beam pattern; all other values are assumed to be zero, see [Figure A-2](#).

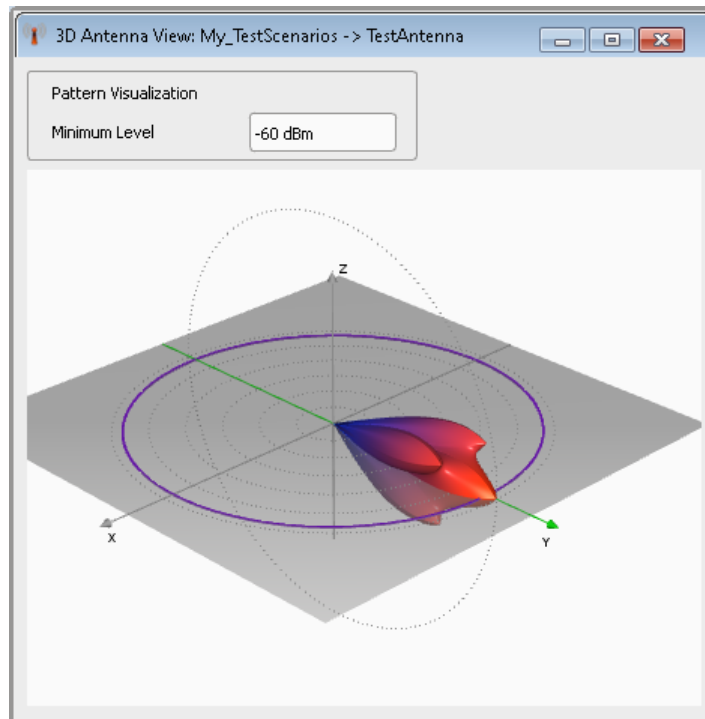


Figure A-2: 3D view of the imported partially defined antenna pattern

Example: Antenna pattern with back lobes

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<antenna_pattern>
<az_res> 5.00000000e+00 </az_res>
<elev_res> 5.00000000e+00 </elev_res>
<data>
-180,-160,-140,-120,-100,-80,-60,-40,-20,0,20, 40, 60, 80, 100,120,140,160,180
-90,100,100,100,100,100,100,100,100,100,100,30,100,100,100,100,100,100,100,100
-80,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
-70,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
-60,100,100,100,100,100,100,100,100,100,100,20, 100,100,100,100,100,100,100
-50,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
-40,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
-30,100,100,100,100,100,100,100,100,100,100,10, 100,100,100,100,100,100,100
-20,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
-10,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
0, 30, 100,100,20 ,100,100,10, 100,100,0, 100,100,10, 100,100,20, 100,100,30
10,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
20,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
30,100,100,100,100,100,100,100,100,100,10, 100,100,100,100,100,100,100,100
40,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
50,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
60,100,100,100,100,100,100,100,100,100,20, 100,100,100,100,100,100,100,100
70,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
80,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
90,100,100,100,100,100,100,100,100,100,30, 100,100,100,100,100,100,100,100
</data>
</antenna_pattern>

```

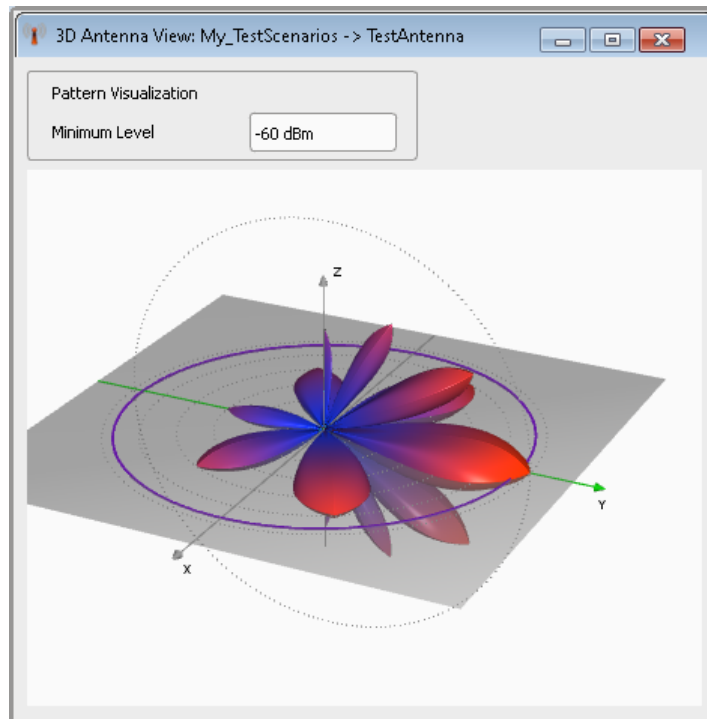


Figure A-3: 3D view of the imported antenna pattern with back lobes

B Formula Syntax

You can use mathematical expression to define custom modulation types or envelope shapes.

The R&S Pulse Sequencer uses the fast math parser library muParser, that is an extensible high performance math expression parser library written in C++.

This section list some of the default features supported by the parser. For detailed information, see the product home page <http://muparser.beltoforion.de/index.html>.

Basic syntax elements

Table B-1: Mathematical operations

Element	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	By the power of

Table B-2: Built-in functions (Extract)

Element	Description
sin	sine function
cos	cosine function
exp	raised to the power of x
etc.	

Table B-3: Other operators

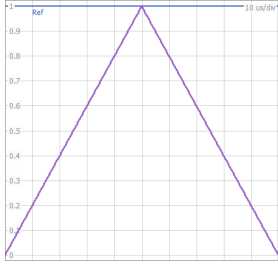
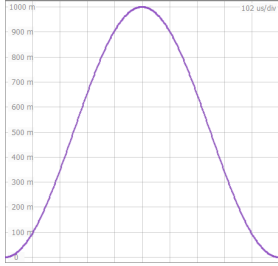
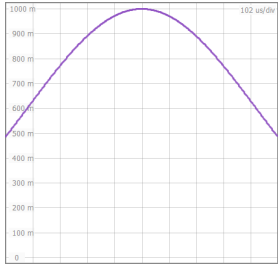
Element	Description	Syntax
?:	if then else operator	<if_condition>?<then_expression>:<else_expression>

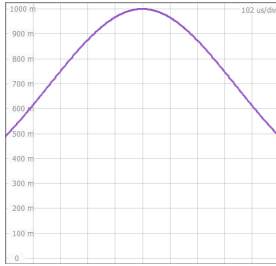
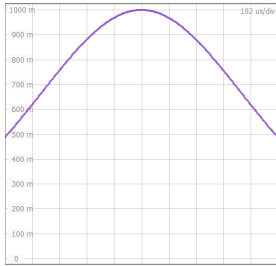
Table B-4: Constant and variables

Element	Description	
Constants:	PI	π
	T_r	Pulse rise time
	T_w	Pulse width time
	T_f	Pulse fall time
	T $T = T_r + T_w + T_f$	Total pulse time

Element		Description
Variables:	ip<n>	Inter-pulse modulation value <n>
	t t = 0 .. T	Time

Equations and parameters used to define custom pulse envelopes

Envelope name	Equation	Envelope shape
"Rectangular Pulse"	1	
"Triangular Pulse"	$(t < T/2) ? t/(T/2) : 1-(t-T/2)/(T/2)$	
"Raised Cosine Pulse"	$0.5 * (1 - \cos(2 * \text{PI} * t/T))$	
"Gaussian Pulse"	$\exp(-4 * \ln(2) * ((t-T_r-T_w/2) / T_w)^2)$	

Envelope name	Equation	Envelope shape
"Lorentzian Pulse"	$1 / ((1 + 4 * (\text{sqrt}(2)-1) * (t-\text{Tr}-\text{Tw}/2) / \text{Tw})^2)^2$	
"Sech Pulse"	$1 / (\cosh((2 * (t-\text{Tr}-\text{Tw}/2) * \ln(1+\text{sqrt}(2))) / \text{Tw}))^2$	

Where Amplitude value range is 0 V to 1 V.

C Plugin Programming API

The functionality of the R&S Pulse Sequencer can be extended by Plugins. A plugin is a Microsoft Windows DLL. This DLL can be loaded into a repository, and then remains there.

You can load Plugins to:

- Specify your custom modulation scheme (MOP)
- Define a custom inter-pulse modulation profile (IPM)
- Create a report file
- Define the file format of the custom antenna pattern files.

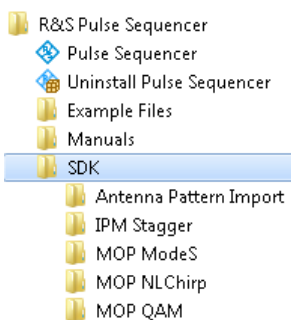
Access:

A subset of example plugins is included in the software.

- ▶ On your PC, open "Start > All Programs > R&S Pulse Sequencer > SDK".

You can load the plugins in your own repository, try out them, and further develop them.

See "To import a plugin" on page 297.



The following is a list of all functions that the plugin has to provide.

• Common Functions	495
• MOP Functions	500
• IPM Functions	505
• Custom Antenna Pattern Import Functions	506
• List of Plugin Functions	509

C.1 Common Functions

getType	495
getVersion	496
getComment	496
getAuthor	497
getName	497
getError	497
int getNextMsg	497
int initPlugin	498
shutdownPlugin	498
int getVariable	498
int setVariable	499

getType (char szModType[1024])

(mandatory)

Queries the plugin type.

The type defines the purpose of the plugin. It is the most important parameter returned to the main application.

Do not change the type of the plugin after it is used within a repository. The type also defines which functions have to be provided by the plugin (API). It also affects in which R&S Pulse Sequencer dialogs the plugin becomes visible.

Parameters:

szModType	string
-----------	--------

(return parameter)
A keyword that denominates the plugin type.

"modulation" | "mop"
Plugin for [MOP](#)

"ipm"
Plugin for [IPM](#)

"report"
Plugin for reports created during the waveform generation

getVersion (char szModVer[1024])

(mandatory)

Queries the version string of the plugin.

The version string is not further evaluated in the software. It only serves as an additional piece of information. The returned version string is visible on the plugin dialog panel.

Parameters:

szModVer	"<Major>.<Minor>.<Revision>"
----------	------------------------------

(return parameter)
<Major>, <Minor>, and <Revision> are numerical values

getComment (char szModComment[4096])

(mandatory)

Queries the comment string of the plugin.

The comment string is not further evaluated in the software. It only serves as an additional piece of information. The returned comment string is visible on the plugin dialog panel.

Parameters:

szModComment	string
--------------	--------

(return parameter)
Line breaks are permissible. To generate new line, insert '`\n`' into the string.

getAuthor (char szModAuthor[1024])

(mandatory)

Queries the author string of the plugin.

The author string is not further evaluated in the software. It only serves as an additional piece of information. The returned author string is visible on the plugin dialog panel.

Parameters:

szModAuthor string
 (return parameter)
 Empty string is allowed

getName (char szModName[1024])

(mandatory)

Queries the plugin name.

The name string is not further evaluated in the software. It only serves as an additional piece of information. The returned name string is visible on the plugin dialog panel.

Return values:

szModName string
 (return parameter)

getError (char szModError[1024])

(mandatory)

Queries the plugins for errors.

This function is called if a plugin function returns an error. In addition, the function is called repeatedly to poll the plugin for errors that are not related to a specific function call. This function must return an empty string if there are no pending error messages. Returning an error stops the R&S Pulse Sequencer calculations.

Error messages should only be returned once.

Parameters:

szModError string
 (return parameter)
 Error message as zero-terminated ASCII string.

int getNextMsg (char szInfoMsg[4096])

(optional)

Sends messages to the application. The messages are passed to the Message Log of the application, merely for user information.

Parameters:

szInfoMsg

<Prefix>:<Text>

(return parameter)

The message may contain a prefix that denominates the type of information sent.

D = Debug

I = Information

W = Warning

E = Error

If no prefix is present, the information type (I) is assumed.

Return values:

boolean

The function is called repeatedly until it returns false (0).

int initPlugin (void)**(optional)**

This function initializes the plug-in. It is called once after the plugin is loaded into memory and may set up internal variables.

Return values:

boolean

true (1)

The initialization completed successfully

false (0)

Error during the initialization. The plugin is unloaded from memory

shutdownPlugin (void)**(optional)**

This function is called before the plugin gets unloaded from memory. This function may be used to release allocated resources. A plugin is unloaded from memory, if it gets removed from the repository or the main application closes.

int getVariable (int iIndex, char szConfig[4096])**(optional)**

Plugins can register variables in the R&S Pulse Sequencer user interface. These variables are initialized with a default value but you can change the value later. The modified variable value is stored as part of the repository. This mechanism allows the reuse of a plugin in different configurations.

This function is called repeatedly by the main application and reads information about the variables to be registered. On each function call, the zero-based variable index value `iIndex` increases by one. The index uniquely references a variable.

Parameters:

<code>iIndex</code>	(input parameter) Zero-based variable index
<code>szConfig</code>	(return parameter) The string of variables to be registered is composed as follows: DBL,<name>,<unit>,<default>,<min>,<max>,<dec. digits> Real Number INT,<name>,<unit>,<default>,<min>,<max> Integer BOOL,<name>,<default> Boolean STR,<name>,<default>[,<validator-regexp>] String SEL,<name>,<opt> <opt> ...,<default index> Selection Value Range of <code><default index></code> : 0 to (number of <code><opt></code> -1)

Return values:

boolean

true
A variable is registered under this index number

false
No variable was registered; the function is not called anymore

Example:

```
SEL, Type, 16-QAM|32-QAM|64-QAM|256-QAM, 2
```

This field is named "Type" and lists the options:
16-QAM, 32-QAM, 64-QAM, and 128-QAM
The default value is 64-QAM

```
int setVariable ( int iIndex, const char szValue[4096] )
```

(optional)

Sets the value of a variable. The variable is referenced by the same zero-based index `iIndex` used in [getVariable\(\)](#). The variable value is transferred as string. This string must be interpreted according to the variable data type.

Parameters:

<code>iIndex</code>	(input parameter) The zero-based index value references the variable
<code>szValue</code>	Zero-terminated ASCII string (input parameter) Variable value as formatted ASCII string. Use functions such as <code>atof()</code> or <code>atoi()</code> for conversion.

C.2 MOP Functions

int isDataNeeded	500
double getModulationClockHint	500
double getModulationSymbolRate	500
setModulationParameter	501
int getBitsNeeded	501
setBits	502
int modulationCore	502
generateDirac	503
setGlobalVariable	503
int getNextReportVariable	504

int isDataNeeded (void)

(optional)

Queries whether this plugin requires data from a data source or not.

Return values:

true (1)

The plugin requires a data source

false (0)

The plugin does not need data from a data source

double getModulationClockHint (void)

(optional)

Queries the minimum clock rate required by the plugin to calculate the I and Q data.

The function is automatically called after the parameterization. This function should evaluate all plugin-internal settings and return the minimum required clock rate to fulfil the Nyquist criteria.

Return values:

If this function is not implemented or the supplied value is outside the limits, the default value is used automatically.

*RST: 100 MHz

double getModulationSymbolRate (void)

(optional)

Queries the symbol rate.

This function informs the main application about the symbol rate. This figure is only required if a baseband filter shall be applied to the plugin I/Q output. If the plugin does not generate signals that require baseband filtering the function can be omitted.

Return values:

The symbol rate in symbols per second.

Range: 0 GHz to 10 GHz

setModulationParameter (const char *szType, const char szValue[1024])

(mandatory)

Sets various parameters before a MOP calculation starts.

This function is repeatedly called from the main application. On each call, the parameter type is defined by a string ID. The value is transferred as string value and has to be interpreted accordingly.

Parameters:

szType	Zero-terminated ASCII string (input parameter) The string ID identifies the MOP parameter "mtime" MOP duration „srat“ Sample rate, used to calculate the MOP „done“ This string ID is sent when the function is called for the last time. Evaluating this ID allows you to check parameters or calculations.
szValue	Zero-terminated ASCII string (input parameter) The numeric value formatted as string. Use functions such as <code>atoi()</code> or <code>atof()</code> to interpret this data.

int getBitsNeeded (double dTime, long *pcStart, int *pcCount)

(optional)

Requests a defined number of bits from the data source.

The function is called by the main application each time before a sample is calculated. For performance reasons, new bits should be requested in blocks and only if they are required for the calculation of the next symbol.

Parameters:

dTime	(input parameter) Time stamp of the following sample
pcStart	(return parameter) Index of the first bit from the data source
pcCount	(return parameter) Number of required bits. Return zero if no bits are required.

Return values:

true (1)
Successful

setBits (long lStart, int iCount, const char *cBuffer)

(optional)

This function is the response to the function [getBitsNeeded\(\)](#). It transmits the requested number bits to the plugin.

Parameters:

lStart	(input parameter) Start index of the first bit; This value matches the <code>pcStart</code> value from <code>int getBitsNeededgetBitsNeeded()</code>
iCount	(input parameter) Number of transmitted bits
cBuffer	(input parameter) The buffer holds the requested number of bits. The bits are transferred in binary format. Each character represents one single bit and has either the value zero or one.

int modulationCore (double dTime, double *pdAmplitude, double *pdPhase, short *puiMkr)

(mandatory)

Transforms samples into I/Q data.

Parameters:

dTime	(input parameter) Time stamp of the sample to be calculated
pdPhase	(input parameter) The phase of the last sample in radian. This parameter is required for modulation types that rotate the phase from sample to sample (e.g. chirps).
pdAmplitude	(return parameter) The voltage-based sample amplitude. This value ranges from zero to one.
pdPhase	(return parameter) The sample phase in radian.
puiMkr	(return parameter) 32-bit unsigned integer bit field defines the marker flags: Bit 0= Marker 1 Bit 1= Marker 2 Bit 3= Marker 3 Bit 4= Marker 4

Return values:**true (1)**

Calculation was successful

generateDirac (int *pcGenI, int *pcGenQ)**(optional)**

Generates a Dirac impulse in the main application for the last sample.

If the function is implemented, it gets called after every call to [modulationCore \(\)](#).

The functionality provided by this function is used if the sample returned by `modulationCore ()` is to be folded with a baseband filter. This, for example, is required if the `modulationCore ()` provides samples used with a digital I/Q modulation such as QPSK or QAM. Typically the output signal is created with oversampling factors of two or larger, thus multiple I/Q samples are generated for each symbol. The folding operation, however, only requires one single sample per symbol as a Dirac impulse. This function indicates which sample is used as the Dirac impulse.

The I and Q channel can be controlled separately. This is required if the constellation points rotate around the origin. In this case, the I and Q waveform are time delayed by a fraction of the symbol duration.

If the function is not implemented all generated samples are assumed to be Dirac pulses. This is equal to a sample and hold operation.

Parameters:***pcGenI, *pcGenQ****(return parameter)**

The return value for `*pcGenI` and `*pcGenQ` is true (1) if the last sample shall be used as the Dirac pulse.

In all other cases, the main application replaces the sample value by zero.

setGlobalVariable (const char szVarName[256], const char szValue[256])**(optional)**

Sends a global variable value to the plug-in.

User defined global variables can be set in the IPM profiles or in the sequences (e.g. loops). In addition, the software provides predefined variables.

Parameters:

szVarName	Zero-terminated ASCII string (input parameter) Variable names always begin with a \$ (dollar sign). The following variables are predefined in R&S Pulse Sequencer: \$RFC RF center frequency, in Hz \$LEV RF level, in dBm \$REP Repetition count of the current sequence line item \$ITEM Item number in a sequence
szValue	Zero-terminated ASCII string (input parameter) Value of the variable formatted as string. Use <code>atoi()</code> or <code>atof()</code> to interpret this parameter.

```
int getNextReportVariable ( int iIndex, const char szName[256], const char
                           szValue[256] )
```

(optional)

MOP plugins may provide variables to the reporting core. This functionality is useful if values calculated by the MOP plugin shall be reported.

The function queries the report variables after a successful calculation and sends them to the report generator. The function is called repeatedly until false (0) is returned.

Parameters:

iIndex	Zero-based variable index (input parameter)
szName	Zero-terminated ASCII string (return parameter) Variable names, without the \$ (dollar sign).
szValue	Zero-terminated ASCII string (return parameter) Variable value as formatted string.

Return values:**true (1)**

Variable data returned for current index

false (0)

No variable data available. The function is not called anymore.

C.3 IPM Functions

setRndFnPtr	505
restart	505
int calculateNextIpmValue	505

setRndFnPtr (void *pRndFn, void *pThis)

(optional)

Sets a pointer to a random function `pRndFn` and the associated `pThis` pointer. Both pointers are needed for calling the built-in random number function.

Plugins must use the random generator provided by the application. This generator produces a random sequence with a defined start seed. If necessary, the application can reproduce a particular calculation with the same values. This mechanism is required because the main application performs an initial dry run to determine parameters such as sequence duration and clock rate. This random generator ensures that the same values are used during dry run and the final calculation.

The prototype of the callback function is defined as follows:

```
typedef int (*RndFnCallback)( void*, int );
```

The random generator is called using the provided pointers:

```
int iRnd = pRndFn( pThis, iMaxVal );
```

The random generator returns a 32-bit random unsigned integer value in the range 0 to 65535.

restart ()

(optional)

Restarts the IPM pattern generation.

This function is called if the IPM pattern generation is requested to restart. The sequence "IPM Configuration" dialog contains the option "Restart IPM for this line item". Setting this option calls this function at the beginning of the line item calculation.

This option is useful if IPM patterns follow a certain rule or have memory, e.g. lists where items shall not be reused.

int calculateNextIpmValue (double dTime, uint64 iCount, double *pdValue)

(mandatory)

Requests the next IPM value from the IPM plugin.

IPM values can either be calculated based on a time stamp or an index. The time is the start time of the current sequence line item, e.g. the pulse start time. The index is an integer number that increases by one each time an IPM value is requested from the plugin.

Parameters:

dTime	(input parameter) Time stamp of the next IPM value
iCount	(input parameter) 64-bit unsigned integer value Zero-based index of the next IPM value
pdValue	(return parameter) Calculated IPM value.

Return values:

true (1)
Calculation successful

C.4 Custom Antenna Pattern Import Functions

The following is a list of functions for importing antenna patterns in custom file format.

These functions are not necessary when importing antenna patterns in one of the files formats listed in [Chapter A.2, "Antenna Pattern File Formats"](#), on page 485.

bool importPattern	506
getConversionInfo	506
int getPatternCount	507
bool getFrequency	507
bool getThetaRange	507
bool getPhiRange	508
bool getPatValue	508
cleanUp	508

bool importPattern (const char *szFile)

(mandatory)

Run the pattern import procedure.

This function imports the antenna pattern into the internal pattern memory. Since this import is done in the foreground thread it should not take too much time.

Return values:

true
Success

getConversionInfo (char szInfo[4096])

(optional)

Get the result from the antenna pattern import.

This function is called after the [importPattern\(\)](#) call. It returns general information and error messages from the import procedure.

Parameters:

szInfo

<Prefix>:<Text>

(return parameter)

The text may contain multiple lines. Each line can start with a prefix that denominates the information type.

D = Debug

I = Information

W = Warning

E = Error

int getPatternCount ()**(mandatory)**

Read the number of imported patterns.

This function is the first one that gets called by the main application after a successful import.

bool getFrequency (int iPatIndex, double *pdFMin, double *pdFMax)**(mandatory)**

Get the frequency for a specific pattern.

This function is called once for each imported pattern. It returns the *frequency range* assigned to an antenna pattern.

Parameters:

iPatIndex

Zero-based antenna index

Return values:**true**

Success

bool getThetaRange (int iPatIndex, double *pdTMin, double *pdTMax, int *piValueCnt)**(mandatory)**

Get the theta range of a specific pattern.

This function is called once for each imported pattern. It returns the *theta range* assigned to an antenna pattern.

Parameters:

iPatIndex

Zero-based antenna index

pdTMin

Minimum theta angle

pdTMax

Maximum theta angle

piValueCnt

Number of theta values

Return values:

true
Success

bool getPhiRange (int iPatIndex, double *pdPMin, double *pdPMax, int *piValueCnt)
(mandatory)

Get the phi range of a specific pattern.

This function is called once for each imported pattern. It returns the *phi range* assigned to an antenna pattern.

Parameters:

iPatIndex	Zero-based antenna index
pdPMin	Minimum phi angle
pdPMax	Maximum phi angle
piValueCnt	Number of phi values

Return values:

true
Success

bool getPatValue (int iPatIndex, int iPhiIndex, int iThetaIndex, float *pfEHorizontal,
float *pfEVertical, float *pfPhaseVertical)
(mandatory)

Get one data point from a pattern.

This function is called once for each imported pattern and once for each data point. It returns the *pattern data* to the main application.

Parameters:

iPatIndex	Zero-based antenna index
iPhiIndex	Zero-based phi index
iThetaIndex	Zero-based theta index
pfEHorizontal	Voltage-based gain, horizontal
pfEVertical	Voltage-based gain, vertical
pfPhaseVertical	Vertical phase with respect to horizontal phase

Return values:

true
Success

cleanUp (void)

Clean up all data.

This function is called after all data has been read from the plugin. All memory can now be freed.

C.5 List of Plugin Functions

bool getFrequency.....	507
bool getPatValue.....	508
bool getPhiRange.....	508
bool getThetaRange.....	507
bool importPattern.....	506
cleanUp.....	508
double getModulationClockHint.....	500
double getModulationSymbolRate.....	500
generateDirac.....	503
getAuthor.....	497
getComment.....	496
getConversionInfo.....	506
getError.....	497
getName.....	497
getType.....	495
getVersion.....	496
int calculateNextIpmValue.....	505
int getBitsNeeded.....	501
int getNextMsg.....	497
int getNextReportVariable.....	504
int getPatternCount.....	507
int getVariable.....	498
int initPlugin.....	498
int isDataNeeded.....	500
int modulationCore.....	502
int setVariable.....	499
restart.....	505
setBits.....	502
setGlobalVariable.....	503
setModulationParameter.....	501
setRndFnPtr.....	505
shutdownPlugin.....	498

D Scripting API

The detailed description of the script syntax is out of scope of this user manual. Refer to the script standard, that can be found under the following link:

<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

Alternatively, any tutorial or documentation on JavaScript could be used.

In addition to the script standard, the R&S Pulse Sequencer introduces 2 proprietary objects: the global object `ps` and the newable object `QVisa`.

The objects `QDir`, `QFile`, `QMessageBox`, `QFileDialog` are Qt classes, where R&S Pulse Sequencer supports only a subset of the methods.

In this section, you find a list of the supported methods per class and the description of the proprietary methods and objects.

The standard Qt classes and methods are not described here. For information, refer to the Qt documentation:

<http://doc.qt.io/qt-5/index.html>

D.1 Global Script Extensions

The global object `ps` extends the script functionality. It provides the following methods within the script context:

QString	<code>scpi</code>	(const QString &sOut)
bool	<code>sync</code>	(int iTimeout)
void	<code>traceError</code>	(const QString &sMsg)
void	<code>traceDebug</code>	(const QString &sMsg)
void	<code>traceInfo</code>	(const QString &sMsg)
void	<code>print</code>	(const QString &sMsg)
void	<code>sleep</code>	(unsigned int iMs)
void	<code>setGlobalVar</code>	(const QString &sToken, const QString &sValue)
QString	<code>getGlobalVar</code>	(const QString &sToken)
QString	<code>getScriptPath</code>	()

ps.scpi	511
ps.sync	511
ps.traceError	511
ps.traceDebug	511
ps.traceInfo	511
ps.print	511

ps.sleep	512
ps.setGlobalVar	512
ps.getGlobalVar	512
ps.getScriptPath	512

ps.scpi (const QString &sOut)

Remote control of the application itself.

This method can be used as command or query. In the latter case, the returned data is a string value.

Example:

```
ps.scpi( '*RST' )
var sIdent = ps.scpi( '*IDN?' )
```

ps.sync (int iTimeout)

Suspends script execution until the R&S Pulse Sequencer has finished a calculation task.

This is done by polling the operation condition register of the SCPI parser and wait for the command `STATus:OPERation:CONDition?` to return zero.

The method returns true if the register value becomes zero within the timeout period `iTimeout`. If there is a timeout, the method returns false.

Example:

```
var iTimoMs = 5000
ps.sync( iTimoMs )
```

ps.traceError (const QString &sMsg)

ps.traceDebug (const QString &sMsg)

ps.traceInfo (const QString &sMsg)

Traces log messages and prints messages to the "Message Log" dialog.

Three methods exist to output text messages to the application message log. By default, the message log shows only information and error messages.

Perform one of the following to enable debug output:

- Start the application with the command-line argument `--debug`
- In the command console window, enter `Set Debug = 1`.
See also ["To open the Message Log dialog in debug mode"](#) on page 478.

Example:

```
ps.info( 'This is an info text.' )
ps.debug( 'This is a debug message.' )
ps.error( 'This is an error message.' )
```

ps.print (const QString &sMsg)

Printing any text or string to the "Script Manager" dialog.

Convert non-string variables into strings before using them with this method.

Example:

```
ps.print( 'My Text' )
var dVal = 1e-3
ps.print( 'Value = ' + dVal.toString() )
```

Manual operation: See "[Script Manager](#)" on page 318

ps.sleep (unsigned int iMs)

Sets a waiting period specified in milliseconds and suspends script execution for at least this amount of time.

Example:

```
var iTimeMs = 500
ps.sleep( iTimeMs )
```

ps.setGlobalVar (const QString &sToken, const QString &sValue)

ps.getGlobalVar (const QString &sToken)

Exchanging Data Between Scripts.

The `ps` script extension provides two methods that access a global variable pool. These variables are of the data type string and remain valid for as long as the application runs. The variables are not saved when the application terminates. Each variable is identified by a name token.

The method `setGlobalVar` registers a new variable or updates an existing variable. The method `getGlobalVar` reads data from a variable. If the variable does not exist, an empty string is returned.

Example:

```
ps.setGlobalVar( 'Var1', 'This is some text.' )
ps.setGlobalVar( 'Var2', dValue.toString() )

dValue = parseFloat( ps.getGlobalVar( 'Var2' ) )
```

ps.getScriptPath ()

Returns the absolute file path of the directory where the script file is stored.

D.2 Newable Script Extensions

The term *newable* describes objects that can be created dynamically using the `new` operator. A script can create multiple instances of these objects. The script engine automatically destroys all dynamically created objects at the end of the script.

D.2.1 QVisa

`QVisa` is a Rohde & Schwarz proprietary class that uses `Qt` data types. The `QVisa` object is a *newable* object that enables remote control access to instruments.

Instrument connections must be defined before data can be exchanged between the script and the instruments. The following structure is mandatory.

```
<Object> = new QVisa();
<Object>.define( '<VISA_String>');
<Object>.open ();

...

<Object>.close ();
```

The following `QVisa` methods are available within the script context:

```
void          define          ( const QString &sRsc )
bool          setTimeout      ( int iValueMs )
bool          open            ( )
bool          close           ( )

bool          write           ( const QString &sOut )
QString       read            ( const QString &sOut )
bool          checkErr        ( )
bool          uploadFile      ( const QString &sLocal, const QString &sRemote )
bool          downloadFile    ( const QString &sLocal, const QString &sRemote )
```

visa.define	513
visa.setTimeout	513
visa.open	514
visa.close	514
visa.write	514
visa.read	514
visa.checkErr	514
visa.uploadFile	514
visa.downloadFile	515

visa.define (const QString &sRsc)

Instrument connections must be defined before data can be exchanged between the script and the instruments.

The device definition is done with the `define` method.

Parameters:

`sRsc` VISA resource string of the instrument

visa.setTimeout (int iValueMs)

Sets the VISA timeout.

Parameters:

`iValueMs` Timeout value in milliseconds

visa.open ()

Opens the VISA connection.

If the device connection was opened successfully, the method returns true.

visa.close ()

Closes a device connection.

The application automatically closes all open connection at the end of the script.

visa.write (const QString &sOut)

Sends a command to the instrument.

Parameters:

sOut Remote command

visa.read (const QString &sOut)

Sends a query to the instrument and then reads the result.

The method returns the data returned from the instrument as string.

Parameters:

sOut Outbound query

visa.checkErr ()

Repeatedly sends the query `SYSTem:ERRor?` on page 455 to the instrument until no more errors are returned.

Return values:

false
If errors were returned

visa.uploadFile (const QString &sLocal, const QString &sRemote)

Sends a file to the instrument.

The method returns true if the transfer was successful. This method uses the SCPI mass memory commands for the file transfer.

The functionality is only tested for Rohde & Schwarz vector signal generators.

Parameters:

sLocal Local file name
sRemote File name on the instrument file system

visa.downloadFile (const QString &sLocal, const QString &sRemote)

Reads a file from the instrument.

The method returns true if the transfer was successful. This method uses the SCPI mass memory commands for the file transfer.

The functionality is only tested for Rohde & Schwarz vector signal generators.

Parameters:

sLocal	Local file name
sRemote	File name on the instrument file system

D.2.2 QDir

For description of the QDir class, see <http://doc.qt.io/qt-5/qdir.html>.

The following QDir methods are available within script context:

```

QString      absoluteFilePath  ( const QString & fileName ) const
QString      absolutePath     ( ) const
bool         cd                ( const QString & dirName )
bool         cdUp              ( )
QString      dirName           ( ) const
bool         exists            ( ) const
QString      filePath          ( const QString & fileName ) const
bool         isReadable        ( ) const
bool         isRoot            ( ) const
bool         mkdir             ( const QString & dirName ) const
bool         mkpath            ( const QString & dirPath ) const
QString      path              ( ) const
bool         remove            ( const QString & fileName )
bool         rmdir             ( const QString & dirName ) const
void         setPath           ( const QString & path )
QStringList  entryList         ( quint32 filters,
                               quint32 sort ) const

// static
QString      currentPath       ( )
QString      homePath          ( )
QString      rootPath          ( )
QString      tempPath          ( )

```

D.2.3 QFile

For description of the QFile class, see <http://doc.qt.io/qt-5/qfile.html>.

The two methods writeString and readLine are Rohde & Schwarz proprietary methods. They work on the QTextStream method.

The following `QFile` methods are available within script context:

```

bool            exists      ( ) const
bool            open        ( quint32 mode )
bool            remove      ( )
void            setFileName ( const QString & name )
bool            copy        ( const QString & newName )
qint64          size        ( ) const
void            close       ( )
quint32         error       ( ) const

// reimplemented on QTextStream
bool            seek        ( qint64 pos )
void            flush       ( )
bool            atEnd       ( ) const

// custom methods (not part of QFile, work on QTextStream)
bool            writeString ( const QString &sStr )
QString         readLine    ( int iMaxLen )

```

file.writeString (const QString &sStr)

Writes the string to the selected file. Strings are appended as separate lines.

file.readLine (int iMaxLen)

Reads the next line of text from the file.

Setting parameters:

`iMaxLen` Maximum number of characters to be read.

D.2.4 QFileDialog

For description of the `QFileDialogFile` class, see <http://doc.qt.io/qt-5/qfiledialog.html>.

The following `QFileDialogFile` methods are available within script context:

```

void            setAcceptMode      ( quint32 mode )
void            setDefaultSuffix   ( const QString & suffix )
void            setDirectory       ( const QString & directory )
void            setFileMode        ( quint32 mode )
void            setNameFilter      ( const QString & filter )
void            setOption          ( quint32 option, bool on )
void            setViewMode        ( quint32 mode )
QStringList     selectedFiles      ( ) const
int             exec                ( )

// static public method

```

```

QString      getOpenFileName      ( const QString & caption,
                                   const QString & dir,
                                   const QString & filter,
                                   quint32 options )

QString      getExistingDirectory( const QString & caption,
                                   const QString & dir,
                                   quint32 options )

QString      getSaveFileName      ( const QString & caption,
                                   const QString & dir,
                                   const QString & filter,
                                   quint32 options )

```

D.2.5 QMessageBox

For description of the `QMessageBox` class, see <http://doc.qt.io/qt-5/qmessagebox.html>.

The following `QMessageBox` methods are available within script context:

```

void          setDetailedText      ( const QString & text )
void          setIcon              ( quint32 Icon )
void          setInformativeText   ( const QString & text )
void          setStandardButtons   ( quint32 buttons )
void          setText              ( const QString & text )
void          setTextFormat        ( quint32 format )
void          setWindowTitle        ( const QString & title )

// static methods
int           critical              ( const QString & title,
                                   const QString & text,
                                   quint32 buttons = QMessageBox::Ok,
                                   quint32 defaultButton = QMessageBox::NoButton )

quint32      information           ( const QString & title,
                                   const QString & text,
                                   quint32 buttons = QMessageBox::Ok,
                                   quint32 defaultButton = QMessageBox::NoButton )

int           question             ( const QString & title,
                                   const QString & text,
                                   quint32 buttons = QMessageBox::StandardButtons
                                   ( QMessageBox::Yes | QMessageBox::No ),
                                   quint32 defaultButton = QMessageBox::NoButton )

int           warning              ( const QString & title,
                                   const QString & text,

```

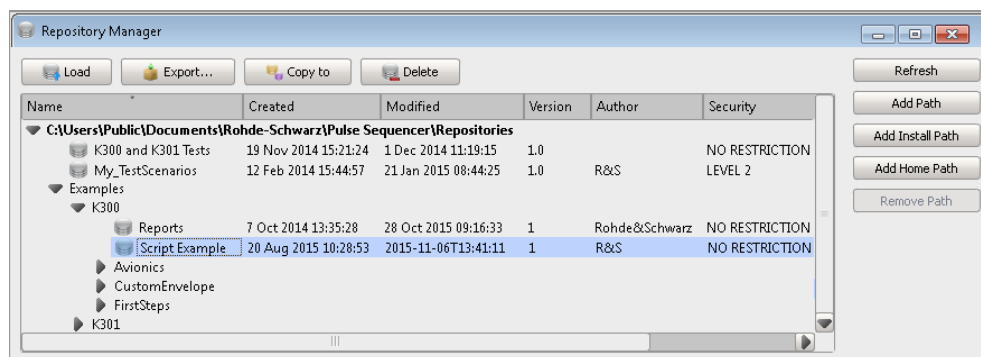
```
quint32 buttons = QMessageBox::Ok,
quint32 defaultButton = QMessageBox::NoButton )
```

D.3 Script Examples

A subset of script examples is provided in the repositories supplied with the software.

Access:

1. Select "File > Load and Manager Repositories".
2. In the "Repository Manager", double click on the "Examples > K300 > Script Example".



3. Select "Script > Manage".

The script examples are loaded in the "Script Manager".

You can try out script examples and further develop them.

The following is the content of one of the scripts.

IPM Diagnostics

```
// This script reads 100 IPM values and writes
// them to a text file. the data can be used for an analysis
// of the IPM data.

// get the temporary path
dir = new QDir();
temppath = dir.tempPath();
fd = new QFileDialog();

// select file
filename = fd.getSaveFileName( 'Save File As', temppath, '', 0 );
if( filename.length > 0 ) {

    // create new IPM profile
    ps.scpi( 'IPM:CRE "ipm_test" );
    ps.scpi( 'IPM:SEL "ipm_test" );
```

```

// switch to normal distribution
ps.scpi( 'IPM:TYPE RAND' );
ps.scpi( 'IPM:RAND:DIST NORM' );

// configure distribution
ps.scpi( 'IPM:RAND:NORM:MEAN 0' );
ps.scpi( 'IPM:RAND:NORM:STD 1' );
ps.scpi( 'IPM:RAND:NORM:LIM 100' );

// start internal diagnostic system
ps.scpi( 'IPM:DIAG:INIT' );

file = new QFile();
file.setFileName( filename );
file.open( 0x0002 );

ps.print( 'Writing file...' );

for( var i=1; i<100; i++ ) {
    value = ps.scpi( 'IPM:DIAG:READ?' );
    file.writeString( value + '\n' );
}

file.close();
ps.print( 'Done' );

// remove IPM profile
ps.scpi( 'IPM:REM "ipm_test"' );
}

```

D.4 List of Script Objects and Methods

file.readLine.....	516
file.writeString.....	516
ps.getGlobalVar.....	512
ps.getScriptPath.....	512
ps.print.....	511
ps.scpi.....	511
ps.setGlobalVar.....	512
ps.sleep.....	512
ps.sync.....	511
ps.traceDebug.....	511
ps.traceError.....	511
ps.traceInfo.....	511
visa.checkErr.....	514
visa.close.....	514
visa.define.....	513

visa.downloadFile.....	515
visa.open.....	514
visa.read.....	514
visa.setTimeout.....	513
visa.uploadFile.....	514
visa.write.....	514

E 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 E-1](#).

Table E-1: Common coding algorithms

Coding	Coding algorithm	Applicable for K bit/symbol
"None"	$b_n = a_n$	$k = 1...8$
"Differential"	$b_n = (a_n + b_{-1}) \text{ modulo } 2^k$	$k = 1...7$
"Differential + Gray"	Gray coding with differential coding	$k = 1...7$

The following example illustrates how one of 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 E-1](#) and for $k = 2$, the recursive coding is defined as $b_n = (a_n + b_{-1}) \text{ modulo } 4$

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

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

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

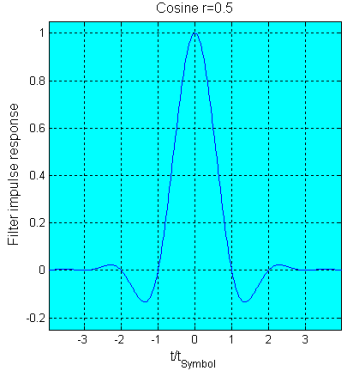
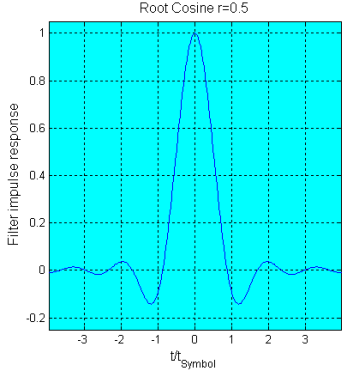
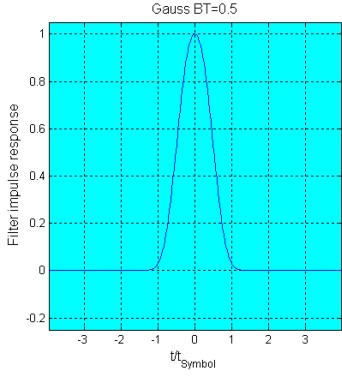
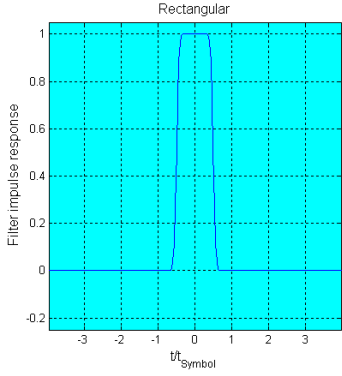
Table E-2: Phase difference for QPSK

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

F Supported Filter and Impact of the Filter Parameters

The [Table F-1](#) shows the filters that are available, together with their associated parameters. The filter characteristic is displayed in graphical form.

Table F-1: Overview of the supported filters

Cosine Roll Off Factor Bandwidth	Root Cosine Roll Off Factor
	
Gauss (FSK) B x T	Rectangular
	

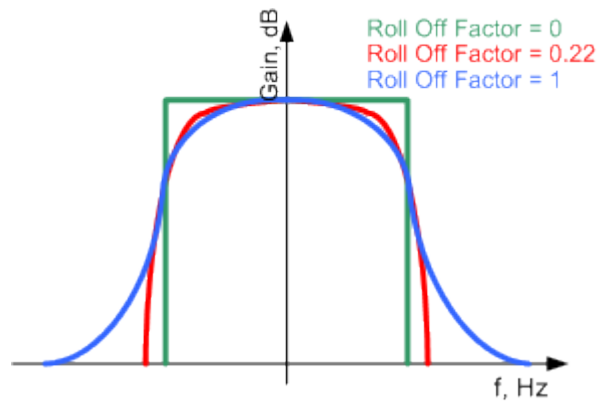
The following is a simple description of the filter parameters and the way they affect the main filter characteristics. Changing filter parameters is an effective way to ensure that the entire bandwidth of the desired signal is allowed to pass and adjust the filter form to reach spectrum mask requirements.

Cut Off Frequency

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

Rolloff Factor

The roll off 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 flanks theoretical possible; values near to 1 make the flanks more flat.



Passband

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

$$\text{Bandwidth} = (1 + \text{"Roll Off Factor"}) * \text{"Symbol Rate"}$$

G References

- R&S SMW200A Vector Signal Generator user manual, the latest version is available for download at [R&S SMW200A product page](#).
- R&S SMBV100A Vector Signal Generator operating manual, the latest version is available for download at [R&S SMBV100A product page](#).
- R&S SGT100A SGMA Vector RF Source operating manual, the latest version is available for download at [R&S SGT100A product page](#).

Glossary: Terms and Abbreviations

A

AM: Amplitude modulation

Antenna: The antenna characterizes the radiation pattern of the [Emitter](#)

ARB: Arbitrary Waveform Generator

An I/Q modulation source forming a part of the supported signal generators. The ARB allows the playback and output of any externally calculated modulation signal in the form of waveform file. It also allows the generation of multi-carrier or multi-segment signals from waveform files.

ASK: Amplitude-shift keying modulation

ASR: Airport Surveillance Radar

AWGN: Additive white gaussian noise

B

Barker: Special sequences (codes) that assures phase modulated signals with low autocorrelation properties.

Boresight: Antenna boresight is the direction to which an antenna shows the maximum gain.

BPSK: Binary phase shift keying modulation

BW: Bandwidth

C

C-BPSK: [BPSK](#) modulation with a constant envelope

Chirp: Signal in which the frequency varies over the time.

CW: Continuous wave signal, that is an unmodulated signal.

D

DQPSK: Differential [QPSK](#)

DUT: Device under test

E

EIRP: Equivalent isotropically radiated power

Emitter: In the context of this software, emitters emulate radar systems.

EW: Electronic warfare

F

FFT: Fast Fourier transform

FM: Frequency modulation

FSK: Frequency-shift keying modulation

G

Gain: Antenna gain is a measure of the antenna's ability to concentrate electromagnetic energy in a narrow beam.

GUI: Graphical User Interface

H

HPBW: Half-Power Beam Width

I

Inter-Pulse Modulation: [IPM](#)

Intra-Pulse Modulation: [MOP](#)

IPM: Inter-Pulse Modulation

M

MOP: Modulation on Pulse

MSW: Multi-segment waveform

O

OQPSK: Offset [QPSK](#)

P

PAPR: Peak to average power ratio

PDW: Pulse Descriptor Word
Format of the reported data.

Plugin: A loadable Microsoft Windows DDL module that describes custom modulation schemes or envelope shapes.

Poly Phase: [polyphase](#)

polyphase: Code used by pulse modulation and suitable for pulse compression.

PRF: Pulse repetition frequency

PRI: Pulse repetition interval
Defines the overall time of a pulse cycle.

PRT: Pulse repetition time

Pulse sequence: [Pulse train](#)

Pulse train: A sequence of repetitive pulses

Pulse-to-Pulse Modulation: [IPM](#)

PW: Pulse width

Q

QPSK: Quadrature phase shift keying modulation

R

RADAR: Radio Detecting and Ranging

RCS: Radar cross section, RCS, or σ
The RCS is a measure of the energy that an object intercepts and scatters back towards the radar.

Repository: In the context of this software, a repository is a file-based database located on the local hard drive or on a network storage drive.

S

Scenario: In the context of this software, a scenario is the top-level description of the signals to be generated.

Sequence: In the context of this software, a sequence describes how pulses are arranged to form a waveform.

Stagger PRI: Pulse train composed of two or more pulse subtrains that use the same PRI.

V

VSG: Vector Signal Generator

Index

Symbols

2D map	
Change units	249
Range	248
3D	
Rotating	22
Zooming, mouse wheel	22
A	
Angular Step	
Spiral scan	184
Antenna	
Create, wizard	29
Model	165
Polarization	166
Show all	252
Antenna pattern	
Azimuth	168
Back lobe	167
Bandwidth	166
Cartesian	168
Comment	165
Cosecant squared	171
Custom	172
Dipole	169, 176
Elevation	168
Frequency	166, 168
Gaussian	169
Horizontal	168
Horn, pyramidal	170
Import form file	177
Import wizard	178
Name	165
Number	168
Overview	161
Parabolic	169
Planar phased array, custom	173
Planar phased array, predefined	171
Polar	168
Settings	164
Sin(x)/x	170
Spatial	168
Supported file formats	177
Vertical	168
XY	168
YZ	168
Antenna scan	
Comment	180
Name	180
Type	180
Antenna scans	
Overview	162
Settings	179
Application cards	11
Application notes	11
ARB file	
Create	62
Load	62
Automation	315
Available connected generators	241

Avoiding data truncation

Rising/falling time	76
---------------------------	----

B

Back lobe	161
Simulate	167
Type	167
Background emitter	
Play time	225
Settings	225
Base power	77
Beam	
Azimuth offset	200
Elevation offset	200
Frequency offset	200
Black	
Item	116
Blank segment	122
Duration	122
Timing	122
Boresight	161
Align to	165
Displacement	165
Brochures	10

C

Cannot change repository	
Obtain write permission	45
Cannot store repository	
Remove write lock	45
Carrier frequency	57
Carrier offset	
see carrier spacing	224
Carrier spacing	
Multi-carrier CW	224
Change units	
Distance	37
Change user	44
Changing colors	
Spectrogram	267
Check generator mapping	
How to access the settings	60, 237
Circular scan	180
Clock rate	
Multitone	221
Waveform, background emitter	225
Waveform, multi-carrier	224
Color	
Background	36
Foreground	36
RGB scheme	36
Traces	36
Color scheme	
Changing	36
Save settings	36
Conditional marker	302
Conical scan	183
Connector	
Find	243, 279

- Create
 - Frequency hopping jitter 149
 - IPM 149
 - Jitter, PRI 149
- Create user 44
- Custom antenna
 - HPBW 172
 - Rolloff factor 172
 - Side lobe level 172
 - Side lobe scale 172
- CW
 - Item 116
- CW segment 122
 - Duration 122
 - Timing 122
- CW signal
 - Settings 224
- D**
- Data sheet 10
- Data Source
 - Comment 294
 - Frequency 294
 - Name 293
 - Preview 295
- Data source guard bits
 - Avoiding data truncation 76
- Data source settings 293
- Data truncation
 - Add guard bits 76
 - Avoiding 76
- DataPath
 - Query 478
- Debugging
 - Info dialog access 478
- Default values
 - Remote 323
- Delete
 - All pulse sequencer waveforms form VSG 60
 - IPM 149
 - Jitter 149
- Delete waveforms
 - Automatically 60
- Delta
 - Frequency 266
 - Marker 266
 - x 266
 - y 266
- Direction
 - Lobe switching scan 185
- Direction finder tests
 - Interferer 219
 - Waveform, multi-carrier 219
- Display
 - Brightness 36
 - Color 36
 - Contrast 36
- Display colors
 - Spectrogram 267
- Display options
 - 2D map 248
- Distance
 - Miles, nautical mile, foot, meter 37
 - Units 37
- Droop 70
 - Definition 77
 - Ratio 77
- Dummy data
 - Item 116
- Dwell time
 - Lobe switching scan 185
- E**
- Emitter
 - Beam 200
 - Comment 199
 - Create, wizard 29
 - EIRP 199
 - Frequency 199, 215
 - Name 199
 - Operating mode 199
- Encoding SQPSK-TG
 - IRIG 106-04 98
- Envelope functions 80
- Error queue query 455, 456
- Export ARB file 62
- F**
- Fall time 70, 76
- Fast data upload
 - FTP 280
- Filter parameter
 - Impact 522
- Flyback time
 - Raster scan 182
 - Sector scan 181
- Frequency diagram 103
- FTP
 - Upload to the signal generator 280
- G**
- Gaming area 248
- Generator
 - Show all 283
- Generator profile
 - ARB 243
 - Baseband bandwidth 243
 - Capabilities 242
 - Comment 241
 - Create, wizard 29
 - Features 242
 - Generator type 241
 - Installed options 241
 - Memory extension 243
 - Name 241
 - Pulse sequencer 242
 - RF settings 243
 - Streams 243
- Generator profiles list
 - Hide unused 277
 - Show all available 277
- Guest user
 - Access rights 44

H

Hardware	
CPU, available	478
CPU, requirements	15
Requirements	15
Virtual Memory	15
Helical scan	184
Horizontal pattern	168

I

I/Q constellation diagram	103
I/Q diagram	103
Identification	
Remote	323
Importing antenna patterns	
File formats	178
Wizard	178
Installation	13
Installing	
Hardware	15
Software	15
Uninstalling old version	15
InstallPath	
see Path	16
Instrument	
Add	238
Create a profile	238
Delete	238
IP address	238
Refresh list	238
Scan interfaces	238
Search for	238
Signal path	58
Instruments settings	237
Inter-pulse modulation (IPM)	117
Interferer	
How to simulate	219, 235
Inverse pulse	
Generation	77
IPM	
Create	149
Create, wizard	29
IPM profile	
Assign to pulse parameter or variable	149
Comment	138
Name	137
Repetition	150
Select	149
Transformation	149
Type	138
Unit	138
View	138
IPM shape	
Equation	143
Interpolated shape	142
List mode	140
Plugin	146
Pseudo random list	143
Random	145
Staircase	139
Steps	139
Waveform	141
Item	
Blank	116
CW	116

Dummy data	116
Filler	116
Inter-pulse modulation (IPM)	117
Overly	116
Repetition	116
Sequencing	116

J

Java scripts	
Executing	317
Jitter	
Add	149
Create	149

L

List all generator profiles	277
Loaded modules (DLLs)	
Query	478
Lobe switching scan	185
Lobes	
Lobe switching scan	185
Lock symbol	
Meaning	45
Locked repository	
Unlock	479
Long waveform	
Reducing size	274

M

Main lobe	161
Map signals to generator	
How to	287
Marker	
Conditional	302
Enable	301
Global	301
Loop run number	302
Pulse	301
Pulse fall time	301
Pulse rise time	301
Pulse width	301
Sequence	301, 302
Show in data view	301
Marker traces	
Enable	266
MOP	
Enable	84
Restrict area	84
MOP Type	86
Multi-carrier	
see Multitone	219
Multi-carrier CW	
Settings	224
Multiple emitters	278, 287

N

Nodding	
Circular scan	180
Sector scan	181

O

OFF time	70
----------------	----

- ON time 70, 76
- Online help 10
- Online manual 10
- Options
 - Identification (remote) 323
- Overlay
 - Item 116
- Oversampling
 - Multitone 221
 - Waveform, background emitter 225
 - Waveform, multi-carrier 224
- Overshoot
 - Definition 78
 - Ratio 78
- P**
- Palmer scan
 - Circular scan 180
 - Raster scan 182
 - Sector scan 181
- Parameters
 - Base Power 77
 - Description 70
 - Droop 77
 - Fall time 76
 - IEEE 181 Standard 70
 - ON Time 76
 - Overshoot 78
 - Pulse Width 76
 - Ripple 79
 - Rise time 76
 - Top Power 77
- Path
 - Query 478
- PDW
 - Report 308
- Phase
 - Set at each pulse start 114
- Phase diagram 103
- Planar phased array
 - Custom 173
 - Distance between elements 171, 173
 - Distribution 171, 173
 - Geometry 173
 - Lattice 173
 - Number of elements 171, 173
 - Parameters 171, 173
 - Pedestal level 171, 173
 - Predefined 171
- Plugin
 - Variable 100
- Polarization
 - Circular 166
 - Co-polarization 166
 - Cross-polarization 166
 - Horizontal 166
 - Linear 166
 - Loss, co-polarization 163
 - Loss, cross-polarization 163
 - Slant angle 166
 - Types overview 162
 - Vertical 166
- Polarization loss
 - Cross-polarization 168
- Power
 - Base 77
 - Top 77
- Pulse
 - Create, wizard 29
 - Envelope shaping, custom 75
 - Envelope shaping, standard 75
 - Fall time, show on graph 101
 - Falling edge 76
 - Falling slope 76
 - Level 105
 - MOP, modulation on pulse 107
 - Name 73
 - Period 70
 - Repetition interval 70
 - Rise time, show on graph 101
 - Rising edge 76
 - Rising slope 76
 - Shape 76
 - Shaping 75
 - Show timing 110
 - Timing 104
 - Timing, visualizing 110
 - Visualizing 110
 - Width 70, 76
 - Width, show on graph 101
- Pulse Train
 - Scenario 55
- Q**
- Query current
 - CPU 478
 - DataPath 478
 - Loaded modules (DLLs) 478
 - Path 478
 - ReportPath 478
 - VISA version 478
- R**
- Radius
 - Conical scan 183
 - Helical scan 184
- Random process
 - Waveform, background emitter 225
- Raster scan 182
- Rate
 - Conical scan 183
 - Helical scan 184
- Receiver
 - Comment 215
 - Name 214
- Receiver tests
 - Interferer 219
 - Waveform, background emitter 219
 - Waveform, multitone 219
- Red lock
 - Repository tree 479
- Ref. level
 - Pulse envelope 57
- Reference level
 - Measure 269
- Release notes 10
- Remote control
 - Enable interface 316
 - Port 316

- Programming examples 322
- SCPI 315
- Remove
 - IPM 149
 - Jitter 149
- Remove waveforms 60
- Repetition
 - Item 116
- Report
 - Plugin 308
 - Template 308
- Reporting
 - Custom template 312
 - Enable 310
 - PDW 310
 - Plugin 312
 - Storage location 310
 - Template 310
 - Template, custom 312
 - Template, edit 311
 - Type 310
- ReportPath
 - Query 478
- Repository
 - Author 43
 - Comment 44
 - Complexity level 43
 - Copy to 48
 - Crash 479
 - Delete 49
 - Discard 49
 - Export 48
 - List of 48
 - Load 48
 - Location 45
 - Name 43
 - Obtain write permission 45
 - Open in Explorer 45
 - Path, add 49
 - Remove 49
 - Remove write lock 45
 - Search 49
 - Security level 43
 - Users 44
- RepositoryPath
 - see DataPath 16
- Reset values
 - Remote 323
- Restart
 - Connected generator 58
- Retrace Time
 - Spiral scan 184
- RF outputs
 - Deactivate all 283
 - Silent 283
- Ripple 79
 - Definition 79
- Rise time 70, 76
- Round Time
 - Spiral scan 184
- Rounds
 - Spiral scan 184
- S**
- Sanitize 60
- Scale
 - Change units 249
- Scan rate
 - Circular scan 180
 - Raster scan 182
 - Sector scan 181
- Scenario
 - 2D map 67
 - Busy 59
 - Comment 56
 - Create ARB file 62
 - Create, wizard 29
 - DF 67
 - Direction finding 67
 - Electro-magnetic spectrum 67
 - Frequency 57
 - Interferer detection 67
 - Multiple emitters 67
 - Name 56
 - Play mode 58
 - Ref. level 57
 - Reset connected generator 58
 - Run 59
 - Settings 55
 - Start 59
 - Stop 59
 - Type 55, 56
- Script 315
- Script editor 317
- Sector scan 181
- Sector width
 - Raster scan 182
 - Sector scan 181
- Security level
 - Repository 43
- Sequence
 - Clear items 116
 - Comment 114
 - Name 114
 - Select item 116
 - Type 114
 - Undo redo 116
- Sequence settings 113
- Serial number
 - Changing 316
- Show
 - Spectrogram 267
 - Spectrum 267
- Show connector 243, 279
- Show timing 101
- Side lobe 161
- Signal generation
 - Start 59
- Socket communication 315
- Software
 - Requirements 15
- SOQPSK-TG
 - Differential encoding 98
- Spectrogram
 - Changing colors 267
 - Rainbow, green, noise, black/white 267
- Spectrum 103
- Spiral scan 184
- Squint angle
 - Lobe switching scan 185
- Start delay
 - 1 ms 289

- Start phase 114
- Storing
 - Reports 310
- T**
- Time delay
 - 1 ms 289
- Time synchronization
 - Filler 122
- Top power 77
- U**
- Unidirectional
 - Raster scan 182
 - Sector scan 181
- Unmodulated signal
 - see CW 224
- Upload
 - Speed up 280
- User account 53
- User data source
 - Rules 292
- User manual 10
- User name
 - Custom 53
 - Set default user 53
- User roles 44
- V**
- Varying
 - Pulse parameters 149
- Vertical pattern 168
- View
 - IPM profile 138
- VISA version
 - Query 478
- W**
- Waterfall diagram
 - see Spectrogram 267
- Waveform
 - Clear 227
 - Comment 224
 - Import 227
 - Name 224
 - Noise bandwidth 227
 - Reference level 227
 - Signal content 227
 - Type 224
 - View 227
 - Visualize 227
- Waveform size
 - Reducing 274
- White noise
 - Bandwidth 227
- White papers 11
- Wizard
 - Emitter, create 29
 - Generator profile, create 29
 - IPM, create 29
 - Pulse, create 29
 - Scenario, create 29
- Write lock
 - Remove 479
- Z**
- Zooming
 - With mouse in 3D graphs 22