# Bluetooth® Enhanced Data Rate Digital Standard for R&S®Signal Generators

## Operating Manual

1171.6496.12 ─ 13

Test & Measurement

Operating Manual

ROHDE&SCHWARZ

This document describes the following software options:

- R&S®SMBV-K60
  1415.8477.xx
- R&S®SMU-K60
  1408.7962.02
- R&S®AMU-K60
  1403.0353.02
- R&S®SMATE-K60
  1404.8412.02
- R&S®SMJ-K60
  1409.2858.02

This manual version corresponds to firmware version:
FW 3.20.281.xx and later of the R&S®SMBV100A
FW 3.20.286.xx and later of the R&S®SMU200A, R&S®SMATE200A, R&S®SMJ100A and R&S®AMU200A


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


The following abbreviations are used throughout this manual: R&S®SMBV100A is abbreviated as R&S SMBV, R&S®SMU200A is abbreviated as R&S SMU, R&S®AMU200A is abbreviated as R&S AMU, R&S®SMATE200A is abbreviated as R&S SMATE, R&S®SMJ100A is abbreviated as R&S SMJ, R&S®WinIQSIM2™ is abbreviated as R&S WinIQSIM2; the license types 02/03/07/11/13/16/12 are abbreviated as xx.

# Contents

# 1 Preface

## 1.1 Documentation Overview

The user documentation for the R&S Signal Generator consists of the following parts:

- Quick start guide, printed manual
- Online help system on the instrument
- Operating manuals and online manual for base unit and options provided on the product page
- Service manual provided for registered users, or on the product page
- Instrument security procedures 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 user documentation on the mainly on the R&S Signal Generator product page.

Additional download paths are stated directly in the following abstracts of the documentation types.

### Quick Start Guide

Introduces the R&S Signal Generator and describes how to set up and start working with the product. Includes basic operations, typical measurement examples, and general information, e.g. safety instructions, etc.

### Online Help

Offers quick, context-sensitive access to the information needed for operation and programming. It contains the description for the base unit and the software options.

### Operating Manuals and Online Manual

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

- **Base unit** manual
  Contains the description of the graphical user interface, an introduction to remote control, the description of all SCPI remote control commands, programming examples, and information on maintenance, instrument interfaces and error messages. Includes the contents of the Quick Start Guide manual.
- **Software option** manuals
  Describe the specific functions of this option. Basic information on operating the base unit is not included.

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

**Service Manual**

Describes the performance test for checking the rated specifications, module replacement and repair, firmware update, troubleshooting and fault elimination, and contains mechanical drawings and spare part lists.

The service manual is available for registered users on the global Rohde & Schwarz information system (GLORIS).

**Instrument Security Procedures**

Deals with security issues when working with the R&S Signal Generator in secure areas.

**Data Sheet and Brochures**

The data sheet contains the technical specifications of the software options, see "Digital Standards for Signal Generators - Data sheet" on the web site.

**Release Notes**

Describes the firmware installation, new and modified features and fixed issues according to the current firmware version. You find the latest version at the product page of the corresponding instrument > "Download" > "Firmware".

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

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

## 1.2 Conventions Used in the Documentation

### 1.2.1 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. |
| *Input* | Input to be entered by the user is displayed in italics. |

| Convention | Description |
|---|---|
| Links | Links that you can click are displayed in blue font. |
| "References" | References to other parts of the documentation are enclosed by quotation marks. |

### 1.2.2 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.

### 1.2.3 Naming of Software Options

In this operating manual, we explicitly refer to options required for specific functions of the digital standard.

The name of software options for signal generators vary in the name of the instrument, but the option name is identical. Therefore we use in this manual the placeholder R&S SMx/AMU.

**Example:**

Naming for an option of the vector signal generator R&S SMBV100A, e.g:

- R&S SMx/AMU-K99, stands for R&S SMBV-K99

The particular software options available for the corresponding instruments are listed on the back of the title page.

# 2 Introduction

The R&S SMx/AMU-K60 is a firmware application that adds functionality to generate signals in accordance with the Bluetooth Specification 2.1+EDR.

To playback a signal from a waveform file created by the simulation software R&S WinIQSIM2, the corresponding R&S WinIQSIM2 digital standard option must be installed.

The basic documents for this specification are regulations for Europe, Japan and North America. The frequency band defined for Bluetooth devices is the unlicensed 2.4 GHz ISM (Industrial Scientific medical) frequency band.

Two modulation modes are used for Bluetooth: a mandatory mode, called the Basic Rate and an optional one, called the Enhanced Data Rate. The Basic Rate mode uses binary FM modulation and has data rate of 1 MBps. The Enhanced Data Rate uses two types of PSK modulation, the π/4-DQPSK or 8DPSK, and achives data rates of 2 Mbps and 3 Mbps, respectively. All modulations schemes have the symbol rate equal to 1Ms/s.

A Time Division Duplex (TDD) scheme for full duplex transmission is defined for both modes.

The latest Bluetooth Low Energy specification is supported as well. See Chapter 3.4, "Bluetooth Low Energy", on page 28 for a detailed description on Bluetooth Low Energy support.

The following list gives an overview of the options provided by the R&S Signal Generator for generating of signals according to the Bluetooth specification:

- Support for three transport modes, the ACL+EDR, SCO, eSCO+EDR transport modes
- Support of all packet types for both the Basic Rate and the Enhanced Data Rate modes
- Generation of signals with up to 5238 frames sequence length
- Configuration of the packet contents with a convenient packet editor or all data packets, both with optional data whitening
- Generation of signals in accordance to the Dirty Transmitter Test specification for both the basic and Enhanced Data Rates and with possibilities to change the start phase, the frequency drift rate and the frequency drift deviation
- Power Ramp Control with possibilities to choose ramp time, rise and fall offset
- Configuration of the clipping, filter and modulation settings

# 2.1 Bluetooth Transport Modes

There are three different transport modes defined in the Bluetooth specification, each of them with special applications:

- Synchronous Connection-Oriented (SCO)
  The SCO transport mode is used for a symmetric point-to-point link establishment between a master and a specific slave in the piconet.
- Extended Synchronous Connection-Oriented (eSCO)
  The eSCO transport mode is used for a symmetric or asymmetric, point-to-point link establishment between the master and a specific slave.
- Asynchronous Connection-Less (ACL)
  The ACL transport mode is used for a point-to-multipoint link establishment between the master and all slaves participating on the piconet.

There are some common transmitted packets used by all transport modes and some specific packets defined for each transport mode.

# 2.2 Bluetooth Packet Types

## 2.2.1 ACL packets

The ACL packets are used for asymmetric links and they contain user data or control data. The table and the figures below give an overview of the ACL packets and their structure.

***Table 2-1: ACL packet - Bascic Rate***

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Slot Num ber |
|---|---|---|---|---|---|
| DM1 | 1 | 0-17 | 2/3 | Yes, 16-bit | 1 |
| DH1 | | 0-27 | no | | |
| DM3 | 2 | 0-121 | 2/3 | | 3 |
| DH3 | | 0-183 | no | | |
| DM5 | | 0-224 | 2/3 | | 5 |
| DH5 | | 0-339 | no | | |
| AUX1 | 1 | 0-29 | | no | |

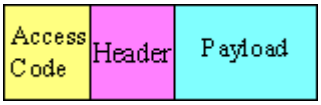



***Figure 2-1: Packet Structure of ACL packets - Basic Rate***

*Table 2-2: ACL packets - Enhanced Rate*

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Slot Num ber |
|---|---|---|---|---|---|
| 2-DH1 | 2 | 0-54 | no | Yes, 16-bit | 1 |
| 2-DH3 | | 0-367 | | | 3 |
| 2-DH5 | | 0-679 | | | 5 |
| 3-DH1 | | 0-83 | | | 1 |
| 3-DH3 | | 0-552 | | | 3 |
| 2-DH5 | | 0-1021 | | | 5 |

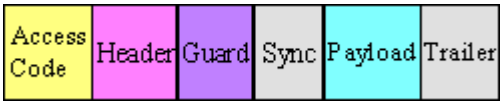



*Figure 2-2: Packet Structure of ACL packets - Enhanced Rate*

## 2.2.2 SCO and eSCO packets

The SCO and eSCO packets are used for symmetric links. The SCO packets are used for 64 kb/s speech transmission and for transparent synchronous data. The eSCO packets are also used for 64kb/s speech transmission and transparent data at 64 kb/s but also at other rates.

The tables and the figures below give an overview of the SCO and eSCO packets and their structure.

*Table 2-3: SCO packets*

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Slot Num ber |
|---|---|---|---|---|---|
| HV1 | n.a. | 10 | 1/3 | no | n.a. |
| HV2 | | 20 | 2/3 | | |
| HV3 | | 30 | | | |
| DV | 1 (Data only) | 10+(0-9) | 2/3 (Data only) | Yes, 16-bit (Data only) | |

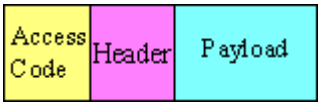



*Figure 2-3: Packet Structure SCO packets*


*Figure 2-4: Packet Structure SCO packets (data only)*

*Table 2-4: eSCO packets - Basic Rate*

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Slot Num ber |
|---|---|---|---|---|---|
| EV3 | n.a. | 1-30 | no | Yes, 16-bit (Data only) | 1 |
| EV4 | | 1-120 | 2/3 | | 3 |
| EV5 | | 1-180 | no | | 3 |

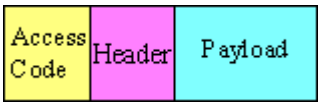



*Figure 2-5: Packet Structure eSCO packets - Basic Rate*

*Table 2-5: eSCO packets - Basic Rate*

| Type | Payload Header (bytes) | User Payload (bytes) | FEC | CRC | Slot Num ber |
|---|---|---|---|---|---|
| 2-EV3 | n.a. | 1-60 | no | Yes, 16-bit | 1 |
| 2-EV5 | | 1-360 | | | 3 |
| 3-EV3 | | 1-90 | | | 1 |
| 3-EV5 | | 1-540 | | | 3 |

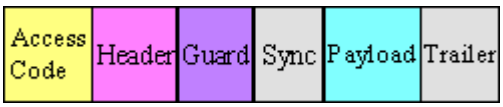



*Figure 2-6: Packet Structure eSCO packets - Enhanced Rate*

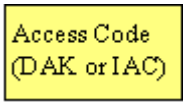
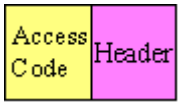
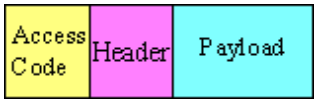
### 2.2.3 Link control packets for ACL, SCO, eSCO transport modes

There are some common kinds of packet types. An overview of these packet types is given in the table below.

*Table 2-6: Common link control packets*

| Transport modes | Type | Payload Header (bytes) | FEC | CRC | Application |
|---|---|---|---|---|---|
| SCO,eSCO,ACL | ID | | | | Paging, inquiry, response |
| SCO,eSCO,ACL | NULL | n.a. | n.a. | n.a. | Carries Link information to the source, e.g. about successfully received signal (ARQN) or the state of the receiving buffer (FLOW) |
| SCO,eSCO,ACL | POLL | | | | Similar to NULL packet, used by master to poll the slaves, must be confirmed |
| SCO,ACL | FHS | 18 | 2/3 | Yes | Page master response, inquiry response, in roll switch |

*Table 2-7: Common link control packets: packet structure*

| Packet Type ID | Packet Types NULL and PULL | Packet Types FHS |
|---|---|---|
|  |  |  |







## 2.3 Packet Structure and Fields

Allmost all Bluetooth transmitted packets have standard format and consist of the access code, the header and the payload with useful information. The exceptions are the ID packet which consists of the access code only and NULL and POLL packets which carry only the access code and the header.

### 2.3.1 Access code

The access code is used for synchronisation, DC offset compensation and identification. The fields of the access code are shown in the figure below and their meaning is explained in the table below.

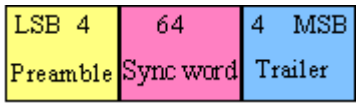


*Table 2-8: The access code fields*

| Field | Description | Packets |
|---|---|---|
| Preamble | A fixed zero-one pattern of 4 symbols, used to facilitate DC compensation | All packets |
| Sync Word | A 64-bit code word derived from a 24 bit address, improves timing acquisition | All packets |
| Trailer | A fixed zero-one pattern of four symbols, extended DC compensation | All packets, except ID |

### 2.3.2 Header

The Header contains link control information. The fields of the header are shown in the figure and their meaning is explained in the table below.

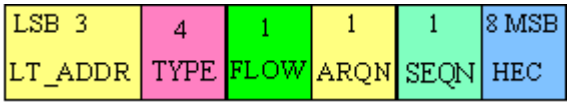


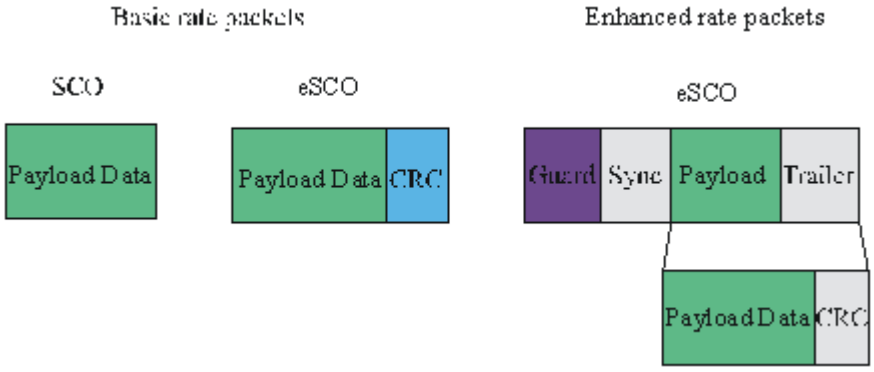

*Table 2-9: The header fields*

| Field | Description | Packets |
|---|---|---|
| LT_ADDR | Logical transport address, indicates the destination slave for a packet in a master-to-slave transmission slot and the source slave for a slave-to-master transmission slot | |
| TYPE | Type code, specifies which packet type is used | |
| FLOW | Flow control, used for flow control of packets over the ACL logical transport. When the RX buffer in the recipient is full, a STOP indication shall be returned. When the RX buffer can accept data, a Go indication shall be returned. | All packets, except ID |
| ARQN | Automatic Repeat Request Number, acknowledgement indication, used to inform the source of a successful transfer of payload data with CRC can be positive acknowledged ACK or negative acknowledged NAK, | |
| SEQN | Sequential numbering scheme to order the data packet stream | |
| HEC | Header-error-check to check the header integrity | |

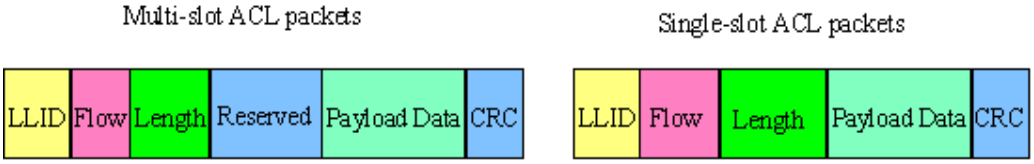
### 2.3.3 Payload format

The payload structure depends on the type of the data field and the data rate. Two fields are defined in the payload: the synchronous data field and the asynchronous data field. The ACL packets only have the asynchronous data field and the SCO and eSCO packets only have the synchronous data field. The exception is DV of SCO transport mode which has both data fields, synchronous and asynchronous.

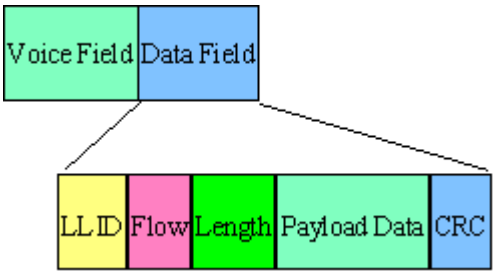
### Synchronous data fields




### Asynchronous data fields




### Synchronous and Asynchronous data fields




The meaning of some payload fields is given in the table below.

***Table 2-10: The payload fields***

| Field | Description |
|---|---|
| CRC | The cyclic redundancy error check |
| Guard, Sync | The guard time and synchronization sequence, used for physical layer change of modulation scheme |
| LLID | The logical link identifier, specifies the logical link |
| Flow | Field which controls the flow on the logical channels |

The payload format and content of the FHS packet are different from other packets. The fields of the FHS packet are shown in the figure below and their meaning is explained in the table below.

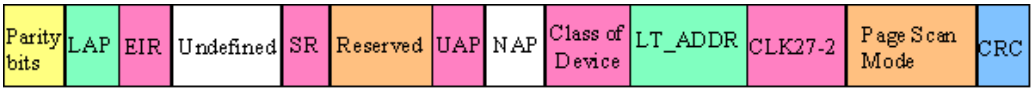


*Table 2-11: The payload fields for the FHS packet*

| Field | Description |
|---|---|
| Parity bits | Form the first part of the sync word of the access code of the device that sends the FHS packet |
| LAP | Contains the lower address part of the device that sends the FHS packet |
| EIR | An extended inquiry response, provides miscellaneous information during the inquiry response procedure |
| Undefined | Reserved for future use and shall be set to zero |
| SR | The scan repetition field, indicates the interval between two consecutive page scan windows |
| Reserved | Shall be set to 10 |
| UAP | Contains the upper address part of the device that sends the FHS packet |
| NAP | Contains the non–significant address part of the device that sends the FHS packet |
| Class of device | Contains the class of device of the device that sends the FHS packet. This field is defined in Bluetooth Assigned Numbers. |
| LT_ADDR | Contains the logical transport address |
| CLK27-2 | Contains the value of the native clock of the device that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet |
| Page scan mode | Indicates which scan mode is used by default by the sender of the FHS packet |

## 2.4 Bluetooth Modulation Schemes

The modulation used for the basic data rate packets is GFSK (Gaussian Frequency Shift Keying) with a bandwidth bit period product BT=0.5. The modulation index is between 0.28 and 0.35.

The modulation scheme used for enhanced data rate packets changes within the packet. The access code and packet header has GFSK modulation scheme and are transmitted with the Basic Rate 1Mbps, while the subsequent synchronisation sequence, payload and trailer sequence have a PSK type of modulation and are transmitted with a data rate of 2 Mbps or optionally 3 Mbps.

The PSK modulation, namely π/4 rotated differential encoded quaternary phase shift keying (π/4–DQPSK) is defined for for the 2 Mbps transmission.

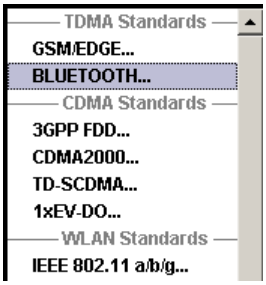
The PSK modulation, namely differential encoded 8-ary phase shift keying (8DPSK), is defined for the 3Mbps transmission.

The modulation types and corresponding packet types are given in the table below.

*Table 2-12: The modulation types and corresponding packet types*

| Modulation type | Packet types |
|---|---|
| GFSK | ID, NULL, POLL, FHS, DM1, DH1, DM3, DH3, DM5, DH5, AUX1, HV1, HV2, HV3, DV, EV3, EV4, EV5 |
| GFSK + π/4-DQPSK | 2-DH1, 2-DH3, 2-DH5, 2-EV3, 2-EV5 |
| GFSK + 8DPSK | 3-DH1, 3-DH3, 3-DH5, 3-EV3, 3-EV5 |

# 3 User Interface




The menu for setting the Bluetooth digital standard is either called from the baseband block or from the menu tree under "Baseband".

The menu is split into several sections for configuring the standard.

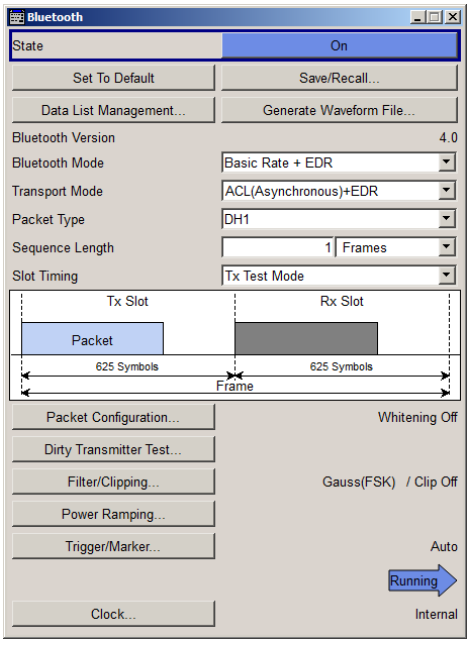
The upper section of the menu is where the Bluetooth digital standard is enabled, the default settings are called, and the transport mode, the packet type and the sequence length are selected.

The valid Bluetooth version in use is displayed.

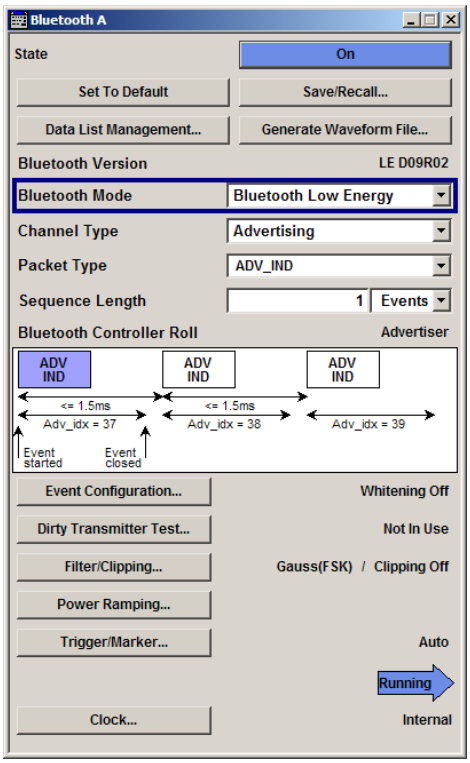
Between an upper part of menu and a lower part of menu Transmission/ ReceptionTX/RX timing is shown. Transmitted packet has duration N × 625 µs where N is an odd integer larger than 0. N depends on the type of the transmitted packet. For Bluetooth Low Energy modes, various packet durations apply depending on the type of channel and packet selected.

Many of the buttons lead to submenus for loading and saving the Bluetooth configuration and for setting the filter, trigger, and clock parameters.

Specific settings of the Bluetooth modes are described in separate chapters. Refer to Chapter 3.2, "Bluetooth Basic Rate + EDR", on page 22 and Chapter 3.4, "Bluetooth Low Energy", on page 28.




Bluetooth Basic Rate + EDR




Bluetooth Low Energy

The screenshots provided in this description show parameter values that have been selected to illustrate as much as possible of the provided functions and possible inter-dependencies between them.

These values are not necessarily representative of realistic test situations.

## 3.1 General Settings for Bluetooth Signals

The upper menu section is where the Bluetooth digital standard is enabled and reset and where the transport mode, packet type and sequence length are selected.

**State**
Activates the standard and deactivates all the other digital standards and digital modulation modes in the same path.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:STATe` on page 71

**Set To Default**
Calls the default settings. The values of the main parameters are listed in the following table.

| Parameter | Value |
|---|---|
| State | Not affected by "Set To Default" |
| Bluetooth Version | 2.1 + EDR (2.1 = current version number) |
| Bluetooth Mode | Basic Rate + EDR |
| Transport mode | ACL (Asynchronous) + EDR |
| Packet type | DH1 |
| Sequence length | 1 Frames |
| Slot Timing | Tx Test Mode |
| Packet configuration | Packet Editor/ Whitening off |
| Dirty Transmitter Test | Not in Use |
| Filter/Clipping | Gauss (FSK) / Clipping off |
| Power ramping | Cosine / 1 Symbols |
| Trigger/Marker | Auto |
| Clock | Internal |

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PRESet` on page 68

**Save/Recall...**
Calls the "Save/Recall" menu.

From the "Save/Recall" menu, the "File Select" windows for saving and recalling Bluetooth configurations and the "File Manager" are called.


Bluetooth configurations are stored as files with the predefined file extension *.bto. The file name and the directory they are stored in are user-definable.

The complete settings in the Bluetooth menu are saved and recalled.

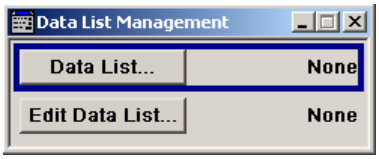
"Recall Bluetooth Setting" Opens the "File Select" window for loading a saved Bluetooth configuration.
The configuration of the selected (highlighted) file is loaded by pressing the "Select" button.

"Save Bluetooth Setting" Opens the "File Select" window for saving the current Bluetooth signal configuration.
The name of the file is specified in the "File name" entry field, the directory selected in the "Save Into" field. The file is saved by pressing the "Save" button.
The "Fast Save" checkbox determines whether the instrument performs an absolute or a differential storing of the settings. Enable this function to accelerate the saving process by saving only the settings with values different to the default ones. "Fast Save" is not affected by the "Preset" function.

"File Manager" Calls the "File Manager".
The "File Manager" is used to copy, delete, and rename files and to create new directories.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:SETTing:CATalog` on page 69
`[:SOURce<hw>]:BB:BTOoth:SETTing:LOAD` on page 70
`[:SOURce<hw>]:BB:BTOoth:SETTing:STORe` on page 70
`[:SOURce<hw>]:BB:BTOoth:SETTing:STORe:FAST` on page 70
`[:SOURce<hw>]:BB:BTOoth:SETTing:DELete` on page 69

**Data List Management…**
Calls the "Data List Management" menu. This menu is used to create and edit a data list.




All data lists are stored as files with the predefined file extension `*.dm_iqd`. The file name and the directory they are stored in are user-definable.

The data lists must be selected as a data source from the submenus under the individual function, e.g. in the channel table of the cells.

**Note:** All data lists are generated and edited by means of the `SOURce:BB:DM` subsystem commands. Files containing data lists usually end with `*.dm_iqd`. The data lists are selected as a data source for a specific function in the individual subsystems of the digital standard.

**Example: Creating and editing the data list**

```
SOUR:BB:DM:DLIS:SEL 'd_list1'
SOUR:BB:DM:DLIS:DATA #B1111010101000001111....
SOUR:BB:DM:DLIS:DATA:APP #B1111010101000001111....
```

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA` on page 95
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DSELection` on page 96

**Generate Waveform File...**
Calls the "Generate Waveform" menu. This menu is used to store the current Bluetooth signal as ARB signal in a waveform file.

This file can be loaded in the "ARB" menu and processed as multicarrier or multisegment signal.

The file name is entered in the submenu. The file is stored with the predefined file extension `*.wv`. The file name and the directory it is stored in are user-definable.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:WAVeform:CREate` on page 72

**Bluetooth Version**
Displays the current version of the Bluetooth / Bluetooth LE standard.

The default settings and parameters provided are oriented towards the specifications of the version displayed.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:VERSion?` on page 72

**Bluetooth Mode**
Determines the digital Bluetooth standard. Basic Rate + EDR or Bluetooth Low Energy are available.

"Basic Rate +EDR"
Selects the Bluetooth mode Basic Rate + EDR.
Specific settings of the basic mode are described in Chapter 3.2, "Bluetooth Basic Rate + EDR", on page 22.

"Bluetooth Low Energy"
Selects the Bluetooth mode Low Energy. The settings concerning Bluetooth Low Energy mode are described in Chapter 3.4, "Bluetooth Low Energy", on page 28.

Remote command:
[:SOURce<hw>]:BB:BTOoth:BMODe on page 108

**Filter/Clipping**
Calls the dialog for setting baseband filter and clipping of the generated signal. The current filter and the clipping state are displayed next to the button.

The dialog is described in Chapter 3.8, "Filter/Clipping Settings", on page 52.

Remote command:
n.a.

**Power Ramping**
Calls the dialog for setting the power ramping.

The dialog is described in Chapter 3.9, "Power Ramping Settings", on page 55.

Remote command:
n.a.

**Trigger/Marker**
Calls the dialog for selecting the trigger mode and trigger source, for configuring the marker signals, and for setting the time delay of an external trigger signal.

The currently selected trigger mode and trigger source are displayed next to the button.

The dialog is described in Chapter 3.10, "Trigger/Marker/Clock Settings", on page 56.

Remote command:
n.a.

**Execute Trigger**
Executes a trigger manually. A manual trigger can be executed only if an internal trigger source and a trigger mode other than "Auto" have been selected.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:EXECute on page 80

**Arm**
Stops signal generation manually. This button appears only with "Running" signal generation in the "Armed_Auto" and "Armed_Retrigger" trigger modes.

Signal generation can be restarted by a new trigger (internally with "Execute Trigger" or externally).

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:ARM:EXECute on page 79

**Clock**
Calls the dialog for selecting the clock source and for setting a delay, see Chapter 3.10, "Trigger/Marker/Clock Settings", on page 56.

Remote command:
n.a.

## 3.2 Bluetooth Basic Rate + EDR

The dialog contains the parameters to define the packet type and provides access to the packet type configuration dialog. The graphic shows the frame structure of the selected packet type.

**Transport Mode**
Only available for "Bluetooth Mode " set to "Basic Rate + EDR"

Selects the transport mode.

"ACL+EDR" The transport mode selected is used for a point-to-multipoint link establishment between the master and all the slaves participating on the piconet.

"SCO" The transport mode selected is used for a point-to-point link establishment between a master and a single slave in the piconet.

"eSCO+EDR" The transport mode selected is used for a symmetric or asymmetric point-to-point link establishment between a master and a specific slave.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TMODe` on page 71

**Packet Type**
Selects the packet type.

The available packets depend on the selected Transport Mode.

All packet types as defined in the Bluetooth specification are supported. For an overview, see Chapter 2.2, "Bluetooth Packet Types", on page 9.

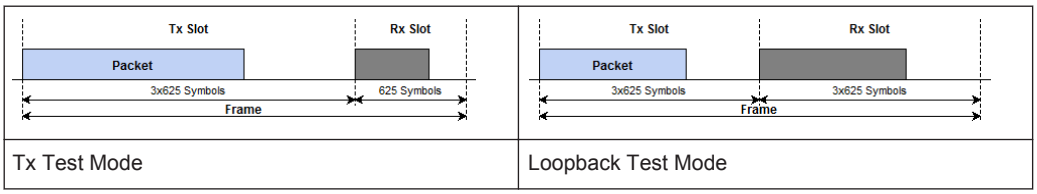
Remote command:
`[:SOURce<hw>]:BB:BTOoth:PTYPe` on page 69

**Sequence Length**
Selects the sequence length in frames of the generated signal. The signal repeats after the specified number of frames.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:SLENgth` on page 71

**Slot Timing**
Selects the timing mode for the Rx slot.

The graphic below shows the frame structure of the selected Packet Type and slot timing.




Tx Test Mode | Loopback Test Mode

A transmitted packet has a duration of N × 625 µs where N is an odd integer larger than 0. N depends on the type of the transmitted packet. In "Tx Test" mode, N = 1 for Rx slots.

"Tx Test Mode"
The transmitted Rx package takes 625 symbols, regardless of the selected packet type.

"Loopback Test Mode"
Extends the Rx slot time according to the selected packet type. For example, the Rx slot of Packet Type > DH3 takes 3 x 625 symbols.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:STIMing` on page 71

**Packet Configuration**
Access the "Packet Configuration" dialog, see Chapter 3.3, "Packet Configuration - Bluetooth Basic Rate + EDR", on page 23.

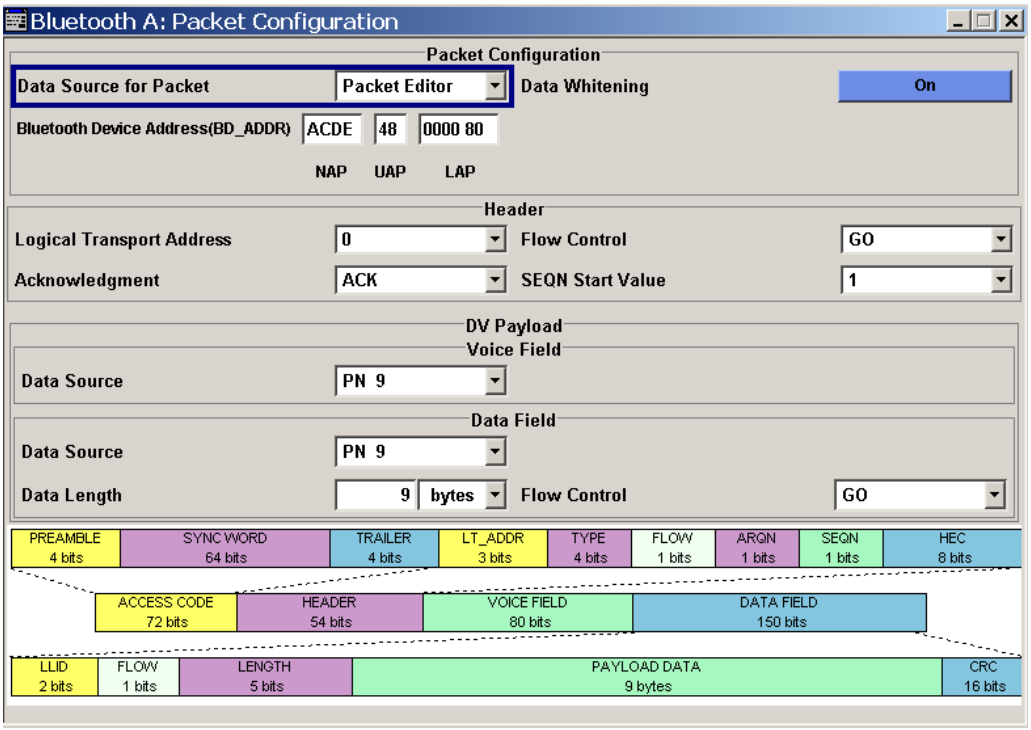
The current data source for packet and the data whitening state are displayed next to the button.

Remote command:
n.a.

## 3.3 Packet Configuration - Bluetooth Basic Rate + EDR

The "Packet Configuration" dialog is accessed via the "Bluetooth" main dialog.

The available settings in the lower part of the dialog depend on the selected data source for packet and the selected packet type. Header configurations are selected in the "Header" settings section.




**Data Source for Packet**
The data sent for each packet can be comfortably edited with the Packet Editor, or filled with a predefined ALL Data sequence.

"Packet Editor" Enables the edit mode to configure the packet fields individually.

"All Data" Fills the generated packets with the selected data source. This mode is useful if you need to load predefined data contents from a data list file or the data contents of the packet are not of interest.

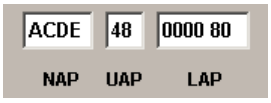
Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DSFPacket` on page 97

**Data Whitening**
Activates the Data Whitening.

Evenly distributed white noise is ideal for the transmission, and real data can be forced to look similar to white noise with different methods called Data Whitening.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DWHitening` on page 97

**Bluetooth Device Address (BD_ADDR)**
Enters Bluetooth Device Address. Each Bluetooth device shall be allocated a unique 48-bit Bluetooth device address (BD_ADDR).




The BD_ADDR may take any values except the 64 reserved LAP values: 0x9E8B00 – 0x9E8B3F.

"NAP" Selects non-significant address part.
The length of NAP is 16 bits or 4 hexadecimal figures.

"UAP" Selects upper address part.
The length of UAP is 8 bits or 2 hexadecimal figures.

"LAP" Selects lower address part.
The length of LAP is 24 bits or 6 hexadecimal figures.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDANap` on page 94
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDAUap` on page 94
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDALap` on page 94

**Logical Transport Address**
(Available for all packet types except ID)

Enters the logical transport address for the header.

Each slave active in a piconet is assigned a primary logical transport address (LT_ADDR). The all-zero LT_ADDR is reserved for broadcast messages.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:LTADdress` on page 98

**Flow Control (Header)**
(Available for all packet types except ID)

Sets the FLOW bit in the header. This bit indicates start or stop of transmission of packets over the ACL logical transport.

"Go" Allows the other devices to transmit new data.

"Stop" Stops the other devices from transmitting data temporarily.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:HFControl` on page 98

**Acknowledgment**
(Available for all packet types except ID)

Sets the ARQN bit of the packet header.

"NAK" Request to retransmit the previous payload.

"ACK" Previous payload has been received successfully.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:ACKNowledgement` on page 93

**SEQN Start Value**
(Available for all packet types except ID)

Sets the start value of the header SEQN bit.

The SEQN bit is present in the header to filter out retransmissions in the destination. The signal generator is altering this bit automatically on consecutive frames, if a sequence length of at least 2 frames is set.

"0" The SEQN bit starts with 0.

"1" The SEQN bit starts with 1.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:SNSValue on page 100

**Data Source**
(Available for all packet types except ID, POLL, NULL and FHS packets)

Selects the data source used for the payload.

The following standard data sources are available:

- "All 0, All 1"
  An internally generated sequence containing 0 data or 1 data.
- "PNxx"
  An internally generated pseudo-random noise sequence.
- "Pattern"
  An internally generated sequence according to a bit pattern.
  Use the "Pattern" box to define the bit pattern.
- "Data List/Select DList"
  A binary data from a data list, internally or externally generated.
  Select "Select DList" to access the standard "Select List" dialog.
  – Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
  – Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
  – Use the standard "File Manager" function to transfer external data lists to the instrument.

  See also "Main Dialog > Data List Management".

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA on page 95
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DPATtern on page 95
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DSELection on page 96

**Data Length**
(Available for all packet types except ID, POLL, NULL and FHS packets)

Enters the payload data length in bytes.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DLENgth on page 96

**Flow Control (Payload)**
(Available for all packets types except ID, POLL, NULL, FHS, HV1, HV2, HV3, EV3, EV4, EV5, 2-EV3, 2-EV5, 3-EV3, 3-EV5 packets.)

Sets the FLOW bit in the payload (flow control per logical link)

| | |
|---|---|
| "Go" | Indicates start of transmission of ACL packets after a new connection has been established. |
| "Stop" | Indicates stop of transmission of ACL packets before an additional amount of payload data is sent. |

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:PFControl on page 99

**Packet Length**
(Available in All Data mode and for all packet types except ID packet)

Enters the packet length in symbols.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:PLENgth on page 99

**EIR packet follows**
(Available for FHS packets)

Indicates that an extended inquiry response packet may follow.

"Yes" Indicates that an EIR packet follows.

"No" Indicates that EIR does not follow.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:EIRPacketfollows
on page 97

**Scan Repetition Mode**
(Available for FHS packets)

The 2-bit scan repetition field indicates the interval between two consecutive page scan windows, determines the behavior of the paging device.

"R0" The scan interval is equal to the scan window Tw page scan (continuous scan) and maximal 1.28s.

"R1" The scan interval is maximal 1.28s.

"R2" The scan interval is maximal 2.56s.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:SRMode on page 100

**Class of Device**
(Available for FHS packets)

A parameter received during the device discovery procedure, indicates the type of device and which types of service that are supported.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PCONfiguration:CODevice on page 95

**Data Source (Voice Field)**
(Available for DV packets)

Selects the Data Source for the voice field.

The following standard data sources are available:

- "All 0, All 1"
  An internally generated sequence containing 0 data or 1 data.
- "PNxx"
  An internally generated pseudo-random noise sequence.
- "Pattern"

An internally generated sequence according to a bit pattern.
Use the "Pattern" box to define the bit pattern.

- "Data List/Select DList"
  A binary data from a data list, internally or externally generated.
  Select "Select DList" to access the standard "Select List" dialog.
  – Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
  – Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
  – Use the standard "File Manager" function to transfer external data lists to the instrument.

  See also "Main Dialog > Data List Management".

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:VDATa` on page 100
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDPAttern` on page 96
`[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDSElection` on page 96

## 3.4 Bluetooth Low Energy

The R&S Signal Generator provides you with the ability to generate signals in accordance with Bluetooth Low Energy Specification (LE). Bluetooth Low Energy makes it possible to transfer data from low power devices running on the smallest of batteries to a larger device, such as a PC, a mobile phone, or a PDA. For the first time, a Bluetooth connection to a wristwatch, or a heart rate sensor, or a data transfer from a digital camera, is possible. The Bluetooth low energy chips will offer capabilities that do not replace or supersede the existing Bluetooth 2.x standards. Data rates are comparable to Bluetooth 1.1, and are data-only (no audio content).

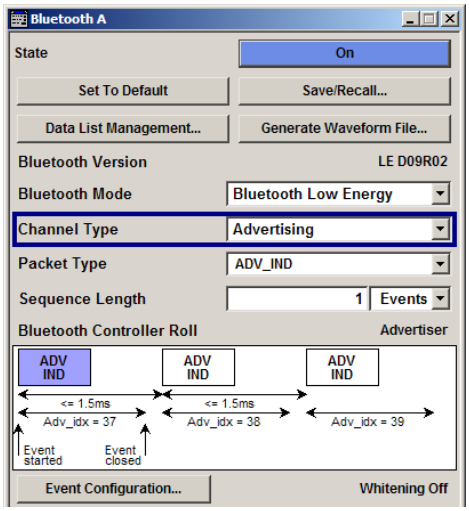
Bluetooth LE mode only uses the Basic Rate. The Basic Rate mode uses binary FM modulation and has a data rate of 1 MBps. The modulation scheme has the symbol rate equal to 1Ms/s.

For full duplex transmission, a Time Division Duplex (TDD) scheme is used.

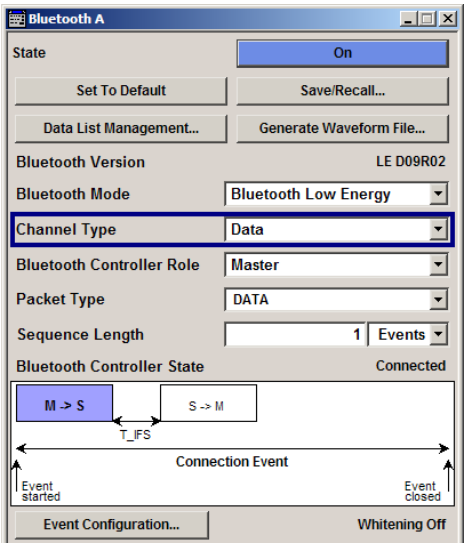
The following list gives an overview of the options provided by the R&S Signal Generator for Bluetooth LE compliant signal generation:

- Support for two channel types, the Advertising and Data channel types.
- Support of all Bluetooth LE packet types.
- Sequence Length unit can be set to Event or Frame.
- Convenient packet editor for all supported packet types including optional data whitening.
- Dirty Transmitter Test compliant to RF test specification, with options to change start phase, frequency drift rate and frequency drift deviation.
- Power Ramp Control with configurable ramp time, rise and fall offsets.
- Clipping, filter and modulation settings supported.

In the following description Bluetooth Low Energy is abbreviated as Bluetooth LE.




Bluetooth LE "Channel Type Advertising"




Bluetooth LE "Channel Type Data"

**Channel Type**
Determines the channel type. Advertising and data are available. Refer to Chapter 3.5, "Event / Frame Configuration - Bluetooth LE", on page 32 for setting the respective parameters.

"Advertising" Selects channel type Advertising.

"Data" Selects channel type Data.
Devices in a connected state transmit data channel packets in connection events with a start point and an interval.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:CTYPe` on page 68

**Packet Type**
Selects the packet type.

The available packet types depend on the selected channel type, as shown in the table below (Table 3-1).

***Table 3-1: Packet types of the respective channel types:***

| Packet Type | Advertising | Data |
|---|---|---|
| ADV_IND/ | x | - |
| ADV_DIRECT_IND | x | - |
| ADV_NONCONN_IND | x | - |
| ADV_DISCOVER_IND | x | - |
| SCAN_REQ | x | - |
| SCAN_RSP | x | - |
| CONNECT_REQ | x | - |

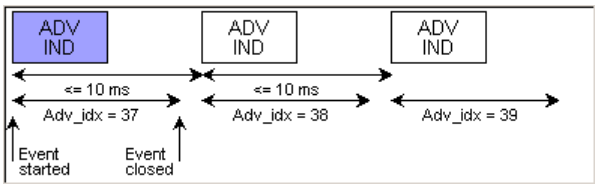
| Packet Type | Advertising | Data |
|---|---|---|
| DATA | - | x |
| CONTROL_DATA ... | - | x |
| TEST PACKET | x | x |

Depending on the Bluetooth Controller Role (master or slave), you can determine in detail the information of the "CONTROL_DATA", as shown in the following table Table 3-2.

***Table 3-2: Control information, available for master or slave.***

| CONTROL_DATA | Master | Slave |
|---|---|---|
| LL_CONNECTION_UPDATE_REQ | x | - |
| LL_CHANNEL_MAP_REQ | x | - |
| LL_TERMINATE_IND | x | - |
| LL_ENC_REQ | x | - |
| LL_RNC_RESP | - | x |
| LL_START_ENC_REQ | x | x |
| LL_START_ENC_RESP | x | x |
| LL_UNKNOWN_RESP | - | x |
| LL_FEATURE_REQ | x | - |
| LL_FEATURE_RESP | - | x |
| LL_PAUSE_ENC_REQ | x | - |
| LL_PAUSE_ENC_RESP | - | x |
| LL_VERSION_IND | x | x |
| LL_REJECT_IND | x | x |

The graphic shows the frame structure of the selected packet type.




Remote command:
`[:SOURce<hw>]:BB:BTOoth:UPTYpe` on page 127

**Sequence Length**
Selects the number of frames or events depending on the packet type. The signal repeats after the specified number of frames/events.

For SCAN_REQ and CONNECT_REQ packet, the sequence length is expressed in "Frames".

For TERMINATE_IND packets, a default value according to the specification is given:
- Master: 'SlaveLatency + 6'
- Slave: '6'

For all other packet types the sequence length is expressed in "Events".

Remote command:
`[:SOURce<hw>]:BB:BTOoth:USLength` on page 128

**Bluetooth Controller Role**
Determines the controller role.

**Note:** The available packet types for the selected channel types and the controller roles are described in "Bluetooth Controller Role" on page 31.

Depending on the channel type, the field either displays the appropriate role or you can select one:
- "Advertiser"
  Displays the controller role corresponding to the packet type:
  – "Advertiser" for all ADV-xxx packet types and SCAN_RSP
  – "Scanner" for SCAN_REQ packet type
  – "Initiator" for CONNECT_REQ packet type
- "Data"
  Assigns a role to the controller:
  – "Master"
  – "Slave"

Remote command:
`[:SOURce<hw>]:BB:BTOoth:BCRole` on page 108

**Bluetooth Controller State**
Shows the state of the bluetooth controller for channel type "Data".

Remote command:
`[:SOURce<hw>]:BB:BTOoth:BCText?` on page 67

**Event / Frame Configuration**
Provides access to the "Event Configuration" dialog, if the sequence length of the packet type is expressed in events, and accordingly, the "Frame Configuration" dialog, if it is expressed in frames, see Chapter 3.5, "Event / Frame Configuration - Bluetooth LE", on page 32.

The data whitening state is displayed next to the button.

Remote command:
n.a.

**Test Packet Configuration**
Provides access to the "Test Packet Configuration" dialog for packet type "TEST PACKET", see Chapter 3.6.2, "Test Packet Configuration Settings", on page 46.

## 3.5 Event / Frame Configuration - Bluetooth LE

The "Event/ Frame Configuration" dialogs are reached via the "Bluetooth" main dialog.

The upper area of the dialogs changes, as a function of the used channel type. For channel type "Data", the section provides "Connection Settings", and for channel type "Advertising" the "Advertising Settings" are given with regard to this channel type. The "Packet Configuration" subdialog can be opened, and a graph represents the physical channel mapping and the channel indices. The table in the lower section gives an overview of the used channels and their parameters.

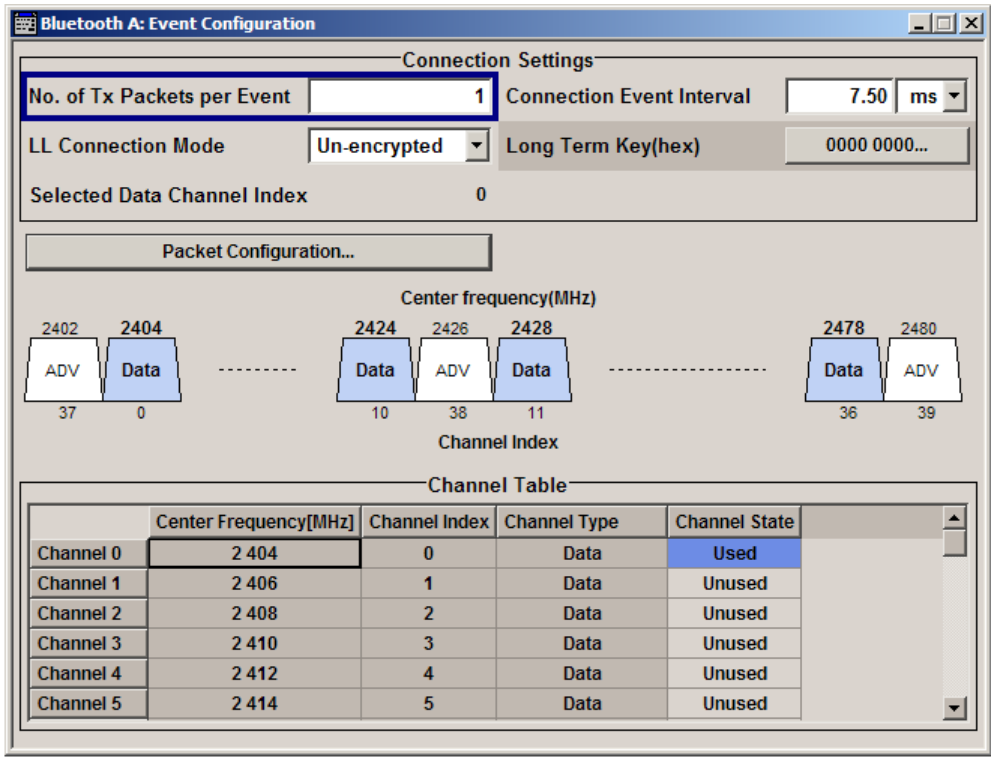



*Figure 3-1: Event Configuration Dialog of channel type Data*

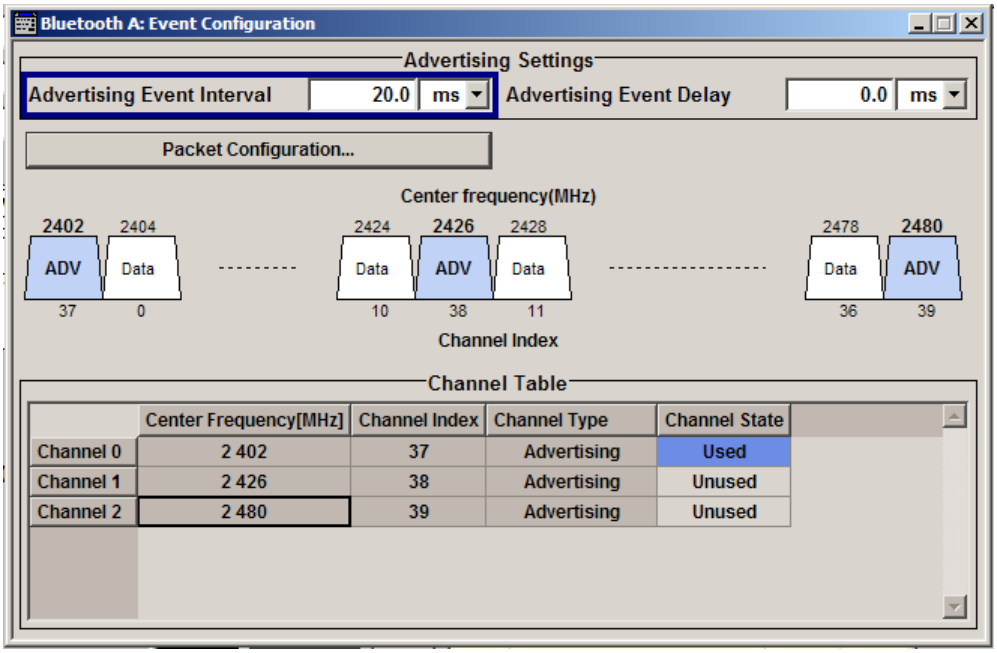



*Figure 3-2: Event Configuration Dialog of channel type Advertising*

## 3.5.1 Advertising Event / Frame Configuration Settings

**Advertising Event Interval**
Sets the time interval between two consecutive advertising events, with regard to the starting points.

**Note:** This parameter is relevant for advertising event configuration and for the packet types ADV-IND, ADV_DIRECT_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:AEINterval` on page 111
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ADINterval` on page 111

**Advertising Event Delay**
Sets a time delay between the start times of two consecutive advertising events. The value is added to the advertising event interval.

**Note:** This parameter is relevant for advertising event configuration and for the packet types ADV-IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:AEDelay` on page 111

**Scan Window**
Sets the length of the window during which the scanner is operating in the advertising channel.

Note that the scan window is less or equal to the value of the scan interval.

**Note:** This parameter is relevant for advertising frame configuration and for the packet type SCAN_REQ.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SWINdow` on page 126

**Scan Interval**
Sets the time interval between the starting points of two consecutive windows during which the scanner is operating in an advertising channel.

**Note:** This parameter is relevant for advertising frame configuration and for the packet type SCAN_REQ.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SINTerval` on page 126

**Advertising Packet Interval**
Sets the time interval between packets starting points of two consecutive packets in the advertising channel.

**Note:** This parameter is relevant for advertising frame configuration and for the packet type SCAN_RSP.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:APINterval` on page 112

**Transmit Window Offset**
Displays the start point of the transmit window.

**Note:** This parameter is relevant for advertising frame configuration and for the packet type CONNECT_REQ.
This parameter is set in the Packet Configuration, see "Transmit Window Offset" on page 45.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WOINfo?` on page 127

**Transmit Window Size**
Indicates the size of the transmit window, regarding to the start point.

Note that the scan window size is less or equal to the value of the connection interval.

**Note:** This parameter is relevant for advertising frame configuration and for the packet type CONNECT_REQ.
The parameter is set in the "Packet Configuration" dialog, see "Transmit Window Size" on page 44.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WSINfo?` on page 127

**Packet Configuration**
Opens the dialog for setting the corresponding packet configuration.

This dialog is described in Chapter 3.6.1, "Packet Configuration Settings", on page 38.

Remote command:
n.a.

### 3.5.2 Data Event Connection Settings

**No. of Tx Packets per Event**
Sets the number of Tx packets per event. Each connection contains at least one data channel packet. The maximum number of packets per event is determined by the duration of the connection event interval.

**Note:** This parameter is relevant for data event connection settings.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PNUMber` on page 126

**Connection Event Interval**
Set the time interval between the start points of two consecutive connection events. Subsequent transmissions within an event are separated by this parameter in order to separate connecting event starting points in time.

**Note:** This parameter is relevant for data event connection settings and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINTerval`
on page 115
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NCINterval`
on page 120
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINTerval`
on page 115

**LL Connection Mode**
Select the link layer connection mode. In order to provide safe transmission of payload data, the data in the packet can be encrypted. If activated, the payload data follows MIC (Message authentication Code).

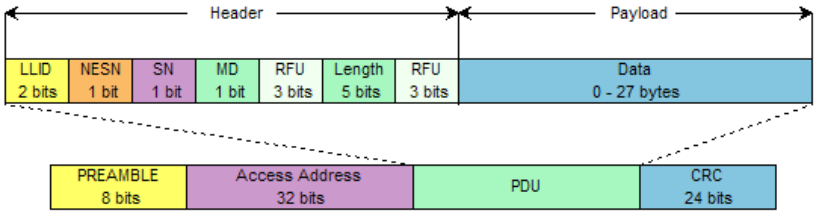
**Note:** This parameter is relevant for data event connection settings.

The following table shows which types of packets can be encrypted and / or unencrypted.

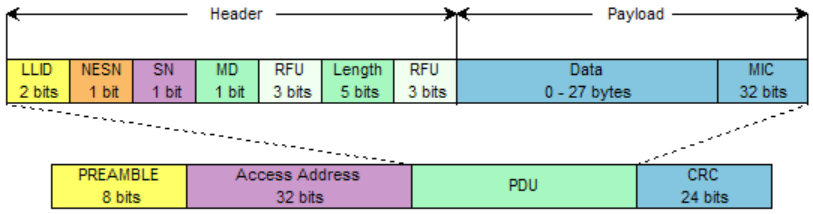
| Packet Type | encrypted | unencrypted |
|---|---|---|
| DATA | X | X |
| CONNECTION_UPDATE_REQ | X | X |
| CHANNEL_MAP_REQ | X | X |
| LL_TERMINATE_IND | X | X |
| LL_ENC_REQ | - | X |
| LL_ENC_RSP | - | X |

| Packet Type | encrypted | unencrypted |
|---|---|---|
| LL_START_ENC_REQ | - | X |
| LL_START_ENC_RSP | X | - |
| LL_FEATURE_REQ | X | X |
| LL_FEATURE_RSP | X | X |
| LL_PAUSE_ENC_REQ | - | X |
| LL_PAUSE_ENC_RSP | X | X |
| LL_VERSION_IND | X | X |
| LL_REJECT_IND | X | X |
| UNKNOWN_RSP | X | X |

"Unencrypted" Payload data is transmitted without encoding.




"Encrypted" The link layer connection runs in encrypted mode.




Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LCMode` on page 112

**Long Term key (hex)**
Indicates the time the controller needs to receive the long term key from the host. After this time, the controller is ready to enter into the last phase of encryption mode setup.

**Note:** This parameter is relevant for data event connection settings. In encrypted mode, the code can be edited.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LTKey` on page 112

**Selected Data Channel Index**
Indicates the number of the first active data channel.

The data channel is selected for each connection event. The master and slave determine the used data channel by selecting from the list of used channels (see "Channel Table" on page 37).

**Note:** This parameter is relevant for data event connection settings.

Displays the data channel index currently selected.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SDCI? on page 123

### 3.5.3 Channel Table Settings

The channel table displays all parameters characterizing the channel and the current state.

**Channel Table**
The channel table displays all parameters characterizing the channel and the current state.

Every channel is represented with bit positioned as per the data channel index. LSB represents data channel index 0 and the bit in position 36 represents data channel index 36.

If the channel is used channel its bit is to be set to '1'. Bit value '0' indicates that the channel is unused.

The bits in positions 37, 38 and 39 shall be set to zero upon transmission and ignored upon receipt.

"Center Frequency"
Indicates the center frequency of a channel.

"Channel Index"
Indicates the channel index.

"Channel Type"
Indicates the channel type.

"Channel State"
Indicates used and unused data channels.

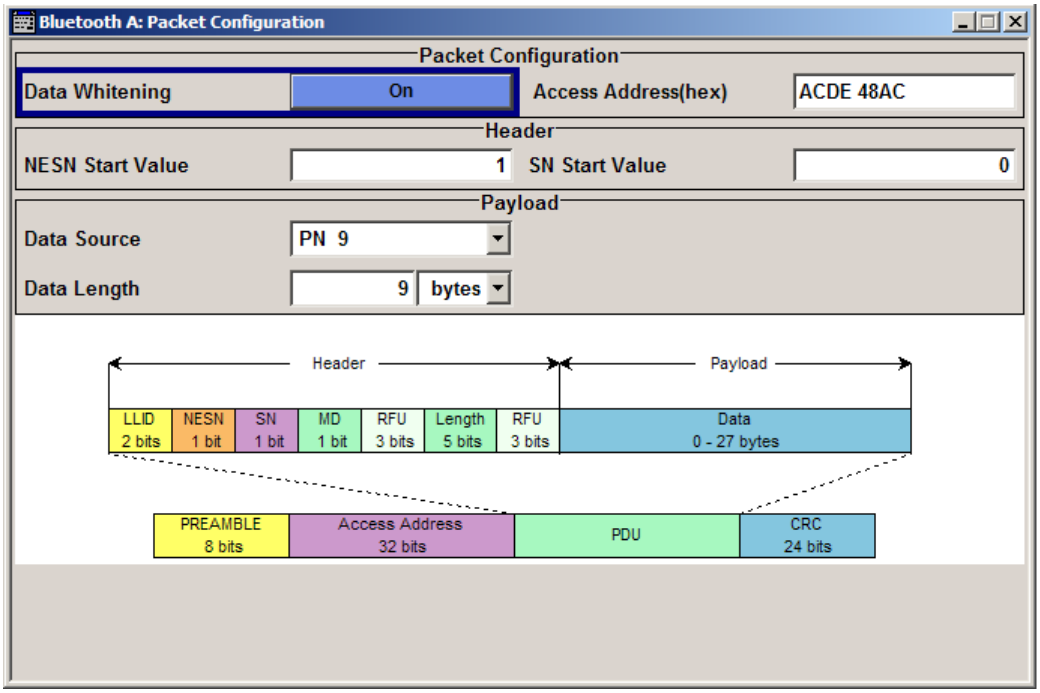
Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ACTable:CHANnel<ch0>:STATe on page 110
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:DCTable:CHANnel<ch0>:STATe on page 110
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DCMTable:CHANnel<ch0>:STATe on page 110

Remote command:
Entire data set:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ACTable on page 109
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:DCTable on page 109
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DCMTable on page 109

# 3.6 Packet Configuration - Bluetooth LE

The "Packet Configuration" dialog is reached via the "Event Configuration" dialog or the "Frame Configuration" dialog, respectively.

The available settings in the dialog depend on the selected channel type and the packet type. Header configurations are selected in the "Header" section. Data and control parameters are set in the "Payload" section. The figure shows the packet structure of the currently selected configuration.




## 3.6.1 Packet Configuration Settings

**Data Whitening**
Activates or deactivates the Data Whitening.

Evenly distributed white noise is ideal for the transmission and real data can be forced to look similar to white noise with different methods called Data Whitening. Applied to the PDU and CRC fields of all packet types, whitening is used to avoid long equal sequences in the data bit stream.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DWHitening`
on page 117

**Access Address**
Sets the access address of the link layer connection.

Bluetooth LE transmissions are based on an interface packet format, that consists of a preamble (8 bits), the access address (32 bits), the PDU and CRC (24 bits).

The access address´ structure depends on the packet type:

- Data channel packets
  The access address is a pseudo-random LL connection address, generated by the initiator of the LL connection. The address has to follow some specific rules, which are described in the "Bluetooth Low Energy Technology Specification".
- Advertising channel packets
  The address is fixed to 01101011011111011001000101110001 with the left most bit sent first and being the LSB.

**Note:** This parameter is relevant for all available package types specified in events in the data channel, and frames in the advertiser channel, i.e. DATA, CONNECTION_UPDATE_REQ, CHANNEL_MAP_REQ, TERMINATE_IND, LLENC_REQ, LL_FEATURE_REQ, LL_PAUSE_ENC_REQ, LL_START_ENC_:RSP, FEATURE_REQ, and CONNECT_REQ.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:AADDress`
on page 113

**NESN Start Value**
Sets the start value of the next expected packet from the same device in the LL connection (**N**ext**E**xpected **S**equence**N**umber). This parameter can be set in the first event. From the second event this field is not indicated.

**Note:** This parameter is relevant for data event configuration and all data channel packet types except TEST_PACKET.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSValue`
on page 121

**SN Start Value**
Sets the sequence number of the packet. This parameter can be set in the first event. From the second event this field is not indicated.

**Note:** This parameter is relevant for data event configuration and all data channel packet types except TEST_PACKET.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SSValue`
on page 124

**Device´s Addr Type**
Selects the address type of the controller device. Depending on the Bluetooth controller role either the Tx ot Rx or both address types are assigned.

Subdivided into private and random, a Bluetooth LE device address consists of 48 bits. The format of the device address differs depending on the selected address type.

**Note:** This parameter is relevant for advertising event or frame configuration.

The bluetooth "Controller Role" and the packet type determine the available entries:

- **Tx** in conjunction with the packet types ADV_IND, ADV_DIRECT_IND, ADV_NONCONN_IND, ADV_DISCOVER_IND, SCAN_REQ, SCAN_RSP and CONNECT_REQ
- **Rx** for the packet types ADV_DIRECT_IND, SCAN_REQ and CONNECT_REQ

"Public" Allocates a unique 48 bit address to each bluetooth LE device. The public address is given from the registration authority IEEE.

"Random" Allocates a 48 bit address to each bluetooth LE device. A random address is optional.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:TATYpe` on page 122

**Data Source**
Selects the data source used for the payload.

**Note:** This parameter is relevant for event configuration and packet types DATA, ADV_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

The following standard data sources are available:

- "All 0, All 1"
  An internally generated sequence containing 0 data or 1 data.
- "PNxx"
  An internally generated pseudo-random noise sequence.
- "Pattern"
  An internally generated sequence according to a bit pattern.
  Use the "Pattern" box to define the bit pattern.
- "Data List/Select DList"
  A binary data from a data list, internally or externally generated.
  Select "Select DList" to access the standard "Select List" dialog.
  – Select the "Select Data List > navigate to the list file *.dm_iqd > Select" to select an existing data list.
  – Use the "New" and "Edit" functions to create internally new data list or to edit an existing one.
  – Use the standard "File Manager" function to transfer external data lists to the instrument.

  See also "Main Dialog > Data List Management".

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA` on page 115
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA:DPATtern` on page 116
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA:DSELection` on page 116

**Data Length**
Enters the payload data length in bytes.

**Note:** This parameter is relevant for event configuration with packet types ADV_IND, ADV_NONCONN_IND and ADV_DISCOVER_IND.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DLENgth
on page 117

**Connection Event Interval**
Set the time interval between the start points of two consecutive connection events. Subsequent transmissions within an event are separated by this parameter in order to separate connecting event starting points in time.

**Note:** This parameter is relevant for data event connection settings and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINTerval
on page 115
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NCINterval
on page 120
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINTerval
on page 115

**Slave Latency**
Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

**Note:** This parameter is relevant for data event and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SLATency
on page 123
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSLatency
on page 121

**LL Connection Timeout**
Defines the maximum time between two correctly received Bluetooth LE packets in the LL connection before the connection is considered lost.

**Note:** This parameter is relevant for data event and advertising frame configuration with the packet types CONNECTION_UPDATE_REQ and CONNECT_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:LCTimeout
on page 119
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NLCTimeout
on page 120

**Connection Instant**
Sets a connection instant for indicating the connection event at which the new connection parameters are taken in use.

Both the master and the slave have a 32-bit connection event counter per LL connection. It is reset to zero on the first connection event of the LL connection and incremented by one on every elapsed connection event interval of the LL connection.

**Note:** This parameter is relevant for data event configuration with the packet types CONNECTION_UPDATE_REQ and CHANNEL_MAP_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINStant
on page 114

**Show / Hide Data Channel (Mapping) Table**
In data event and advertising frame configuration with the packet types CHANNEL_MAP_REQ and CONNECT_REQ, calls / hides the channel map table that displays the used channels and their parameters.

The channel table is described in Chapter 3.5.3, "Channel Table Settings", on page 37.

Remote command:
n.a.

**Hop Length**
Sets the difference from the current channel to the next channel. The master and slave devices determine the data channel in use for every connection event from the channel map. Hop_length is set for the LL connection and communicated in the CONNECT_REQ packets.

**Note:** This parameter is relevant for data event and advertising frame configuration with the packet type CONNECT_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:HLENgth
on page 118

**Random Vector (hex)**
Sets the random vector of the master for device identification.

The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

**Note:** This parameter is relevant for data event configuration with the packet type LLENC_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:RVECtor
on page 123

**Encrypted DIVersifier (hex)**
Sets the encrypted diversifier of the master for device identification. The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

**Note:** This parameter is relevant for data event configuration with the packet type LLENC_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:EDIVersifier on page 118

**Session Key iD (hex)**
Sets the master's or the slave's portion of the session key diversifier (SKDm/SKDs).

**Note:** This parameter is relevant for data event configuration with the packet types LLENC_REQ (Master) and LL_ENC_RSP (Slave).

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:MSKD on page 119

**Initialization Vector (hex)**
Sets the master's or the slave's portion of the initialization vector(IVm/IVs).

**Note:** This parameter is relevant for data event configuration with the packet types LLENC_REQ (Master) and LL_ENC_RSP (Slave).

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:MIVector on page 119

**Feature Set Length**
Enables that the feature set length is indicated.

FeatureSet indicates whether the Controller features are used or not. All the data in FeatureSet is RFU(zero).

**Note:** This parameter is relevant for data event configuration with the packet types FEATURE_REQ (Master) and FEATURE_RSP (Slave).

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:FSLength on page 118

**Unknown Type (hex)**
Enables that an invalid control packet is indicated.

The CtrType field indicates the value of the LL control packet that caused the transmission of this packet.

This parameter is relevant for data event configuration with the packet type UNKNOWN_RSP (Slave).

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:UTYPe on page 124

**Controller´s Device Addr**
Sets the advertiser´s device address.

In Bluetooth LE systems all the transmissions start with an 8 bit preamble followed by an access address. The access address is composed of a the parts "Company_Id" (LSB) and the "Company_assigned" (MSB). Beside the address fields the notation is given.

For advertising channel packets the format of the device address differs, depending on the selected address type.

**Note:** This parameter is relevant for advertising event or frame configuration. Refer to Tx/Rx Addres Type - Bluetooth LEDevice´s Addr Type for information about the available package types for the respective "Controller Roles".

- "Public Address Types"
  The public address is given from the registration authority IEEE and is composed of:
  – LSB: 24 bits = company_assigned
  – MSB: 24 bits = company_id
- "Private Address Type"
  A private address is optional and composed of:
  – LSB: 24 bits = hash
  – MSB: 24 bits = random

Remote command:
Company_Assigned and Company_Id in Advertiser´s Device Address
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ACID`
on page 113
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ACASsigned`
on page 113
Company_Assigned and Company_Id in Scanner's Device Address
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCASsigned`
on page 113
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCID`
on page 113
Company_Assigned and Company_Id in Initiator's Device Address
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ICASsigned`
on page 113
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ICID`
on page 113

**CRC Initial**
Sets the initialization value for the CRC (Cyclic Redundary Check, 24 bits) calculation. A packet has been received correctly, when it has passed the CRC check.

**Note:** This parameter is relevant for advertising frame configuration and the packet type CONNECT_REQ.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CIValue`
on page 115

**Transmit Window Size**
Sets the size of the transmit window, regarding to the start point.

Note that the scan window size is less or equal to the value of the connection interval, see "Connection Event Interval" on page 35.

**Note:** This parameter is relevant for advertising frame configuration and for the packet types CONNECT_REQ and CONNECTION_UPDATE_REQ.
This parameter is also indicated in the Frame Configuration Dialog.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:WSIZe
on page 125
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NWSize
on page 122

**Transmit Window Offset**
Sets the start point of the transmit window.

**Note:** This parameter is relevant for advertising frame configuration and for the packet types CONNECT_REQ and CONNECTION_UPDATE_REQ.
This parameter is also indicated in the Frame Configuration Dialog.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:WOFFset
on page 125
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NWOFfset
on page 121

**Sleep Clock Accuracy**
Defines the master´s clock accuracy with specified encoding. This parameter is used by the slave to determine required listening windows in the LL connection. It is a controller design parameter known by the Controller.

**Note:** This parameter is relevant for advertising frame configuration and the packet type CONNECT_REQ.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCACcuracy
on page 123

**Error Code**
Sets the error code value to inform the remote device why the connection is about to be terminated in case of LL_TERMINATE_IND packet. On the other hand, this parameter for LL_REJECT_IND packet is used for the reason a request was rejected. A 8 bit value is set.

**Note:** This parameter is relevant for data frame configuration and the packet type LL_TERMINATE_IND and LL_REJECT_IND.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ECODe
on page 117

**Company ID**
Sets the company identifier of the manufacturer of the Bluetooth Controller. A 16 bit value is set.

**Note:** This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CID
on page 114

**Version Number**
Sets the version of the Bluetooth Controller specification (8 bits).

**Note:** This parameter is relevant for data frame configuration and the packet type LL_VERSION_IND

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:VNUMber
on page 125

**Sub Version Number**
Sets a unique value for each implementation or revision of an implementation of the Bluetooth Controller.

A 16 bit value is set.

**Note:** This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

Remote command:
[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SVNumber
on page 124

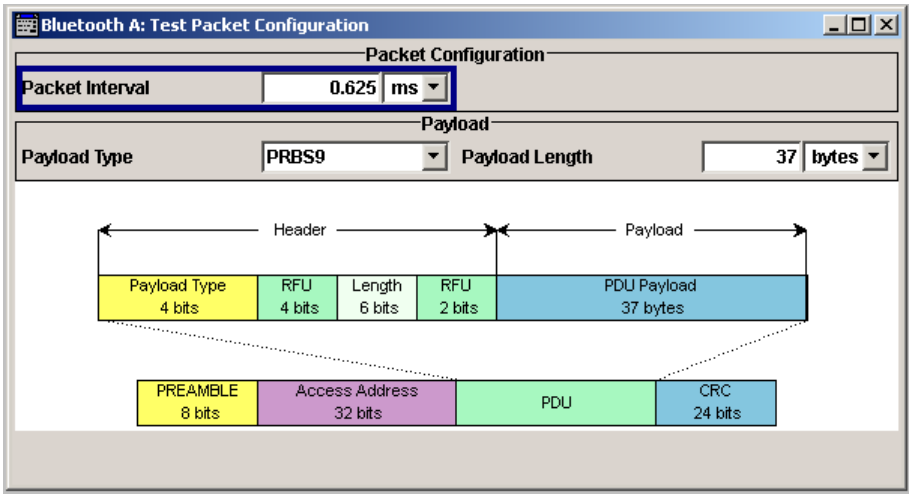
**Graph**
The figure in the packet configuration dialog shows the packet structure of the currently selected packet type.

Remote command:
n.a.

### 3.6.2 Test Packet Configuration Settings

The "Test Packet Configuration" dialog is reached via the "Bluetooth" main dialog, or alternatively the "Dirty Transmitter" test dialog.

The dialog contains the settings, necessary to configure the test packet and graphically shows the distribution of the packets.




**Packet Interval**
Sets the time interval between two consecutive test packets, with regard to the starting points.

Test packet Interval

**Note:** This parameter is relevant for test packet types only.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:TPINterval`
on page 108

**Payload Type**
Selects the data source used for the payload test packets.

**Note:** This parameter is relevant for test packet types only.

"PRBS 9, 15" Select a PRBS-modulated data sequence (PRBS = pseudo random binary sequence) for testing.

"Pattern 1, 2, 3, 4, 5, 6"
Pattern is predefined.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:UPSource`
on page 109

**Payload Length**
Sets the payload length.

**Note:** This parameter is relevant for test packet types only.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:UPLength
on page 109

## 3.7 Dirty Transmitter Test

The Dirty Transmitter settings contain parameters which can be changed for the master signal in order to test the connection under 'dirty transmitter' conditions and define the influence on the receiver quality (bit error rate tests).

Dirty transmitter parameters according to the Bluetooth test specification (Basic Rate) are given in the table below.

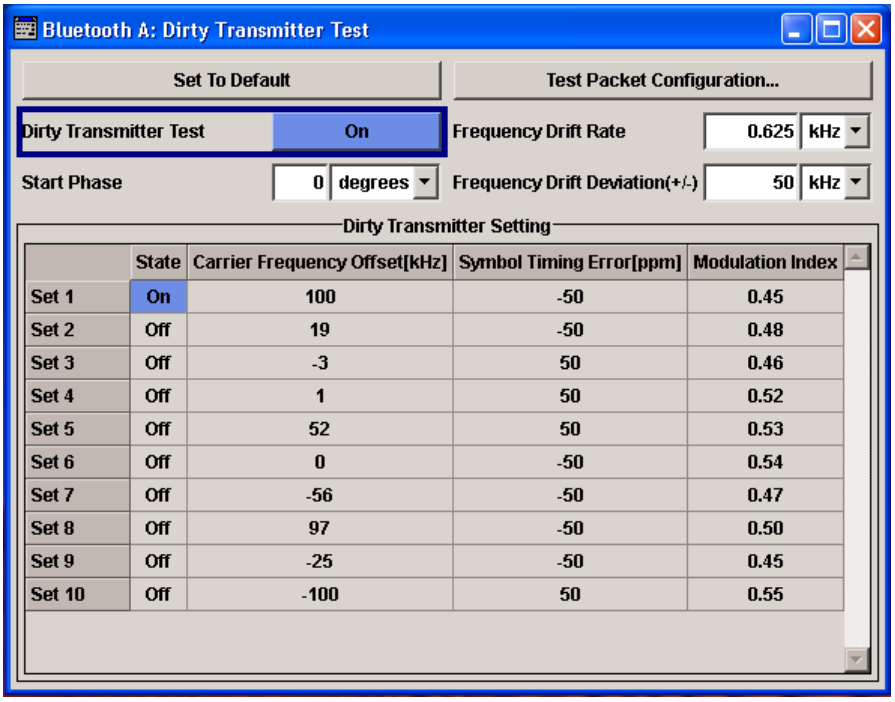
***Table 3-3: Dirty transmitter parameters according to the Bluetooth test specification (Basic Rate)***

| Set | Carrier Frequency Offset | Symbol Timing Error | Modulation Index |
|---|---|---|---|
| 1 | 75 | -20 | 0.28 |
| 2 | 14 | -20 | 0.30 |
| 3 | -2 | +20 | 0.29 |
| 4 | 1 | +20 | 0.32 |
| 5 | 39 | +20 | 0.33 |
| 6 | 0 | -20 | 0.34 |
| 7 | -42 | -20 | 0.29 |
| 8 | 74 | -20 | 0.31 |
| 9 | -19 | -20 | 0.28 |
| 10 | -75 | +20 | 0.35 |

Dirty transmitter parameters according to the Bluetooth test specification (EDR) are given in the table below.

***Table 3-4: Dirty transmitter parameters according to the Bluetooth test specification (EDR)***

| Set | Carrier Frequency Offset | Symbol Timing Error |
|---|---|---|
| 1 | 0 | 0 |
| 2 | +65 | -20 |
| 3 | -65 | +20 |




Provided are the following settings:

**Set to Default**
Calls the default settings for the Dirty Transmitter Test.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:DTTest:STDefault` on page 103

**Test Packet Configuration**
Opens the dialog for setting a test packet configuration, see Chapter 3.6.2, "Test Packet Configuration Settings", on page 46.

**Dirty Transmitter Test**
(Available only for packet types DH1, DH3, DH5, 2-DH1, 2-DH3, 2 - DH5, 3-DH1, 3-DH3, 3-DH5, 2-EV3, 2-EV5, 3-EV3, 3-EV5.)

Activates or deactivates the Dirty Transmitter Test.

For Basic Rate packets, each set of parameters in the "Dirty Transmitter Setting" table below is used for a duration of 20 ms. After 20 ms, the following set is used, continuing with the first set after the sequence is completed.

For EDR packets, the parameter sets apply for 20 packets each.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:DTTest:DTTState` on page 101

**Start Phase**
Enters a start phase.

The start phase of the sine wave used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset is set here.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:SPHase on page 102

**Frequency Drift Rate**
Enters a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth siganal around center frequency + carrier frequency offset with the set frequency drift rate.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:FDRate on page 102

**Frequency Drift Deviation (+/-)**
Enters a frequency drift deviation.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset. The maximum deviation reached during the drift equals the set frequency drift deviation.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:FDDeviation on page 101

**Dirty Transmitter Setting**
Indicates the dirty transmitter parameters according to the Bluetooth test specification (Basic Rate).

**State ← Dirty Transmitter Setting**
Activates or deactivates the corresponding parameter set.

If deactivated, the parameters are skipped in the sequence, and the next active set is used.

For Basic Rate packets, each set applies to 20ms of signal. For EDR packets, each set applies to 20 packets.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:STATe
on page 104
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:STATe
on page 105

**Carrier Frequency Offset kHz ← Dirty Transmitter Setting**
Determines a carrier frequency offset.

The center frequency of the modulated RF carrier is offset by the specified value.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:CFOFfset
on page 103
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:CFOFfset
on page 105

**Symbol Timing Error ← Dirty Transmitter Setting**
Sets the symbol timing error in ppm.

The symbol timing error modifies the symbol clock frequency by the set value

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:STERror
on page 104
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:STERror
on page 106

**Modulation Index ← Dirty Transmitter Setting**
(Only for Basic Rate Packets)

Sets the modulation index.

The modulation index **h** specifies the frequency deviation, defined as:

$$h = \frac{2\Delta f}{f_{symbol}}$$

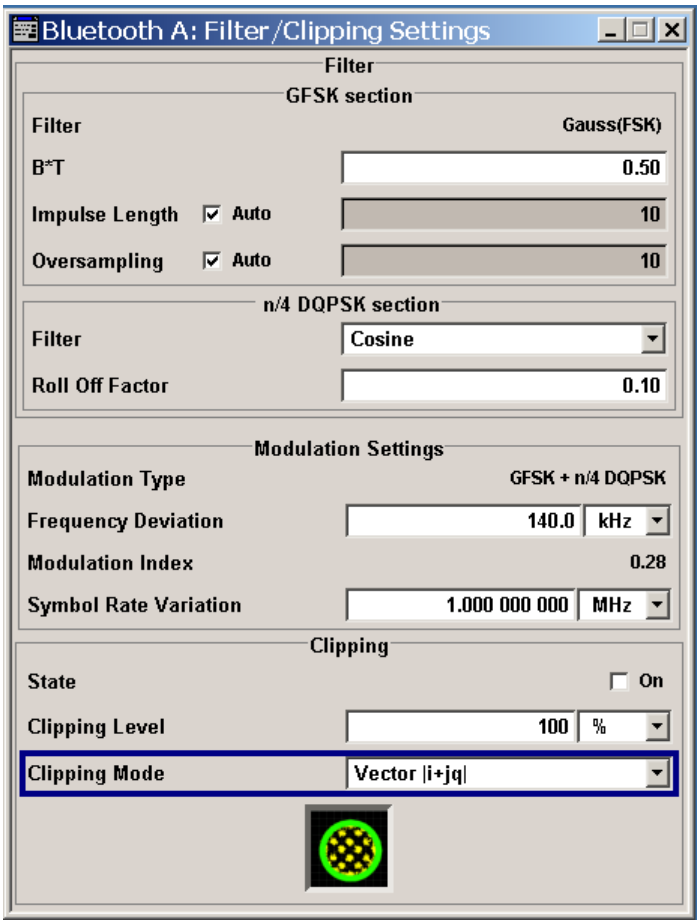
where $f_{symbol}$ is the "symbol rate" and $\Delta f$ is the "frequency deviation".

According to the Bluetooth standard, the modulation index may vary between 0.28 and 0.35.

Remote command:
[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:MINDex
on page 103

## 3.8 Filter/Clipping Settings

► To access the filter / clipping settings, select "Main dialog > Filter/Clipping".




The dialog comprises the settings, necessary to configure the baseband filter, the modulation settings and to enable clipping.

### 3.8.1 Filter Settings

Provided are the following settings for configuring the baseband filter:

**Filter**
Indicates the filter used for GFSK part.

Remote command:
n.a.

**Roll Off Factor / B xT**
Sets the filter parameter.

Sets the filter parameter.

The filter parameter offered ("Roll Off Factor" or "BxT") depends on the currently selected filter type. This parameter is preset to the default for each of the predefined filters.

Remote command:
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:APCO25 on page 76
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:COSine on page 76
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:FGAuss on page 77
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:GAUSs on page 77
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:PGAuss on page 78
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:RCOSine on page 78
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:SPHase on page 78

**Cut Off Frequency Factor**
(available for filter parameter Lowpass only)

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

Remote command:
[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:LPASs on page 77

**Filter (π/4 DQPSK section)**
Selects the filter used for DQPSK/8DPSK sections with EDR packets.

Remote command:
[:SOURce<hw>]:BB:BTOoth:FILTer:TYPE on page 74

## 3.8.2 Modulation Settings

Provided are the following settings for configuring the modulation:

**Modulation type**
Displays the modulation type used for the current packet selection.

Remote command:
[:SOURce<hw>]:BB:BTOoth:FILTer:MTYPe on page 76

**Frequency deviation**
Enter the frequency deviation of the frequency modulated part.

The frequency deviation can be varied in a range from 100.0 kHz to 200.0 kHz according to Bluetooth specification.

Remote command:
[:SOURce<hw>]:BB:BTOoth:MSETtings:FDEViation on page 76

**Modulation index**
Displays the modulation index resulting from the entered frequency deviation value.

Modulation index is calculated from the given frequency deviation and symbol rate values.

The modulation index **h** is defined as:

$$h = \frac{2\Delta f}{f_{symbol}}$$

where $f_{symbol}$ is the "symbol rate" and $\Delta f$ is the "frequency deviation".

According to the Bluetooth standard, the modulation index is allowed to vary between 0.28 and 0.35.

Remote command:
[:SOURce<hw>]:BB:BTOoth:FILTer:MINDex on page 75

**Symbol Rate Variation**
Enter the symbol rate.

Remote command:
[:SOURce<hw>]:BB:BTOoth:SRATe:VARiation on page 79

### 3.8.3 Clipping Settings

Provided are the following settings for configuring clipping:

**Clipping State**
Switches baseband clipping on and off.

Baseband clipping is a very simple and effective way of reducing the crest factor of the signal. Since clipping is done prior to filtering, the procedure does not influence the spectrum. The EVM however increases.

Remote command:
[:SOURce<hw>]:BB:BTOoth:CLIPping:STATe on page 74

**Clipping Level**
Sets the limit for clipping.

This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

Remote command:
[:SOURce<hw>]:BB:BTOoth:CLIPping:LEVel on page 73

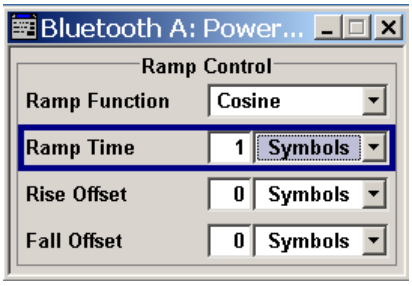
**Clipping Mode**
Selects the clipping method. A graphic illustration of the way in which these two methods work is given in the dialog.

- "Vector | i + jq |"
  The limit is related to the amplitude | i + q |. The I and Q components are mapped together, the angle is retained.
- "Scalar | i | , | q |"
  The limit is related to the absolute maximum of all the I and Q values | i | + | q |. The I and Q components are mapped separately, the angle changes.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:CLIPping:MODE` on page 73

## 3.9 Power Ramping Settings

► To access this dialog, select "Main Menu > Power Ramping".




The dialog comprises the settings, necessary to configure power ramping.

**Ramp Function**
Selects the form of the transmitted power, i.e. the shape of the rising and falling edges during power ramp control.

"Linear" The transmitted power rises and falls with linear fashion.

"Cosine" The transmitted power rises and falls with a cosine-shaped edge. This gives rise to a more favorable spectrum than the Linear setting.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PRAMping:RFUNction` on page 92

**Ramp Time**
Sets the power ramping rise time and fall time for a burst.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:PRAMping:RTIMe` on page 92

**Rise Offset**
Sets the offset in the rising edge of the envelope at the start of a burst. A positive value moves the ramp into the beginning of a transmitted packet and a negative value introduces an additional guard period before the start of the packet.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PRAMping:ROFFset on page 92

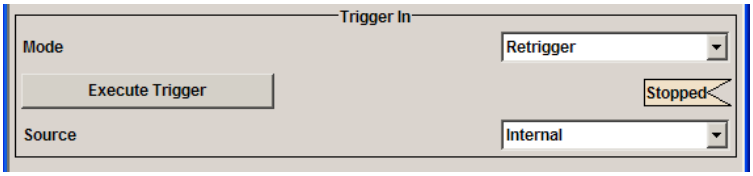
**Fall Offset**
Sets the offset of the falling edge of the envelope at the end of a burst. A positive value introduces a guard period after the end of the packet and a negative value moves the ramp into the end part of the transmitted packet.

Remote command:
[:SOURce<hw>]:BB:BTOoth:PRAMping:FOFFset on page 91

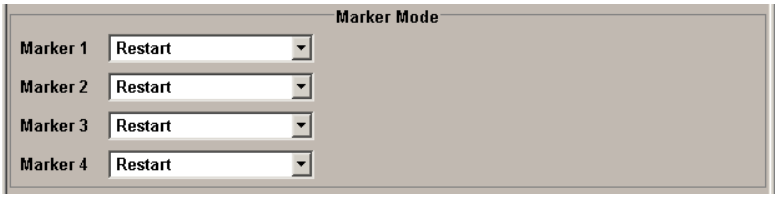
## 3.10 Trigger/Marker/Clock Settings

To access this dialog, select "Main Menu > Trigger/Marker".

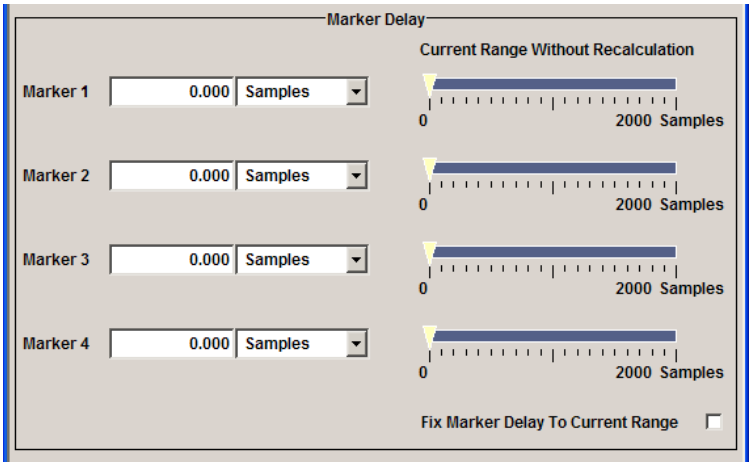
The "Trigger In" section is where the trigger for the signal is set. Various parameters will be provided for the settings, depending on which trigger source - internal or external - is selected. The current status of signal generation ("Running" or "Stopped") is indicated for all trigger modes.



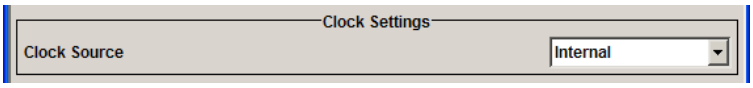

The "Marker Mode" section is where the marker signals at the MARKER output connectors are configured.




The "Marker Delay" section is where a marker signal delay can be defined, either without restriction or restricted to the dynamic section, i.e., the section in which it is possible to make settings without restarting signal and marker generation.




The "Clock Settings" section is where the clock source is selected and - in the case of an external source - the clock type.




The buttons in the last section lead to submenu for general trigger, clock and mapping settings.


### 3.10.1 Trigger Settings

The "Trigger In" section is where the trigger for the signal is set. Various parameters will be provided for the settings, depending on which trigger source - internal or external - is selected. The current status of signal generation ("Running" or "Stopped") is indicated for all trigger modes.

**Trigger Mode**
Selects trigger mode, i.e. determines the effect of a trigger event on the signal generation.

- "Auto"
  The signal is generated continuously.
- "Retrigger"
  The signal is generated continuously. A trigger event (internal or external) causes a restart.
- "Armed_Auto"
  The signal is generated only when a trigger event occurs. Then the signal is generated continuously.
  An "Arm" stops the signal generation. A subsequent trigger event (internal with or external) causes a restart.
- "Armed_Retrigger"

The signal is generated only when a trigger event occurs. Then the signal is generated continuously. Every subsequent trigger event causes a restart.
An "Arm" stops signal generation. A subsequent trigger event (internal with or external) causes a restart.

- "Single"
  The signal is generated only when a trigger event occurs. Then the signal is generated once to the length specified at "Signal Duration".
  Every subsequent trigger event (internal or external) causes a restart.

Remote command:
`[:SOURce<hw>]:BB:BTOoth[:TRIGger]:SEQuence` on page 84

**Signal Duration Unit**
Selects the unit for the entry of the length of the signal sequence to be output in the Single trigger mode. Available units are sequence length (SL) or frames.

| | |
|---|---|
| "Sequence Length" | The selected unit for the entry of the length of the signal sequence at the output in the Single trigger mode is sequence length . |
| "Frames" | The selected unit for the entry of the length of the signal sequence at the output in the Single trigger mode is frame. |

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger:SLUNit` on page 82

**Signal Duration**
Enters the length of the signal sequence to be output in the "Single" trigger mode.

Use this parameter to deliberately output part of the signal, an exact sequence of the signal, or a defined number of repetitions of the signal.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger:SLENgth` on page 82

**Running/Stopped**
For enabled modulation, displays the status of signal generation for all trigger modes.

- "Running"
  The signal is generated; a trigger was (internally or externally) initiated in triggered mode.
- "Stopped"
  The signal is not generated and the instrument waits for a trigger event.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger:RMODe` on page 81

**Arm**
Stops signal generation manually. This button appears only with "Running" signal generation in the "Armed_Auto" and "Armed_Retrigger" trigger modes.

Signal generation can be restarted by a new trigger (internally with "Execute Trigger" or externally).

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:ARM:EXECute on page 79

**Execute Trigger**
Executes a trigger manually. A manual trigger can be executed only if an internal trigger source and a trigger mode other than "Auto" have been selected.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:EXECute on page 80

**Trigger Source**
Selects trigger source. This setting is effective when a trigger mode other than "Auto" has been selected.

- "Internal"
  The trigger event is executed by "Execute Trigger".
- "Internal (Baseband A/B)"
  (two-path instruments)
  The trigger event is the trigger signal from the second path
- "External (Trigger 1/2)"
  The trigger event is the active edge of an external trigger signal, supplied at the TRIGGER 1/2 connector.
  Use the "Global Trigger/Clock Settings" dialog to define the polarity, the trigger threshold and the input impedance of the trigger signal.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:SOURce on page 83

**Sync. Output to External Trigger**
(enabled for Trigger Source External)

Enables/disables output of the signal synchronous to the external trigger event.

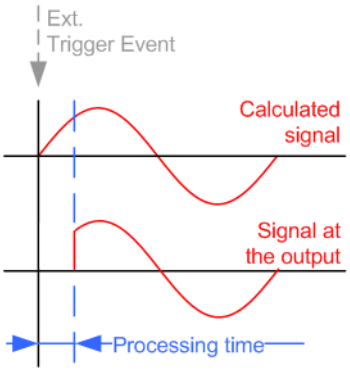
For R&S SMBV instruments:

For or two or more R&S SMBVs configured to work in a master-slave mode for synchronous signal generation, configure this parameter depending on the provided system trigger event and the properties of the output signal. See the table below for an overview of the required settings.

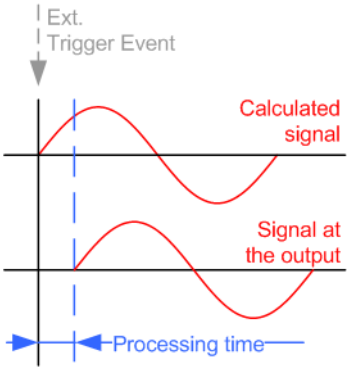
***Table 3-5: Typical Applications***

| System Trigger | Application | "Sync. Output to External Trigger" |
|---|---|---|
| Common External Trigger event for the master and the slave instruments | All instruments are synchronous to the external trigger event | ON |
| | All instruments are synchronous among themselves but starting the signal from first symbol is more important than synchronicity with external trigger event | OFF |
| Internal trigger signal of the master R&S SMBV for the slave instruments | All instruments are synchronous among themselves | OFF |

"On" Corresponds to the default state of this parameter.
The signal calculation starts simultaneously with the external trigger event but because of the instrument's processing time the first samples are cut off and no signal is outputted. After elapsing of the internal processing time, the output signal is synchronous to the trigger event.




"Off" The signal output begins after elapsing of the processing time and starts with sample 0, i.e. the complete signal is outputted.
This mode is recommended for triggering of short signal sequences with signal duration comparable with the processing time of the instrument.




Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal]:SYNChronize:OUTPut` on page 80

**Trigger Delay**
Delays the trigger event of the signal from:
- the external trigger source
- the other path

Use this setting to:
- synchronize the instrument with the device under test (DUT) or other external devices

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal<ch>]:DELay` on page 83
`[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:DELay` on page 80

**Trigger Inhibit**
Sets the duration for inhibiting a new trigger event subsequent to triggering. The input is to be expressed in samples.

In the "Retrigger" mode, every trigger signal causes signal generation to restart. This restart is inhibited for the specified number of samples.

This parameter is only available on external triggering or on internal triggering via the second path.

For two-path instruments, the trigger inhibit can be set separately for each of the two paths.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal<ch>]:INHibit` on page 83
`[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:INHibit` on page 81

## 3.10.2 Marker Mode

The marker output signal for synchronizing external instruments is configured in the marker settings section "Marker Mode".

The R&S SMBV supports only two markers.

**Marker Mode**
Selects a marker signal for the associated MARKER output.

| | |
|---|---|
| "Restart" | A marker signal is generated at the start of each signal sequence. |
| "Frame Start" | A marker signal is generated at the start of each frame. |
| "Frame Active Part" | The marker masks the active part of the frame. At the start of each burst, the marker signal changes to high. It changes back to low after the end of each burst. |
| "Pulse" | A regular marker signal is generated. The pulse frequency is defined by entering a divider. The frequency is derived by dividing the sample rate by the divider. The input box for the divider opens when "Pulse" is selected, and the resulting pulse frequency is displayed below it. |

Remote command:
`[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PULSe:DIVider`
on page 88
`[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PULSe:FREQuency?`
on page 88

"Pattern " A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 32 bits and is defined in an input field which opens when pattern is selected.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PATTern on page 88

"ON/OFF Period" A regular marker signal that is defined by an ON/OFF ratio is generated. A period lasts one ON and OFF cycle.
The "ON Time" and "OFF Time" are each expressed as a number of samples and are set in an input field which opens when ON/OFF ratio is selected.


Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:ONTime on page 87
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:OFFTime on page 87

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:MODE on page 86

### 3.10.3 Marker Delay

The delay of the signals on the MARKER outputs is set in the"Marker Delay" section.

The R&S SMBV supports only two markers.

**Marker x Delay**
Enters the delay between the marker signal at the marker outputs and the start of the frame or slot.

The input is expressed as a number of symbols/samples. If the setting "Fix marker delay to dynamic range" is enabled, the setting range is restricted to the dynamic range. In this range the delay of the marker signals can be set without restarting the marker and signal.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay on page 85

**Current Range without Recalculation**
Displays the dynamic range within which the delay of the marker signals can be set without restarting the marker and signal.

The delay can be defined by moving the setting mark.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MINimum?
on page 86
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MAXimum?
on page 86

**Fix marker delay to current range**
Restricts the marker delay setting range to the dynamic range. In this range the delay can be set without restarting the marker and signal.

Remote command:
[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut:DELay:FIXed on page 85

### 3.10.4 Clock Settings

The Clock Settings is used to set the clock source and a delay if required.

**Sync. Mode**
(for R&S SMBV only)

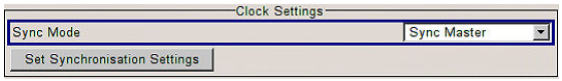
Selects the synchronization mode.

This parameter is used to enable generation of very precise synchronous signal of several connected R&S SMBVs.

**Note:** If several instruments are connected, the connecting cables from the master instrument to the slave one and between each two consecutive slave instruments must have the same length and type.
Avoid unnecessary cable length and branching points.

"None" The instrument is working in stand-alone mode.

"Sync. Master" The instrument provides all connected instrument with its synchronisation (including the trigger signal) and reference clock signal.




"Sync. Slave" The instrument receives the synchronisation and reference clock signal from another instrument working in a master mode.

Remote command:
[:SOURce<hw>]:BB:BTOoth:CLOCk:SYNChronization:MODE on page 91

**Set Synchronization Settings**
(for R&S SMBV only)

Performs automatically adjustment of the instrument's settings required for the synchronization mode, selected with the parameter "Synchronization Mode".

Remote command:
[:SOURce<hw>]:BB:BTOoth:CLOCk:SYNChronization:EXECute on page 90

**Clock Source**
Selects the clock source.

"Internal" The internal clock reference is used to generate the symbol clock.

"External" The external clock reference is fed in as the symbol clock or multiple thereof via the CLOCK connector.
The symbol rate must be correctly set to an accuracy of +/-2 % (see data sheet).
The polarity of the clock input can be changed with the aid of "Global Trigger/Clock Settings".
In the case of two-path instruments this selection applies to path A.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:CLOCk:SOURce` on page 90

**Clock Mode**
(This feature is available for the external clock source only.)

Selects the type of externally supplied clock.

"Sample" A sample clock is supplied via the CLOCK connector.

"Multiple Sample" A multiple of the sample clock is supplied via the CLOCK connector. The sample clock is derived internally from this. The value range is 1 to 64.
The "Clock Multiplier" field provided allows the multiplication factor to be entered.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:CLOCk:MODE` on page 89

**Clock Multiplier**
(This feature is available for the external clock source only.)

Enters the multiplication factor for clock type Multiple Sample.

Remote command:
`[:SOURce<hw>]:BB:BTOoth:CLOCk:MULTiplier` on page 89

**Measured External Clock**
Provided for permanent monitoring of the enabled and externally supplied clock signal.

Remote command:
`CLOCk:INPut:FREQuency?`

## 3.10.5 Global Settings

The buttons in this section lead to dialogs for general trigger, clock and mapping settings.

**Global Trigger/Clock Settings**
Calls the "Global Trigger/Clock/Input Settings" dialog.

This dialog is used among other things for setting the trigger threshold, the input impedance and the polarity of the clock and trigger inputs.

The parameters in this dialog affect all digital modulations and standards, and are described in chapter "Global Trigger/Clock/Input Settings" in the Operating Manual.

**User Marker / AUX I/O Settings**
Calls the "User Marker AUX I/O Settings" dialog, used to map the connector on the rear of the instruments.

See also "User Marker / AUX I/O Settings" in the Operating Manual.

# 4 Remote-Control Commands

The following commands are required to perform signal generation with the Bluetooth options in a remote environment. We assume that the R&S Signal Generator has already been set up for remote operation in a network as described in the R&S Signal Generator documentation. A knowledge about the remote control operation and the SCPI command syntax are assumed.

**Conventions used in SCPI command descriptions**

For a description of the conventions used in the remote command descriptions, see section "Remote Control Commands" in the R&S Signal Generator operating manual.

The commands in the `SOURce:BB:BTOoth` subsystem are described in three sections, separated into general remote commands, commands for Packet Configuration settings and commands for Dirty Transmitter Test settings.

This subsystem contains commands for the primary and general settings of the Bluetooth standard. These settings concern activation and deactivation of the standard, setting filter, clock, trigger and clipping settings, defining the symbol rate variation and the sequence length, as well as the preset and power adjust setting.

### Common Suffixes

The following common suffixes are used in remote commands:

| Suffix | Value range | Description |
|---|---|---|
| `SOURce<hw>` | [1]\|2 | available baseband signals |
| `OUTPut<ch>` | 1 .. 4 | available markers<br>R&S SMBV supports two markers |
| `EXTernal<ch>` | 1\|2 | external trigger connectors |

### Placeholder <root>

For commands that read out or save files in the default directory, the default directory is set using command `MMEM:CDIRectory`. The examples in this description use the place holder `<root>` in the syntax of the command.

- `D:\` - for selecting the internal hard disk of a Windows instrument
- `E:\` - for selecting the memory stick which is inserted at the USB interface of a Windows instrument
- `/var/user/` - for selecting the internal flash card of a Linux instrument
- `/usb/` - for selecting the memory stick which is inserted at the USB interface of a Linux instrument.

Tasks (in manual or remote operation) that are also performed in the base unit in the same way are not described here.

In particular, this includes:

- Managing settings and data lists, i.e. storing and loading settings, creating and accessing data lists, accessing files in a particular directory, etc.
- Information on regular trigger, marker and clock signals as well as filter settings, if appropriate.
- General instrument configuration, such as configuring networks and remote operation
- Using the common status registers

For a description of such tasks, see the R&S Signal Generator operating manual.

The following commands specific to the Bluetooth are described here:

## 4.1 General Commands



**[:SOURce<hw>]:BB:BTOoth:BCText?**

Queries the state/roll of the controller.

**Return values:**

<BcText> string

**Connected**
(for data channel type)
Shows that the state is Connected.

**Advertiser**
(for advertising channel type)
the current bluetooth Controller Role is Advertiser
for all ADV-xxx packet types and SCAN_RSP.

**Scanner**
(for advertising channel type)
the current bluetooth Controller Role is Scanner
for SCAN_REQ packet type

**Initiator**
(for advertising channel type)
the current bluetooth Controller Role is Initiator
for CONNECT_REQ packet type

**Example:** `SOUR:BB:BTO:BCT?`
Queries the state/roll of the controller.

**Usage:** Query only

**Manual operation:** See "Bluetooth Controller State" on page 31

---

**[:SOURce<hw>]:BB:BTOoth:CTYPe** <CType>

Determines the channel type. Advertising and data are available.

**Parameters:**

<CType> ADVertising | DATA

**ADVertising**
Selects channel type Advertising.

**DATA**
Selects channel type Data. Devices in a connected state transmit data channel packets in connection events with a start point and an interval.

*RST: ADVertising

**Example:** `SOUR:BB:BTO:CTYP ADV`
channel type Advertising.
`SOUR:BB:BTO:CTYP DATA`
channel type Data.

**Manual operation:** See "Channel Type" on page 29

---

**[:SOURce<hw>]:BB:BTOoth:PRESet**

Sets the parameters of the digital standard to their default values (`*RST` values specified for the commands).

Not affected is the state set with the command SOURce<hw>:BB:BTOoth:STATe

[:SOURce<hw>]:BB:BTOoth:STATe

**Example:** BB:BTO:PRES
resets all the Bluetooth settings to default values.

**Usage:** Event

**Manual operation:** See "Set To Default" on page 18

---

**[:SOURce<hw>]:BB:BTOoth:PTYPe** <PType>

The available packets depend on the selected transport mode. All packet types as defined in the Bluetooth specifications are supported.

**Parameters:**

<PType> ID | NULL | POLL | FHS | DM1 | DH1 | DM3 | DH3 | DM5 | DH5 | AUX1 | ADH1 | ADH3 | ADH5 | AEDH1 | AEDH3 | AEDH5 | HV1 | HV2 | HV3 | DV | EV3 | EV4 | EV5 | EEV3 | EEV5 | EEEV3 | EEEV5

*RST: DH1

**Example:** BB:BTO:PTYP NULL
sets the packet type.

**Manual operation:** See "Packet Type" on page 22

---

**[:SOURce<hw>]:BB:BTOoth:SETTing:CATalog** <Catalog>

This command reads out the files with Bluetooth settings in the default directory. The default directory is set using command MMEM:CDIRectory. Only files with the file extension *.bto will be listed.

**Parameters:**

<Catalog> string

**Example:** MMEM:CDIR '<root>bluetooth'
sets the default directory to <root>bluetooth.
BB:BTO:SETT:CAT?
reads out all the files with Bluetooth settings in the default directory.
Response: 'Bluetooth_EDR','Bluetooth_SCO'
the files Bluetooth_EDR and Bluetooth_SCO are available.

**Manual operation:** See "Save/Recall..." on page 18

---

**[:SOURce<hw>]:BB:BTOoth:SETTing:DELete** <Filename>

This command deletes the selected file with Bluetooth settings. The directory is set using command MMEM:CDIRectory. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension *.bto will be deleted.

**Parameters:**
<Filename> string

**Example:** `BB:BTO:SETT:DEL '<root>bluetooth'`
deletes the specified file with Bluetooth settings.

**Manual operation:** See "Save/Recall..." on page 18

---

**[:SOURce<hw>]:BB:BTOoth:SETTing:LOAD** <Filename>

This command loads the selected file with Bluetooth settings. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. The file extension may be omitted. Only files with the file extension `*.bto` will be loaded.

**Parameters:**
<Filename> string

**Example:** `BB:BTO:SETT:LOAD 'bluetooth_1'`
loads file `bluetooth_1`.

**Manual operation:** See "Save/Recall..." on page 18

---

**[:SOURce<hw>]:BB:BTOoth:SETTing:STORe** <Filename>

This command stores the current Bluetooth settings into the selected file. The directory is set using command `MMEM:CDIRectory`. A path can also be specified, in which case the files in the specified directory are read. Only the file name has to be entered. Bluetooth settings are stored as files with the specific file extensions `*.bto`.

**Setting parameters:**
<Filename> string

**Example:** `BB:BTO:SETT:STOR 'bluetooth_1'`
stores the current Bluetooth settings into file `bluetooth_1`.

**Usage:** Setting only

**Manual operation:** See "Save/Recall..." on page 18

---

**[:SOURce<hw>]:BB:BTOoth:SETTing:STORe:FAST** <Fast>

Determines whether the instrument performs an absolute or a differential storing of the settings.

Enable this function to accelerate the saving process by saving only the settings with values different to the default ones.

**Note:** This function is not affected by the "Preset" function.

**Parameters:**
<Fast> 0 | 1 | OFF | ON
*RST: 1

**Manual operation:** See "Save/Recall..." on page 18

**[:SOURce<hw>]:BB:BTOoth:SLENgth** <SLength>

Sets the sequence length of the Bluetooth signal in number of frames. This signal is calculated in advance and output in the arbitrary waveform generator.

**Parameters:**

<SLength> integer

Range: depending on the number of states in dirty transmitter test to dynamics

*RST: 1

**Example:** `BB:BTO:SLEN 10`
sets the sequence length to 10 frames.

**Manual operation:** See "Sequence Length" on page 22

**[:SOURce<hw>]:BB:BTOoth:STATe** <State>

Activates the standard and deactivates all the other digital standards and digital modulation modes in the same path.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: 0

**Example:** `SOURce1:BB:BTOoth:STATe`

**Manual operation:** See "State" on page 18

**[:SOURce<hw>]:BB:BTOoth:STIMing** <SlotTiming>

Selects the Rx slot timing mode.

**Parameters:**

<SlotTiming> TX | LOOPback

*RST: TX

**Example:** `BB:BTO:PTYP DH3`
sets the packet type.
`BB:BTO:STIM LOOP`
selects loopback test mode.

**Manual operation:** See "Slot Timing" on page 22

**[:SOURce<hw>]:BB:BTOoth:TMODe** <TMode>

The command selects the transport mode.

**Parameters:**

<TMode> ACL | SCO | ESCO

**ACL**
The selected transport mode is used for a point-to-point multi-point link establishment between the master and all the slaves participating on the piconet.

**SCO**
The selected transport mode is used for a point-to-point link establishment between a master and a single slave in the pico-net.

**ESCO**
The selected transport mode is used for a symmetric or asym-metric point-to point link establishment between a master and a specific slave.

*RST: ACL

**Example:** `BB:BTO:TMOD ACL`
selects transport mode ACL.

**Manual operation:** See "Transport Mode" on page 22

---

**[:SOURce<hw>]:BB:BTOoth:VERSion?**

Queries the version of the Bluetooth standard underlying the definitions.

**Return values:**

<Version> string

**Example:** `BB:BTO:VERS?`
queries the Bluetooth version.
`Response: Version 2.1+EDR`

**Usage:** Query only

**Manual operation:** See "Bluetooth Version" on page 20

---

**[:SOURce<hw>]:BB:BTOoth:WAVeform:CREate** <Filename>

This command creates a waveform using the current settings of the "Bluetooth" menu. The file name is entered with the command. The file is stored with the predefined file extension `*.wv`. The file name and the directory it is stored in are user-definable.

**Setting parameters:**

<Filename> string

**Example:** `MMEM:CDIR '<root>waveform'`
sets the default directory to <root>waveform.
`BB:BTO:WAV:CRE 'blutooth_1'`
creates the waveform file `bluetooth.wv` in the default direc-tory.

**Usage:** Setting only

**Manual operation:** See "Generate Waveform File…" on page 20

## 4.2 Filter/Clipping Settings



**[:SOURce<hw>]:BB:BTOoth:CLIPping:LEVel** <Level>

Sets the limit for level clipping (Clipping). This value indicates at what point the signal is clipped. It is specified as a percentage, relative to the highest level. 100% indicates that clipping does not take place.

**Parameters:**

<Level> integer

Range: 1 to 100
Increment: 1
*RST: 100
Default unit: PCT

**Example:** `BB:BTO:CLIP:LEV 80`
sets the limit for level clipping to 80% of the maximum level.
`BB:BTO:CLIP:STAT ON`
activates level clipping.

**Manual operation:** See "Clipping Level" on page 54

**[:SOURce<hw>]:BB:BTOoth:CLIPping:MODE** <Mode>

The command sets the method for level clipping (Clipping).

**Parameters:**

<Mode> VECTor | SCALar

**VECTor**
The reference level is the amplitude | i+jq |.

**SCALar**
The reference level is the absolute maximum of the I and Q values.

*RST: VECTor

**Example:** `BB:BTO:CLIP:MODE VECT`
sets the amplitude as reference level.

**Manual operation:** See "Clipping Mode" on page 54

---

**[:SOURce<hw>]:BB:BTOoth:CLIPping:STATe** <State>

The command activates level clipping (Clipping). The value is defined with the command `BB:BTO:CLIPping:LEVel`, the mode of calculation with the command `BB:BTO:CLIPping:MODE`.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: OFF

**Example:** `BB:BTO:CLIP:STAT ON`
activates level clipping.

**Manual operation:** See "Clipping State" on page 54

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:TYPE** <Type>

Selects the filters used for π/4 DQPSK and 8DPSK modulations. This opens a selection window containing all the filters available to the instrument.

**Parameters:**

<Type> RCOSine | COSine | GAUSs | LGAuss | CONE | COF705 | COEQualizer | COFequalizer | C2K3x | APCO25 | SPHase | RECTangle | PGAuss | LPASs | DIRac | ENPShape | EWPShape

*RST: RCOSine

**Example:** `BB:BTO:FILT:TYPE RCOS`
sets the filter type RCOSine.

**Manual operation:** See "Filter (π/4 DQPSK section)" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:ILENgth** <ILength>

Sets the impulse length (the number of filter taps).

**Parameters:**

<ILength> integer

Range: 1 to depends on oversampling
*RST: 10

**Example:** `BB:BTO:FILT:ILEN 10`
sets the number of filter tabs to 10.

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:ILENgth:AUTO[:STATe]** <State>

The command acivates/deactivates the impulse length state. If activated, the most sensible parameter values are selected. The value depends on the coherence check.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: ON

**Example:** `BB:BTO:FILT:ILEN:AUTO ON`
the most sensible parameters are selected automatically.

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:OSAMpling** <OSampling>

The command sets the upsampling factor.

**Parameters:**

<OSampling> integer

Range: 1 to 32
*RST: 10

**Example:** `BB:BTO:FILT:OSAM 10`
sets the upsampling factor to 10.

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:OSAMpling:AUTO[:STATe]** <State>

The command activates/deactivates the upsampling factor state. If activated, the most sensible parameter values are selected. The value depends on the coherence check. If deactivated, the values can be changed manually.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: ON

**Example:** `BB:BTO:FILT:OSAM:AUTO ON`
the most sensible parameters are selected automatically.

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:MINDex** <MIndex>

Queries the modulation index resulting from the entered frequency deviation value.

**Parameters:**

<MIndex> string

**Example:** `BB:BTO:FILT:MIND`
Queries the modulation index
`Response: 0.5`

**Manual operation:** See "Modulation index" on page 54

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:MTYPe** <MType>

Queries the modulation type used for the current packet selection.

**Parameters:**
<MType> string

**Example:** `BB:BTO:FILT:MTYP?`
Queries the modulation type

**Manual operation:** See "Modulation type" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:MSETtings:FDEViation** <FDeviation>

The frequency deviation can be varied in a range from 100.0 kHz to 200.0 kHz.

**Parameters:**
<FDeviation> float

Range: depends on bluetooth mode to depends on bluetooth mode
Increment: 0.1
*RST: depends on bluetooth mode

**Example:** `BB:BTO:MSET:FDEV 160`
sets a frequency deviation.

**Manual operation:** See "Frequency deviation" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:APCO25** <Apco25>

Sets the roll-off factor for filter type APCO25.

**Parameters:**
<Apco25> float

Range: 0.05 to 0.99
Increment: 0.01
*RST: 0.2

**Example:** `BB:BTO:FILT:PAR:APCO25 0.2`
sets the roll-off factor to 0.2 for filter type APCO25.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:COSine** <Cosine>

Sets the roll-off factor for the Cosine filter type.

**Parameters:**

<Cosine> float

Range: 0 to 1.0
Increment: 0.01
*RST: 0.1

**Example:** `BB:BTO:FILT:PAR:COS 0.35`
sets the roll-off factor to 0.35 for filter type Cosine.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:FGAuss** <FGauss>

Sets the B x T for the Gauss filter type.

**Parameters:**

<FGauss> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.5

**Example:** `BB:BTO:FILT:PAR:FGA 0.5`
sets B x T to 0.5 for the Gauss filter type for the GFSK section of the packet.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:GAUSs** <Gauss>

Sets the B x T for the Gauss filter type.

**Parameters:**

<Gauss> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.5

**Example:** `BB:BTO:FILT:PAR:GAUS 0.5`
sets B x T to 0.5 for the Gauss filter type for π/4 DQPSK or 8DPSK sections.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:LPASs** <LPass>

Sets the cut off frequency factor for a lowpass filter (ACP Opt.).

**Parameters:**

<LPass> float

Range: 0.05 to 2
Increment: 0.01
*RST: 0.5

**Example:** `BB:BTO:FILT:PAR:LPAS 1`
sets the cut off frequency factor for a lowpass filter

**Manual operation:** See "Cut Off Frequency Factor" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:PGAuss** <PGauss>

Sets the B x T for the Pure Gauss filter type.

**Parameters:**
<PGauss> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 0.5

**Example:** `BB:BTO:FILT:PAR:PGA 0.5`
sets B x T to 0.5 for the Pure Gauss filter type.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:RCOSine** <RCosine>

Sets the roll-off factor for the Root Cosine filter type.

**Parameters:**
<RCosine> float

Range: 0 to 1.0
Increment: 0.01
*RST: 0.4

**Example:** `BB:BTO:FILT:PAR:RCOS 0.22`
sets the roll-off factor to 0. 22 for filter type Root Cosine.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

---

**[:SOURce<hw>]:BB:BTOoth:FILTer:PARameter:SPHase** <SPhase>

Sets the B x T for the Split Phase filter type.

**Parameters:**
<SPhase> float

Range: 0.15 to 2.5
Increment: 0.01
*RST: 2

**Example:** `BB:BTO:FILT:PAR:SPH 0.5`
sets B x T to 0.5 for the Split Phase filter type.

**Manual operation:** See "Roll Off Factor / B xT" on page 53

**[:SOURce<hw>]:BB:BTOoth:SRATe:VARiation** <Variation>

Sets the symbol rate.

**Parameters:**

<Variation> float

Range: 4E2 to 15E6
Increment: 1E-3
*RST: 1E6

**Example:** `BB:BTO:SRAT:VAR 1`
sets the symbol rate variation to 1 MHz.

**Manual operation:** See "Symbol Rate Variation" on page 54

## 4.3 Trigger Settings

**EXTernal<ch>**

The numeric suffix to `EXTernal<ch>` distinguishes between the external trigger via the TRIGGER 1 (suffix 1) and TRIGGER 2 (suffix 2) connector.



**[:SOURce<hw>]:BB:BTOoth:TRIGger:ARM:EXECute**

The command stops signal generation for trigger modes "Armed Auto" and "Armed Retrigger". A subsequent internal or external trigger event restart signal generation.

**Example:** `BB:BTO:TRIG:ARM:EXEC`
stops signal generation for trigger modes "Armed Auto "and "Armed Retrigger".

**Usage:** Event

**Manual operation:** See "Arm" on page 21

**[:SOURce<hw>]:BB:BTOoth:TRIGger:EXECute**

The command executes a trigger. The internal trigger source must be selected using the command `SOUR:BB:BTO:TRIG:SOUR INT` and a trigger mode other than "AUTO" must be selected using the command `SOUR:BB:BTO:TRIG:SEQ`.

**Example:** `BB:BTO:TRIG:SOUR INT`
sets internal triggering.
`BB:BTO:SEQ RETR`
sets Retrigger mode, i.e. every trigger event causes signal generation to restart.
`BB:BTO:TRIG:EXEC`
executes a trigger.

**Usage:** Event

**Manual operation:** See "Execute Trigger" on page 21

**[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal]:SYNChronize:OUTPut** <Output>

(enabled for "Trigger Source" External)

Enables/disables output of the signal synchronous to the external trigger event.

For R&S SMBV instruments:

See also "Sync. Output to External Trigger" on page 59 for a detailed description of the applications of this setting.

**Parameters:**

<Output> 0 | 1 | OFF | ON

**ON**
The signal calculation starts simultaneously with the external trigger event but because of the instrument's processing time the first samples are cut off and no signal is outputted. After elapsing of the internal processing time, the output signal is synchronous to the trigger event.

**OFF**
The signal output begins after elapsing of the processing time and starts with sample 0, i.e. the complete signal is outputted.

*RST: ON

**Example:** `BB:BTO:TRIG:SOUR EXT`
sets external triggering.
`BB:BTO:TRIG:EXT:SYNC:OUTP ON`
enables synchrounous output to external trigger

**Manual operation:** See "Sync. Output to External Trigger" on page 59

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:DELay** <Delay>

Specifies the trigger delay (expressed as a number of samples) for triggering by the trigger signal from the second path.

**Parameters:**

<Delay> float

Range: 0 to 65535
Increment: 0.01
*RST: 0

**Example:** `BB:BTO:TRIG:SOUR OBAS`
sets for path A the internal trigger executed by the trigger signal from the second path (path B).
`BB:BTO:TRIG:OBAS:DEL 50`
sets a delay of 50 symbols for the trigger.

**Manual operation:** See "Trigger Delay" on page 60

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OBASeband:INHibit** <Inhibit>

The command specifies the number of samples by which a restart is to be inhibited following a trigger event. This command applies only for triggering by the second path.

**Parameters:**

<Inhibit> integer

Range: 0 samples to (1<<26)-1u samples
Increment: 1 sample
*RST: 0 samples

**Example:** `BB:BTO:TRIG:SOUR OBAS`
sets for path A the internal trigger executed by the trigger signal from the second path (path B).
`BB:BTO:TRIG:OBAS:INH 200`
sets a restart inhibit for 200 samples following a trigger event.

**Manual operation:** See "Trigger Inhibit" on page 61

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:RMODe** <RMode>

Queries the current status of signal generation for all trigger modes with Bluetooth modulation on.

**Parameters:**

<RMode> RUN | STOP

**RUN**
the signal is generated. A trigger event occurred in the triggered mode.

**STOP**
the signal is not generated. A trigger event did not occur in the triggered modes, or signal generation was stopped by the command `:BB:BTO:TRIG:ARM:EXECute` (armed trigger modes only).

*RST: STOP

**Example:** `BB:BTO:SEQ ARET`
selects the Armed_Retrigger mode.
`BB:BTO:TRIG:RMOD?`
queries the current status of signal generation.
`Response: RUN`
the signal is generated, an external trigger was executed.

**Manual operation:** See "Running/Stopped" on page 58

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:SLENgth** <SLength>

Defines the length of the signal sequence to be output in the Single trigger mode (`SOUR:BB:BTO:SEQ SING`). The unit is defined with command `SOUR:BB:BTO:TRIG:SLUNit`.

**Parameters:**

<SLength> integer

Range: 1 to 7000
*RST: 1

**Example:** `BB:BTO:SEQ SING`
sets trigger mode Single.
`BB:BTO:TRIG:SLUN FRAM`
sets unit frames for the entry of sequence length.
`BB:BTO:TRIG:SLEN 200`
sets a sequence length of 200 frames.

**Manual operation:** See "Signal Duration" on page 58

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:SLUNit** <SlUnit>

Defines the unit for the entry of the length of the signal sequence (`SOUR:BB:BTO:TRIG:SLEN`) to be output in the "Single" trigger mode (`SOUR:BB:BTO:SEQ SING`).

**Parameters:**

<SlUnit> FRAMe | SEQuence | EVENt

*RST: SEQuence

**Example:** `BB:BTO:SEQ SING`
sets trigger mode Single.
`BB:BTO:TRIG:SLUN FRAM`
sets unit frames for the entry of sequence length.
`BB:BTO:TRIG:SLEN 2`
sets a sequence length of 2 frames. The current frame will be output twice after the next trigger event.

**Manual operation:** See "Signal Duration Unit" on page 58

**[:SOURce<hw>]:BB:BTOoth:TRIGger:SOURce** <Source>

Selects the trigger source.

**Parameters:**

<Source> INTernal|OBASeband|BEXTernal|EXTernal

**INTernal**
manual trigger or *TRG.
**EXTernal|BEXTernal**
trigger signal on the TRIGGER 1/2 connector.
**OBASeband**
trigger signal from the other path

*RST: INTernal

**Example:** SOURce1:BB:BTOoth:TRIGger:SOURce EXTernal
sets external triggering via the TRIGGER 1 connector.

**Manual operation:** See "Trigger Source" on page 59

**[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal<ch>]:DELay** <Delay>

Specifies the trigger delay (expressed as a number of samples) for external triggering.

The numeric suffix to EXTernal distinguishes between the external trigger via the TRIGGER 1 (suffix 1) and TRIGGER 2 (suffix 2) connector.

**Parameters:**

<Delay> float

Range: 0 to 65535
Increment: 0.01
*RST: 0

**Example:** BB:BTO:TRIG:SOUR EXT
sets an external trigger via the TRIGGER 1 connector.
BB:BTO:TRIG:EXT:DEL 50
sets a delay of 50 symbols for the trigger.

**Manual operation:** See "Trigger Delay" on page 60

**[:SOURce<hw>]:BB:BTOoth:TRIGger[:EXTernal<ch>]:INHibit** <Inhibit>

Specifies the number of samples by which a restart is to be inhibited following a trigger event. This command applies only in the case of external triggering. The numeric suffix to EXTernal distinguishes between the external trigger via the TRIGGER 1 (suffix 1) and TRIGGER 2 (suffix 2) connector.

**Parameters:**

<Inhibit> integer

Range: 0 to 67108863
*RST: 0

**Example:** `BB:BTO:TRIG:SOUR EXT`
selects an external trigger via the TRIGGER 1 connector.
`BB:BTO:TRIG:EXT:INH 200`
sets a restart inhibit for 200 samples following a trigger event.

**Manual operation:** See "Trigger Inhibit" on page 61

---

**[:SOURce<hw>]:BB:BTOoth[:TRIGger]:SEQuence** <Sequence>

The command selects the trigger mode.

**Parameters:**

<Sequence> AUTO | RETRigger | AAUTo | ARETrigger | SINGle

**AUTO**
The modulation signal is generated continuously.

**RETRigger**
The modulation signal is generated continuously. A trigger event (internal or external) causes a restart.

**AAUTo**
The modulation signal is generated only when a trigger event occurs. After the trigger event the signal is generated continuously, signal generation is stopped with command `SOUR:BB:BTO:TRIG:ARM:EXEC` and started again when a trigger event occurs.

**ARETrigger**
The modulation signal is generated only when a trigger event occurs. The device automatically toggles to RETRIG mode. Every subsequent trigger event causes a restart.
Signal generation is stopped with command `SOUR:BB:BTO:TRIG:ARM:EXEC` and started again when a trigger event occurs.

**SINGle**
The modulation signal is generated only when a trigger) event occurs. After the trigger event, the signal is generated once to the set sequence length (`SOUR:BB:BTO:TRIG:SLEN`). Every subsequent trigger event causes a restart.

*RST: AUTO

**Example:** `BB:BTO:SEQ AAUT`
sets the "Armed_auto" trigger mode; the device waits for the first trigger (e.g. with `*TRG`) and then generates the signal continuously.

**Manual operation:** See "Trigger Mode" on page 57

## 4.4 Marker Settings

This section lists the remote control commands, necessary to configure the markers.

**OUTPut<ch>**

The numeric suffix to OUTPut distinguishes between the available markers.

Only two markers are available for the R&S SMBV, i.e. the allowed values for the suffix are 1 or 2.



---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut:DELay:FIXed** <Fixed>

The command restricts the marker delay setting range to the current range. In this range the delay can be set without restarting the marker and signal. If a delay is entered in setting ON but is outside this range, the maximum possible delay is set and an error message is generated.

The numeric suffix in OUTPut has no significance for this command, since the setting always affects every marker.

**Parameters:**

<Fixed> 0 | 1 | OFF | ON
*RST: OFF

**Example:** BB:BTO:TRIG:OUTP:DEL:FIX ON
restricts the marker signal delay setting range to the current range.

**Manual operation:** See "Fix marker delay to current range" on page 63

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay** <Delay>

Defines the delay between the signal on the marker outputs and the start of the signal, expressed in terms of samples. Command BB:BTO:TRIGger:OUTPut:DELay:FIXed can be used to restrict the range of values to the dynamic range, i.e. the range within which a delay of the marker signals can be set without restarting the marker and signal.

**Parameters:**

<Delay> float
Range: 0 to 16777215
Increment: 1E-3
*RST: 0

**Example:** `BB:BTO:TRIG:OUTP2:DEL 1600`
sets a delay of 1600 samples for the corresponding marker signal.

**Manual operation:** See "Marker x Delay" on page 62

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MINimum?**
**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:DELay:MAXimum?**

Queries the maximum marker delay for setting `:BB:BTO:TRIG:OUTP:DEL:FIX ON`.

**Return values:**
<Maximum> float

Range: 0 to 16777215
Increment: 1E-3
*RST: 2000

**Example:** `BB:BTO:TRIG:OUTP:DEL:FIX ON`
restricts the marker signal delay setting range to the dynamic range.
`BB:BTO:TRIG:OUTP:DEL:MAX?`
queries the maximum of the dynamic range.
`Response: 2000`
the maximum for the marker delay setting is 2000 samples.

**Usage:** Query only

**Manual operation:** See "Current Range without Recalculation" on page 62

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:MODE** <Mode>

The command defines the signal for the selected marker output.

**Parameters:**

<Mode> RESTart | STARt | ACTive | PULSe | PATTern | RATio | TRIGger

**RESTart**
A marker signal is generated at the start of each signal sequence.

**STARt**
A marker signal is generated at the start of each frame

**ACTive**
The marker masks the active part of the frame. At the start of each burst, the marker signal changes to high. It changes back to low after the end of each burst.

**PULSe**
A regular marker signal is generated. The clock frequency is defined by entering a divider. The frequency is derived by dividing the symbol rate by the divider. The input box for divider opens when Pulse is selected, and the resulting pulse frequency is displayed below.

**PATTern**
A marker signal that is defined by a bit pattern is generated. The pattern has a maximum length of 32 bits and is defined in an input field which opens when pattern is selected.

**RATio**
A regular marker signal corresponding to the Time Off / Time On specifications in the commands
`SOURce:BB:BTO:TRIGger:OUTPut:OFFTime` and
`SOURce:BB:BTO:TRIGger:OUTPut:ONTime` is generated.

**TRIGger**
A received internal or external trigger signal is output at the marker connector.

*RST: RESTart

**Example:** `BB:BTO:TRIG:OUTP:MODE REST`
selects the marker generation at the start of each signal sequence

**Manual operation:** See "Marker Mode" on page 61

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:ONTime** <OnTime>
**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:OFFTime** <OffTime>

Sets the number of samples in a period (ON time + OFF time) during which the marker signal in setting `SOURce:BB:BTO:TRIGger:OUTPut:MODE RATio` on the marker outputs is OFF.

**Parameters:**

<OffTime> integer

Range: 1 to 16777215
*RST: 1

**Example:** `BB:BTO:TRIG:OUTP:OFFT 2000`
sets an OFF time of 2000 samples for the corresponding marker signal.

**Manual operation:** See "Marker Mode" on page 61

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PATTern** <Pattern>

Selects the data for a pattern.

**Parameters:**
<Pattern> 64 bits
*RST: #H2,2

**Example:** `BB:BTO:TRIG:OUTP:MODE PATT`
sets the marker mode for the corresponding marker signal
`BB:BTO:TRIG:OUTP:PATT #B010101,6`
sets the pattern

**Manual operation:** See "Marker Mode" on page 61

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PULSe:DIVider** <Divider>

The clock frequency is defined by entering a divider. The frequency is derived by dividing the symbol rate by the divider. The input box for divider opens when Pulse is selected, and the resulting pulse frequency is displayed below it.

**Parameters:**
<Divider> integer
Range: 2 to 1024
*RST: 2

**Example:** `BB:BTO:TRIG:OUTP:MODE:PULS`
selects marker mode Pulse
`BB:BTO:TRIG:OUTP:PULS:DIV 2`
sets a divider for the clock frequency.

**Manual operation:** See "Marker Mode" on page 61

---

**[:SOURce<hw>]:BB:BTOoth:TRIGger:OUTPut<ch>:PULSe:FREQuency?**

Queries the marker oulse frequency.

**Return values:**
<Frequency> float
Range: 2 to 1024
Increment: 1E-3
*RST: 2

**Example:** `BB:BTO:TRIG:OUTP:MODE:PULS`
selects marker mode Pulse
`BB:BTO:TRIG:OUTP:PULS:DIV 2`
sets a divider for the clock frequency
`BB:BTO:TRIG:OUTP:PULS:FREQ`
displays resulting pulse frequency

**Usage:** Query only

**Manual operation:** See "Marker Mode" on page 61

## 4.5 Clock Settings

This section lists the remote control commands, necessary to configure the clock.



---

**[:SOURce<hw>]:BB:BTOoth:CLOCk:MODE** <Mode>

Enters the type of an externally supplied clock. When `MSAMple` is used, a multiple of the clock is supplied and the clock is derived internally from this. The multiplier is entered with the command `[:SOURce<hw>]:BB:BTOoth:CLOCk:MULTiplier`.

For two-path instruments, the only numerical suffix allowed for SOURce is 1, since the external clock source is permanently allocated to path A.

**Parameters:**

<Mode> SAMPle | MSAMple
*RST: SAMPle

**Example:** `SOURce1:BB:BTO:CLOCk:MODE MSAMple`
sets the type of externally supplied clock.

**Manual operation:** See "Clock Mode" on page 64

---

**[:SOURce<hw>]:BB:BTOoth:CLOCk:MULTiplier** <Multiplier>

Specifies the multiplication factor for clock type multiple (`:BB:BTO:CLOCk:MODE MSAMple`) in the case of an external clock source.

For two-path instruments, the only numerical suffix allowed for `SOURce` is 1, since the external clock source is permanently allocated to path A.

**Parameters:**

<Multiplier> integer

Range: 1 to 64
Increment: 1
*RST: 4

**Example:** `SOURce1:BB:BTO:CLOCk:SOURce EXT`
selects the external clock source.
`SOURce1:BB:BTO:CLOCk:MODE MSAMple`
sets the clock type
`SOURce1:BB:BTO:CLOC:MULTiplier 12`
the multiplier for the external clock rate is 12.

**Manual operation:** See "Clock Multiplier" on page 64

---

**[:SOURce<hw>]:BB:BTOoth:CLOCk:SOURce** <Source>

Selects the clock source.

For two-path instruments, selecting `EXTernal` is only possible for path A, since the external clock source is permanently allocated to path A. Selection `AINternal` is only possible for path B.

**Parameters:**

<Source> INTernal | EXTernal | AINTernal

**INTernal**
The internal clock reference is used.

**EXTernal**
The external clock reference is supplied to the CLOCK connector. Commands `:BB:BTO:CLOCk:MODE` and `:MULTiplier` are used to enter the type of the external clock.

**AINTernal**
The clock source of path A is used for path B.

*RST: INTernal

**Example:** `BB:BTO:CLOC:SOUR EXT`
selects the external clock source. The clock is supplied via the CLOCK connector.
`BB:BTO:CLOC:MODE MSAM`
selects clock type "Multiple Sample", i.e. the supplied clock has a rate which is a multiple of the sample rate.
`BB:BTO:CLOC:MULT 12`
the multiplier for the external clock rate is 12.

**Manual operation:** See "Clock Source" on page 63

---

**[:SOURce<hw>]:BB:BTOoth:CLOCk:SYNChronization:EXECute**

Performs automatically adjustment of the instrument's settings required for the synchronization mode, set with the command `BB:BTO:CLOC:SYNC:MODE`.

**Example:** `BB:BTO:CLOC:SYNC:MODE MAST`
the instrument is configured to work as a master one.
`BB:BTO:CLOC:SYNC:EXEC`
all synchronizations settings are adjusted accordingly.

**Usage:** Event

**Manual operation:** See "Set Synchronization Settings" on page 63

---

**[:SOURce<hw>]:BB:BTOoth:CLOCk:SYNChronization:MODE** <Mode>

Selects the synchronization mode.

This parameter is used to enable generation of very precise synchronous signal of several connected R&S SMBVs.

**Note:** If several instruments are connected, the connecting cables from the master instrument to the slave one and between each two consecutive slave instruments must have the same length and type. Avoid unnecessary cable length and branching points.

**Parameters:**

<Mode> NONE | MASTer | SLAVe

**NONE**
The instrument is working in stand-alone mode.

**MASTer**
The instrument provides all connected instrument with its synchronization (including the trigger signal) and reference clock signal.

**SLAVe**
The instrument receives the synchronization and reference clock signal from another instrument working in a master mode.

*RST: NONE

**Example:** `BB:BTO:CLOC:SYNC:MODE MAST`
the instrument is configured to work as a master one.

**Manual operation:** See "Sync. Mode" on page 63

## 4.6 Power Ramping



---

**[:SOURce<hw>]:BB:BTOoth:PRAMping:FOFFset** <FOffset>

The command sets the offset of the falling edge of the envelope at the end of a burst. A positive value introduces a guard period after the end of the packet and negative value moves the ramp into the end part of the transmitted packet.

**Parameters:**

<FOffset> integer

Range: -32 to 32
Increment: 1
*RST: 0
Default unit: symbols

**Example:** `BB:BTO:PRAM:FOFF 8.0`
sets the offset in the falling edge of the envelope to 8.0 symbols.

**Manual operation:** See "Fall Offset" on page 56

---

**[:SOURce<hw>]:BB:BTOoth:PRAMping:RFUNction** <RFunction>

The command selects the form of the transmitted power, i.e. the shape of the rising and falling edges during power ramp control.

**Parameters:**

<RFunction> LINear | COSine

*RST: COSine

**Example:** `BB:BTO:PRAM:RFUN LIN`
sets linear shape for the rising and falling edges during power ramp control.

**Manual operation:** See "Ramp Function" on page 55

---

**[:SOURce<hw>]:BB:BTOoth:PRAMping:ROFFset** <ROffset>

The command sets the offset in the rising edge of the envelope at the start of a burst. A positive value moves the ramp into beginning of a transmitted packet and a negative value introduce an additional guard period after the end of the packet.

**Parameters:**

<ROffset> integer

Range: -32 symbols to 32 symbols
Increment: 1 symbol
*RST: 0 symbols

**Example:** `BB:BTO:PRAM:ROFF 8.0`
sets the offset in the rising edge of the envelope to 8.0 symbols.

**Manual operation:** See "Rise Offset" on page 55

---

**[:SOURce<hw>]:BB:BTOoth:PRAMping:RTIMe** <RTime>

The command sets the power ramping rise time and fall time for a burst.

**Parameters:**

<RTime> integer

Range: 1 symbol to 32 symbols
Increment: 1 symbol
*RST: 1 symbol

**Example:** `BB:BTO:PRAM:TIME 2.0`
sets the power ramping rise time and fall time for a burst to 2 samples.

**Manual operation:** See "Ramp Time" on page 55

## 4.7 Packet Configuration Setting



---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:ACKNowledgement**
<Acknowledgement>

This command sets the ARQN bit of the packet header..

**Parameters:**

<Acknowledgement> NAK | ACK

**NAK**
Request to retransmit the previous payload.

**ACK**
Previous payload has been received successfully.

*RST: ACK

**Example:** `BB:BTO:PTYP DH1`
selects the packet type DH1.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:ACKN ACK`
sets positive acknowledgement

**Manual operation:** See "Acknowledgment" on page 25

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDALap** <BdaLap>

Sets the lower address part of Bluetooth Device Address. The length of LAP is 24 bits or 6 hexadecimal figures.

**Parameters:**

<BdaLap> 24 bits

Range: #H000000 to #HFFFFFF
*RST: 80

**Example:** `BB:BTO:PCON:BDAL #H000000, 24`
sets the lower address part

**Manual operation:** See "Bluetooth Device Address (BD_ADDR)" on page 24

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDANap** <BdaNap>

Enters the non-significant address part of Bluetooth Device Address. The length of NAP is 16 bits or 4 hexadecimal figures.

**Parameters:**

<BdaNap> 16 bits

Range: #H0000 to #HFFFF
*RST: ABCD

**Example:** `BB:BTO:PCON:BDAN #H0000, 16`
sets the non-significant address part

**Manual operation:** See "Bluetooth Device Address (BD_ADDR)" on page 24

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:BDAUap** <BdaUap>

Enters the upper address part of Bluetooth Device Address. The length of UAP is 8 bits or 2 hexadecimal figures.

**Parameters:**

<BdaUap> 8 bits

Range: #H00 to #HFF
Increment: 1
*RST: 48

**Example:** `BB:BTO:PCON:BDAN #H00, 8`
sets the non-significant address part

**Manual operation:** See "Bluetooth Device Address (BD_ADDR)" on page 24

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:CODevice** <CoDevice>

A parameter received during the device discovery procedure, indicates the type of device and which types of service that are supported.

**Parameters:**

<CoDevice> 24 bits

Range: #H000000 to #HFFFFFF

**Example:** `BB:BTO:PTYP FHS`
sets the packet type
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:COD '020104'`
sets the class of device

**Manual operation:** See "Class of Device" on page 27

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA** <Data>

Selects the data source used for the payload.

**Parameters:**

<Data> ALL0 | ALL1 | PATTern | PN09 | PN11 | PN15 | PN16 | PN20 | PN21 | PN23 | DLISt

*RST: PN09

**Example:** `BB:BTO:PTYP FHS`
sets the packet type
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:DATA ALL1`
sets the data type.

**Manual operation:** See "Data List Management…" on page 19

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DPATtern** <DPattern>

Selects the data for a pattern.

**Parameters:**

<DPattern> 64 bits

*RST: #H0,1

**Example:** `BB:BTO:PCON:DATA PATT`
sets the data type.
`BB:BTO:PCON:DATA:DPAT #B010101, 6`
selects the data for a pattern

**Manual operation:** See "Data Source" on page 26

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:DSELection** <DSelection>

The command selects data list file.

**Parameters:**

<DSelection> string

Increment: 1

**Example:** `BB:BTO:PCON:DATA DLIS`
selects the data type.
`BB:BTO:PCON:DSEL bluetooth_1`
selects the file for the data.

**Manual operation:** See "Data List Management…" on page 19

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDPAttern** <VdPattern>

Sets the bit pattern for the voice data.

**Parameters:**

<VdPattern> 64 bits

Increment: 1
*RST: #H0,1

**Example:** `BB:BTO:PCON:DATA:PATT`
selects the data type.
`BB:BTO:PCON:DATA:VDPA #B010101, 6`
selects the bit pattern for the voice data.

**Manual operation:** See "Data Source (Voice Field)" on page 27

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DATA:VDSElection** <VdSelection>

The command selects the data list for voice data.

**Parameters:**

<VdSelection> string

Increment: 1

**Example:** `BB:BTO:PCON:VDAT DLIS`
selects the data type.
`BB:BTO:PCON:VDSE bluetooth_1`
selects the file for the data.

**Manual operation:** See "Data Source (Voice Field)" on page 27

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DLENgth** <DLength>

Sets the payload data length in bytes.

**Parameters:**

<DLength> integer

Range: 0 to depends on packet type
Increment: 1
*RST: depends on packet type

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:DLEN 25`
sets the data length.

**Manual operation:** See "Data Length" on page 26

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DSFPacket** <DsfPacket>

The command selects the data source for the selected packet type.

**Parameters:**

<DsfPacket> PEDit | ADATa

**PED**
Enables Packet Editor. All packet fields can be configured individually.

**ADAT**
Fills the generated packets with the selected data source. Useful if predefined data contents are loaded with a data list file or the data contents of the packet are not of interest.

*RST: PEDit

**Example:** `BB:BTO:PCON:DSFP PED`
enables packet editor under data source for packet.

**Manual operation:** See "Data Source for Packet" on page 24

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:DWHitening** <DWhitening>

The command activates or deactivates the Data Whitening.

**Parameters:**

<DWhitening> 0 | 1 | OFF | ON

*RST: OFF

**Example:** `BB:BTO:PCON:DWH ON`
activates data whitening.

**Manual operation:** See "Data Whitening" on page 24

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:EIRPacketfollows** <EirPacketFollow>

The command indicates that an extended inquiry response packet may follow.

**Parameters:**

<EirPacketFollow> YES | NO

**YES**
Indicates that EIR packet follows.

**NO**
Indicates that EIR packet does not follow.

*RST: NO

**Example:** `BB:BTO:PCON:PTYP FHS`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:EIRP YES`
the EIR packet follows.

**Manual operation:** See "EIR packet follows" on page 27

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:HFControl** <HfControl>

The command sets the FLOW bit in the header. This bit indicates start or stop of transmission of packets over the ACL logical transport.

**Parameters:**

<HfControl> GO | STOP

**GO**
Allows the other devices to transmit new data.

**STOP**
Stops the other devices from transmitting data temporarily.

*RST: GO

**Example:** `BB:BTO:PCON:PTYP DH1`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet.
`BB:BTO:PCON:HFC GO`
allows the other devices to transmit new data.

**Manual operation:** See "Flow Control (Header)" on page 25

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:LTADdress** <LtAddress>

The command enters the logical transport address for the header. Each slave active in a piconet is assigned a primary logical transport address (LT_ADDR). The all-zero LT_ADDR is reserved for broadcast messages.

**Parameters:**

<LtAddress> integer

Range: 0 to 7
Increment: 1
*RST: 0

**Example:** `BB:BTO:PCON:PTYP DH1`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:LTAD 0`
sets the logical transport address equal zero.

**Manual operation:** See "Logical Transport Address" on page 25

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:PFControl** <PfControl>

The command sets the FLOW bit in the payload (flow control per logical link).

**Parameters:**
<PfControl> GO | STOP

**GO**
Indicates the start of transmission of ACL packets after a new connection has been established.

**STOP**
Indicates the stop of transmission of ACL packets before an additional amount of payload data is sent.

*RST: GO

**Example:** `BB:BTO:PCON:PTYP DH1`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enable packet editor under data source for packet
`BB:BTO:PCON:PFC GO`
allows the flow per logical link.

**Manual operation:** See "Flow Control (Payload)" on page 26

---

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:PLENgth** <PLength>

Sets the packet length in symbols.

**Parameters:**
<PLength> integer

Range: 1 to depends on packet type
Increment: 1
*RST: depends on packet type

**Example:** `BB:BTO:PCON:DSFP ADAT`
fills the all data under data source for packet.
`BB:BTO:PCON:PLEN 1`
sets the packet length.

**Manual operation:** See "Packet Length" on page 27

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:SNSValue** <SnSvalue>

The command sets the start value of the header SEQN bit. The SEQN bit is present in the header to filter out retransmissions in the destination. The signal generator is altering this bit automatically on consecutive frames, if a sequence length of at least 2 frames is set.

**Parameters:**

<SnSvalue> integer

Range: 0 to 1
Increment: 1
*RST: 1

**Example:** `BB:BTO:PCON:PTYP DH1`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enables packet editor under data source for packet.
`BB:BTO:PCON:SNSV ONE`
sets the SEQN bit of the first CRC data packet at the start of a connection.

**Manual operation:** See "SEQN Start Value" on page 25

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:SRMode** <SrMode>

The command indicates the interval between two consecutive page scan windows, determines the behavior of the paging device.

**Parameters:**

<SrMode> R0 | R1 | R2

**R0**
The scan interval is equal to the scan window T w page scan (continuous nscan) and maximal 1.28s.

**R1**
The scan interval is maximal 1.28s.

**R2**
The scan interval is maximal 2.56s.

*RST: R0

**Example:** `BB:BTO:PCON:PTYP FHS`
sets the packet type.
`BB:BTO:PCON:DSFP PED`
enables packet editor under data source for packet.
`BB:BTO:PCON:SRM R0`
sets the scan repetition mode.

**Manual operation:** See "Scan Repetition Mode" on page 27

**[:SOURce<hw>]:BB:BTOoth:PCONfiguration:VDATa** <VData>

The command selects the data source for the voice field.

**Parameters:**

<VData> ALL0 | ALL1 | PATTern | PN09 | PN11 | PN15 | PN16 | PN20 | PN21 | PN23 | DLISt

*RST: PN09

**Example:** `BB:BTO:PCON:VDAT ALL1`
sets the voice data type.

**Manual operation:** See "Data Source (Voice Field)" on page 27

## 4.8 Dirty Transmitter Test



---

**[:SOURce<hw>]:BB:BTOoth:DTTest:DTTState** <DttState>

The command activates/deactivates the Dirty Transmitter Test. For Basic Rate packets, each set of parameters in the Dirty Transmitter Setting table below is used for a duration of 20 ms. After 20 ms, the following set is used, continuing with the first set after the sequence is completed.

For EDR packets, the parameter sets apply for 20 packets each.

**Parameters:**

<DttState> 0 | 1 | OFF | ON

*RST: OFF

**Example:** `BB:BTO:DTT:DTTS ON`
activates the Dirty Transmitter Test.

**Manual operation:** See "Dirty Transmitter Test" on page 49

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:FDDeviation** <FdDeviation>

Sets a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset. The maximum deviation reached during the drift equals the set frequency drift deviation.

**Parameters:**

<FdDeviation> integer

Range: -100 to 100
*RST: 25

**Example:** `BB:BTO:DTT:FDD 25`
enters frequency drift deviation.

**Manual operation:** See "Frequency Drift Deviation (+/-)" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:FDRate** <FdRate>

Sets a frequency drift rate.

A sine wave is used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset with the set frequency drift rate.

**Parameters:**

<FdRate> 0.3 KHz | 0.5 KHz | 1.6 KHz | 10 KHz

Range: depends on packet type to depends on packet type
Increment: 0.001
*RST: depends on packet type

**Example:** `BB:BTO:DTT:FDR 1.6`
enters frequency drift rate.

**Manual operation:** See "Frequency Drift Rate" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:SPHase** <SPhase>

The command enters a start phase.

The start phase of the sine wave used to drift the modulated Bluetooth signal around center frequency + carrier frequency offset is set here.

**Parameters:**

<SPhase> integer

Range: 0 to 359
Increment: 1
*RST: 0
Default unit: degree

**Example:** `BB:BTO:DTT:SPH 0`
enters a start phase.

**Manual operation:** See "Start Phase" on page 49

**[:SOURce<hw>]:BB:BTOoth:DTTest:STDefault**

The command calls the default settings for the Dirty Transmitter Test.

**Example:** `BB:BTO:DTT:STD`
calls the default settings.

**Usage:** Event

**Manual operation:** See "Set to Default" on page 49

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe** <Table>

Opens the table settings.

**Parameters:**
<Table> NOTable | SHORt | LONG

**Example:** `BB:BTO:PTYP DH1`
calls the default settings.
`BB:BTO:DTT:TABL LONG`
calls the default settings.

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:CFOFfset** <CfOffset>

Sets a carrier frequency offset.

The carrier frequency offset shows the deviation of the transmitted initial center frequency from carrier frequency.

**Parameters:**
<CfOffset> integer
Range: -150 to 150
*RST: 1
Default unit: kHz

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL LONG`
enters the table type
`BB:BTO:DTT:TABL:LONG:SET2:CFOF 14`
enters a carrier frequency offset.

**Manual operation:** See "Carrier Frequency Offset kHz" on page 50

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:MINDex** <MIndex>

Sets the modulation index.

The modulation index specifies the frequency deviation.

The modulation index h is defined as

$$h = \frac{2\Delta f}{f_{symbol}}$$

where

$f_{symbol}$ = "symbol rate" , set with the command `[:SOURce<hw>]:BB:BTOoth:SRATe:VARiation`

$\Delta f$ = "frequency deviation", set with the command `[:SOURce<hw>]:BB:BTOoth:MSETtings:FDEViation`

According to the Bluetooth standard, the modulation index is allowed to vary between 0.28 and 0.35.

**Parameters:**

<MIndex> float

Range: 0.28 to 0.55
Increment: 0.01
*RST: 0.28

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL LONG`
enters the table type
`BB:BTO:DTT:TABL:LONG:SET2:MIND 0.3`
enters a modulation index.

**Manual operation:** See "Modulation Index" on page 51

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:STATe** <State>

Activates or deactivates the corresponding parameter set for the long table. If a set deactivated, its parameters are skipped in the sequence. Instead, the next active set is used.

For Basic Rate packets, each set applies to 20ms of signal.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: 1

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL LONG`
enters the table type
`BB:BTO:DTT:TABL:LONG:SET2:STAT ON`
activates the set 2 in the long table.

**Manual operation:** See "State" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:LONG:SET<ch>:STERror** <StError>

Sets a symbol timing error in ppm.

The Symbol Timing Error modifies the symbol clock frequency by the set amount.

**Parameters:**

<StError> integer

Range: -150 to 150
Increment: 1
*RST: 1
Default unit: ppm

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL LONG`
enters the table type
`BB:BTO:DTT:TABL:LONG:SET2:STER -20`
enters a symbol timing error.

**Manual operation:** See "Symbol Timing Error" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:CFOFfset** <CfOffset>

Sets a carrier frequency offset.

The carrier frequency offset shows the deviation of the transmitted initial center frequency from carrier frequency.

**Parameters:**

<CfOffset> integer

Range: -150 to 150
*RST: 1
Default unit: kHz

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL SHOR`
enters the table type
`BB:BTO:DTT:TABL:SHOR:SET2:CFOF 65`
enters a carrier frequency offset.

**Manual operation:** See "Carrier Frequency Offset kHz" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:STATe** <State>

Activates or deactivates the corresponding parameter set in the short table. If a set deactivated, its parameters are skipped in the sequence. Instead, the next active set is used.

For EDR packets, each set applies to 20 packets.

**Parameters:**

<State> 0 | 1 | OFF | ON

*RST: 1

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL SHOR`
enters the table type
`BB:BTO:DTT:TABL:SHOR:SET2:STAT ON`
activates the set 2 in the short table.

**Manual operation:** See "State" on page 50

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TABLe:SHORt:SET<ch>:STERror** <StError>

Sets a symbol timing error in ppm.

The Symbol Timing Error modifies the symbol clock frequency by the set amount.

**Parameters:**

<StError> integer

Range: -150 to 150
Increment: 1
*RST: 1

**Example:** `BB:BTO:PTYP DH1`
sets the packet type.
`BB:BTO:DTT:TABL SHOR`
enters the table type
`BB:BTO:DTT:TABL:SHOR:SET2:STER 20`
enters a symbol timing error.

**Manual operation:** See "Symbol Timing Error" on page 50

## 4.9 LE Commands

**[:SOURce<hw>]:BB:BTOoth:BCRole** <BcRole>

Determines the controller role. Depending on the selected channel type different roles are assigned to the controller. For channel type "Data", master or slave can be assigned. If channel type "Advertising" is selected, the parameter is read only and displayed directly above the graph.

**Parameters:**

<BcRole> MASTer | SLAVe | ADVertiser | SCANner | INITiator

**MASTER**
Assigns master role to the controller.

**SLAVE**
Selects slave as controller role.

**ADVertiser|SCANner|INITiator**
Assigned roles depending on the selected packet type of the respective channel type.

*RST: MASTer

**Example:** `SOUR:BB:BTO:BCR MAST`
master as controller role.
`SOUR:BB:BTO:BCR SLAV`
slave as controller role.

**Manual operation:** See "Bluetooth Controller Role" on page 31

**[:SOURce<hw>]:BB:BTOoth:BMODe** <BMode>

Determines the digital Bluetooth standard. Basic Rate + EDR or Bluetooth Low Energy are available.

**Parameters:**

<BMode> BASic | BLENergy

**BASic**
Selects the Bluetooth mode Basic Rate + EDR.

**BLENergy**
Selects the Bluetooth mode Low Energy.

*RST: BASic

**Example:** `SOUR:BB:BTO:BMOD BLEN`
Bluetooth mode Low Energy.

**Manual operation:** See "Bluetooth Mode" on page 20

**[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:TPINterval** <TpInterval>

Sets the time interval between two consecutive test packets, with regard to the starting points.

**Parameters:**

<TpInterval> float

Range: 0.625E-3 to 12.5E-3
Increment: 0.625E-3
*RST: 0.625E-3

**Example:** `SOUR:BB:BTO:DTT:TPC:TPIN 1.0`
sets a time interval of 1.0.

**Manual operation:** See "Packet Interval" on page 47

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:UPLength** <UpLength>

Sets the payload length.

**Parameters:**

<UpLength> integer

Range: 1 to 37
*RST: 1

**Example:** `SOUR:BB:BTO:DTT:TPC:UPL 37`
sets a payload length of 37.

**Manual operation:** See "Payload Length" on page 47

---

**[:SOURce<hw>]:BB:BTOoth:DTTest:TPConfiguration:UPSource** <UpSource>

Selects the data source used for the payload test packets.

**Parameters:**

<UpSource> PN09 | PAT1 | PAT2 | PN15 | PAT3 | PAT4 | PAT5 | PAT6

**PN9 / PN15**
Select a PRBS-modulated data sequence
(PRBS = pseudo random binary sequence) for testing.

**PAT1 … PAT6**
Pattern is predefined.

*RST: PN09

**Example:** `SOUR:BB:BTO:DTT:TPC:UPS PN09`
PRBS-modulated data sequence for testing.

**Manual operation:** See "Payload Type" on page 47

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ACTable** <AcTable>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:DCTable** <DcTable>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DCMTable** <DcmTable>

The data channel map table indicates the entire data set of the channels in a table.

Every channel is represented with bit positioned as per the data channel index. LSB represents data channel index 0 and the bit in position 36 represents data channel index 36.

If the channel is used channel its bit is to be set to '1'. Bit value '0' indicates that the channel is unused.

The bits in positions 37, 38 and 39 shall be set to zero upon transmission and ignored upon receipt.

**Note:** This parameter is relevant for data event and advertising frame configuration with the packet types:

- CHANNEL_MAP_REQ
- CONNECT_REQ

**Parameters:**
<DcmTable> string

**Example:** `SOUR:BB:BTO:ECON:ACTable NOT`
Entire data set for Advertising Channel Table
`SOUR:BB:BTO:ECON:DCT NOT`
Entire data set for Data Channel Table
`SOUR:BB:BTO:ECON:PCON:DCMT NOT`
Entire data set for Channel Map Table

**Manual operation:** See "Channel Table" on page 37

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ACTable:CHANnel<ch0>:STATe** <State>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:DCTable:CHANnel<ch0>:STATe** <State>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DCMTable: CHANnel<ch0>:STATe** <State>

Indicates used and unused data channels.

**Note:** The previously used syntax `...:SET<ch>:STATe` has been replaced by `...:CHANnel<ch>:STATe`. Compatibility to the previous commands is given.

This parameter is relevant for data event and advertising frame configuration with the packet types:

- CHANNEL_MAP_REQ
- CONNECT_REQ

**Parameters:**
<State> 0 | 1 | OFF | ON
*RST: 0

**Example:** `SOUR:BB:BTO:ECON:ACT:CHAN:STAT ON`
State in Advertising Channel Table
`SOURce:BB:BTO:ECON:DCT:CHAN:STAT ON`
State in Data Channel Table
`SOUR:BB:BTO:ECON:PCON:DCMT:CHAN:STAT ON`
State in Data Channel Map Table

**Manual operation:** See "Channel Table" on page 37

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:AEDelay** <AeDelay>

Sets a time delay between the start times of two consecutive advertising events. The value is added to the advertising event interval.

**Parameters:**
<AeDelay> float

Range: 0 to 10E-3
Increment: 0.1E-3
*RST: 0

**Example:** `SOUR:BB:BTO:ECON:AED 10.0`
sets a time delay of 10.0 s.

**Manual operation:** See "Advertising Event Delay" on page 33

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:ADINterval** <AdInterval>

Sets the time interval between two consecutive advertising events for packet type "ADV_DIRECT_IND".

**Parameters:**
<AdInterval> float

Range: 1.05E-3 to 3.75E-3
Increment: 0.01E-3
*RST: 3.75E-3

**Example:** `SOUR:BB:BTO:ECON:ADIN 1.3`
sets a time interval of 1.3.

**Manual operation:** See "Advertising Event Interval" on page 33

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:AEINterval** <AeInterval>

Sets the time interval between two consecutive advertising events, with regard to the starting points.

**Parameters:**
<AeInterval> float

Range: 5E-3 to depends on oversampling
Increment: 0.1E-3
*RST: 20E-3

**Example:** `SOUR:BB:BTO:ECON:AEIN 20.0`
sets a time interval of 20.0.

**Manual operation:** See "Advertising Event Interval" on page 33

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:APINterval** <ApInterval>

Sets the time interval between packets starting points of two consecutive packets in the advertising channel.

**Parameters:**
<ApInterval> float

Range: 1.3E-3 to 10E-3
Increment: 0.1E-3
*RST: 10E-3

**Example:** `SOUR:BB:BTO:ECON:APIN 1.3`
sets a time interval of 1.3.

**Manual operation:** See "Advertising Packet Interval" on page 34

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LCMode** <LcMode>

Selects the link layer connection mode. In order to provide safe transmission of payload data, the data in the packet can be encrypted. If activated, the payload data follows MIC (Message authentication Code).

**Parameters:**
<LcMode> UENC | ENC

**UENC**
Payload data is transmitted without encoding.

**ENC**
The link layer connection runs in encrypted mode.

*RST: UENC

**Example:** `SOUR:BB:BTO:ECON:LCM UENC`
without encoding.
`SOUR:BB:BTO:ECON:LCM ENC`
in encrypted mode.

**Manual operation:** See "LL Connection Mode" on page 35

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:LTKey** <LtKey>

Indicates the time the controller needs to receive the long term key from the host. After this time, the controller is ready to enter into the last phase of encryption mode setup.

**Parameters:**
<LtKey> 128 bits

**Example:** `SOUR:BB:BTO:ECON:LCM ENC`
`SOUR:BB:BTO:ECON:LTK #H00000000000000000000000000000000,128`
In encrypted mode, the code can be edited.

**Manual operation:** See "Long Term key (hex)" on page 36

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:AADDress** <AAddress>

Sets the access address of the link layer connection (32-bit string).

**Parameters:**
<AAddress> 32 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:AADD #H00000000,32`
sets an access address.

**Manual operation:** See "Access Address" on page 38

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ACID** <Acid>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ACASsigned** <AcAssigned>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCASsigned** <ScAssigned>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCID** <Scid>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ICASsigned** <IcAssigned>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ICID** <Icid>

Sets the advertiser´s device address. In bluetooth low energy systems all the transmissions start with an 8 bit preamble followed by an access address. The access address is composed of a the parts "Company_Id" (LSB) and the "Company_assigned" (MSB). Beside the address fields the notation is given. For advertising channel packets the format of the device address differs, depending on the selected address type.

- "Public Address Types"
  The public address is given from the registration authority IEEE and is composed of:
  - LSB: 24 bits = company_assigned
  - MSB: 24 bits = company_id
- "Private Address Type"
  A private address is optional and composed of:
  - LSB: 24 bits = hash
  - MSB: 24 bits = random

**Parameters:**
<Icid> 24 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:ACID #H000000,24`
`SOUR:BB:BTO:ECON:PCON:ACAS #H000000,24`
Company_Assigned and Company_Id in Advertiser´s Device Address)
`SOUR:BB:BTO:ECON:PCON:SCAS #H000000,24`
`SOUR:BB:BTO:ECON:PCON:SCID #H000000,24`
Company_Assigned and Company_Id in Scanners Device Address)
`SOUR:BB:BTO:ECON:PCON:ICAS #H000000,24`
`SOUR:BB:BTO:ECON:PCON:ICID #H000000,24`
Company_Assigned and Company_Id in Initiators Device Address)

**Manual operation:** See "Controller´s Device Addr" on page 43

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CID** <Cid>

Sets the company identifier of the manufacturer of the Bluetooth Controller. A 16 bit value is set.

**Note:** This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

**Parameters:**

<Cid> 16 bits
*RST: 0

**Example:** `SOUR:BB:BTO:ECON:PCON:CID #H0000,16`
sets the company ID.

**Manual operation:** See "Company ID" on page 45

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINStant** <CInstant>

Sets a connection instant for indicating the connection event at which the new connection parameters are taken in use.

**Parameters:**

<CInstant> integer
Range: 1 to depends on sequence length
*RST: 1

**Example:** `SOUR:BB:BTO:ECON:PCON:CINS 2`
sets a connection instant.

**Manual operation:** See "Connection Instant" on page 41

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CINTerval**
<CInterval>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Set the time interval between the start points of two consecutive connection events.

**Parameters:**

<CInterval> float

Range: 7.5E-3 to depends on oversampling
Increment: 1.25E-3
*RST: 7.5E-3

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:CINT 7.5`
sets a time interval.

**Manual operation:** See "Connection Event Interval" on page 35

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:CIValue** <CiValue>

Sets the initialization value for the CRC (Cyclic Redundary Check, 24 bits) calculation. A packet has been received correctly, when it has passed the CRC check.

**Parameters:**

<CiValue> 24 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:CIV #H000000,24`
sets the initialization value for the CRC.

**Manual operation:** See "CRC Initial" on page 44

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA** <Data>

Selects the data source used for the payload.

**Parameters:**

<Data> ALL0 | ALL1 | PATTern | PN09 | PN11 | PN15 | PN16 | PN20 | PN21 | PN23 | DLISt

**All 0 / All 1**
0 data and 1 data is generated internally.

**Pattern**
Pattern is user definable.

**PN xx**
Pseudo-random noise sequence. XX can be equal to 9, 11, 15, 16, 20, 21, 23.

**Data List**
Internal data from a programmable data list is used. The data list can be generated by the Data List Editor or generated externally.

*RST: PN09

**Example:**

```
SOUR:BB:BTO:ECON:PCON:DATA ALL0 | ALL1
SOUR:BB:BTO:ECON:PCON:DATA PATT
SOUR:BB:BTO:ECON:PCON:DATA:DPAT #H3F,8
SOUR:BB:BTO:ECON:PCON:DATA PN09
SOUR:BB:BTO:ECON:PCON:DATA DLIS
SOUR:BB:BTO:ECON:PCON:DATA:DSEL 'bluetooth-le'
```

**Manual operation:** See "Data Source" on page 40

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA:DPATtern**
<DPattern>

Selects the data for a pattern.

**Parameters:**

<DPattern> 64 bits

*RST: #H0,1

**Example:**

```
BB:BTO:ECON:PCON:DATA PATT
```
sets the data type.
```
BB:BTO:ECON:PCON:DATA:DPAT #B010101, 6
```
selects the data for a pattern

**Manual operation:** See "Data Source" on page 40

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DATA:DSELection**
<DSelection>

Selects data list file.

**Parameters:**

<DSelection> string

**Example:** `BB:BTO:ECON:PCON:DATA DLIS`
selects the data type.
`BB:BTO:ECON:PCON:DSEL Bluetooth-le`
selects the file for the data.

**Manual operation:** See "Data Source" on page 40

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DLENgth**
<DLength>

Sets the payload data length in bytes.

**Parameters:**
<DLength> integer
Range: 0 to depends on packet type
*RST: 31

**Example:** `SOUR:BB:BTO:ECON:PCON:DLEN 31`
payload data length is 31 bytes.

**Manual operation:** See "Data Length" on page 40

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:DWHitening**
<DWhitening>

Activates or deactivates the Data Whitening. Evenly distributed white noise is ideal for the transmission and real data can be forced to look similar to white noise with different methods called Data Whitening. Applied to the PDU and CRC fields of all packet types, whitening is used to avoid long equal seqeunces in the data bit stream.

**Parameters:**
<DWhitening> 0 | 1 | OFF | ON
*RST: 0

**Example:** `BB:BTO:ECON:PCON:DWH ON`
activates data whitening.

**Manual operation:** See "Data Whitening" on page 38

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:ECODe** <ECode>

Sets the error code value to inform the remote device why the connection is about to be terminated in case of LL_TERMINATE_IND packet. On the other hand, this parameter for LL_REJECT_IND packet is used for the reason a request was rejected. A 8 bit value is set.

**Note:** This parameter is relevant for data frame configuration and the packet type:

- LL_TERMINATE_IND
- LL_REJECT_IND

**Parameters:**
<ECode> 8 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:ECOD #H00,8`
sets the error code

**Manual operation:** See "Error Code" on page 45

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:EDIVersifier**
<EDiversifier>

Sets the encrypted diversifier of the master for device identification. The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

**Parameters:**
<EDiversifier> 16 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:EDIV #H0000, 16`
sets the encrypted diversifier of the master.

**Manual operation:** See "Encrypted DIVersifier (hex)" on page 42

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:FSLength**
<FsLength>

Enables that the feature set length is indicated. FeatureSet indicates whether the Controller features are used or not. All the data in FeatureSet is RFU(zero).

**Parameters:**
<FsLength> integer

Range: 1 to 26
*RST: 26

**Example:** `SOUR:BB:BTO:ECON:PCON:FSL 12`
feature set length is 12.

**Manual operation:** See "Feature Set Length" on page 43

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:HLENgth**
<HLength>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the difference from the current channel to the next channel. The master and slave devices determine the data channel in use for every connection event from the channel map. Hop_length is set for the LL connection and communicated in the CONNECT_REQ and CHANNEL_MAP_REQ packets.

**Parameters:**
<HLength> integer

Range: 5 to 16
*RST: 5

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:HLEN 10`
hop length is 10.

**Manual operation:** See "Hop Length" on page 42

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:LCTimeout**
<LcTimeout>

(for data event and advertising frame configuration with the packet type:

CONNECT_REQ

Defines the maximum time between two correctly received Bluetooth Low Energy packets in the LL connection before the connection is considered lost.

**Parameters:**
<LcTimeout> float

Range: 100E-3 to 32000E-3
Increment: 10E-3
*RST: 100E-3

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:LCT 150`
LL Connection Timeout is 150.

**Manual operation:** See "LL Connection Timeout" on page 41

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:MIVector**
<MiVector>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SIVector**
<SiVector>

Sets the master's or the slave's portion of the initialization vector(IVm/IVs).

**Parameters:**
<SiVector> 32 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:MIV #H0000000000000000,32`
(Master).
`SOUR:BB:BTO:ECON:PCON:SIV #H0000000000000000,32`
(Slave).

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:MSKD** <Mskd>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SSKD** <Sskd>

Sets the master's or the slave's portion of the session key diversifier (SKDm/SKDs).

**Parameters:**
<Sskd> 64 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:MSKD #H0000000000000000,64`
(Master).
`SOUR:BB:BTO:ECON:PCON:SSKD #H0000000000000000,64`
(Slave).

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NCINterval**
<NcInterval>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the time interval between the start points of two consecutive connection events.

**Parameters:**
<NcInterval> float

Range: 7.5E-3 to depends on oversampling
Increment: 1.25E-3
*RST: 7.5E-3

**Example:** `SOUR:BB:BTO:UPTY CUR`
sets packet type CONNECTION_UPDATE_REQ
`SOUR:BB:BTO:ECON:PCON:NCIN 7.5`
sets a time interval.

**Manual operation:** See "Connection Event Interval" on page 35

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NLCTimeout**
<NlcTimeout>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Defines the maximum time between two correctly received Bluetooth Low Energy packets in the LL connection before the connection is considered lost.

**Parameters:**
<NlcTimeout> float

Range: 100E-3 to 32000E-3
Increment: 10E-3
*RST: 100E-3

**Example:** `SOUR:BB:BTO:UPTY CUR`
sets packet type CONNECTION_UPDATE_REQ
`SOUR:BB:BTO:ECON:PCON:NLCT 150`
LL Connection Timeout is 150.

**Manual operation:** See "LL Connection Timeout" on page 41

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSLatency**
<NsLatency>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

**Parameters:**

<NsLatency> integer

Range: 0 to depends on LL connection timeout and connection event interval
*RST: 0

**Example:** `SOUR:BB:BTO:UPTY CUR`
sets packet type CONNECTION_UPDATE_REQ
`SOUR:BB:BTO:ECON:PCON:NSL 10`
sets the number of consecutive connection events.

**Manual operation:** See "Slave Latency" on page 41

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NSValue**
<NsValue>

Sets the start value of the next expected packet from the same device in the LL connection ("N"ext"E"xpected "S"equence"N"umber). This parameter can be set in the first event. From the second event this field is not indicated.

**Parameters:**

<NsValue> integer

Range: 0 to 1
*RST: 1

**Example:** `SOUR:BB:BTO:ECON:PCON:NSV 1`
Start Value is 1.

**Manual operation:** See "NESN Start Value" on page 39

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NWOFfset**
<NwOffset>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the start point of the transmit window.

**Parameters:**

<NwOffset> float

Range: 0 to depends on connection event interval
Increment: 1.25E-3
*RST: 0

**Example:** SOUR:BB:BTO:UPTY CUR
sets packet type CONNECTION_UPDATE_REQ
SOUR:BB:BTO:ECON:PCON:NWOF 800.0
sets the start point of the transmit window.

**Manual operation:** See "Transmit Window Offset" on page 45

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:NWSize** <NwSize>

(for data event and advertising frame configuration with the packet type CONNECTION_UPDATE_REQ)

Sets the size of the transmit window, regarding to the start point.

**Parameters:**
<NwSize> float

Range: 1.25E-3 to depends on connection event interval
Increment: 1.25E-3
*RST: 1.25E-3

**Example:** SOUR:BB:BTO:UPTY CUR
sets packet type CONNECTION_UPDATE_REQ
SOUR:BB:BTO:ECON:PCON:NWS 8.0
sets the size of the transmit window.

**Manual operation:** See "Transmit Window Size" on page 44

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:TATYpe** <TaType>
**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:RATYpe** <RaType>

Selects the address type of the controller device. Depending on the Bluetooth controller role either Tx or Rx or both address types are assigned. Subdivided into private and random, a Bluetooth LE device address consits of 48 bits. The format of the device address differs depending on the selected address type.

**Parameters:**
<RaType> PUBLic | RANDom

**PUBlic**
Allocates a unique 48 bit address to each bluetooth LE device. The public address is given from the registration authority IEEE.

**RANDom**
Allocates a 48 bit address to each bluetooth LE device. A random address is optional.

*RST: PUBLic

**Example:** SOUR:BB:BTO:ECON:PCON:TATY PUBL
SOUR:BB:BTO:ECON:PCON:RATY RAND

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:RVECtor** <RVector>

Sets the random vector of the master for device identification.The parameter is an initialization vector provided by the Host in the HCI_ULP_Start_Encryption command.

**Parameters:**

<RVector> 64 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:RVEC #H0000000000000000,64`
sets the random vector of the master.

**Manual operation:** See "Random Vector (hex)" on page 42

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SCACcuracy** <ScAccuracy>

Defines the master´s clock accuracy with specified encoding. This parameter is used by the slave to determine required listening windows in the LL connection. It is a controller design parameter known by the Controller.

**Parameters:**

<ScAccuracy> SCA0 | SCA1 | SCA2 | SCA3 | SCA4 | SCA5 | SCA6 | SCA7
*RST: SCA0

**Example:** `SOUR:BB:BTO:ECON:PCON:SCAC SCA1`
sets the encoding value

**Manual operation:** See "Sleep Clock Accuracy" on page 45

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SDCI?**

Queries the number of the first active data channel.

**Return values:**

<SelectedChannel> integer
Range: 0 to 36
*RST: 0

**Example:** `SOURce1:BB:BTOoth:ECONfiguration:SDCI?`

**Usage:** Query only

**Manual operation:** See "Selected Data Channel Index" on page 36

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SLATency** <SLatency>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets a number of consecutive connection events the slave can ignore for asymmetric link layer connections.

**Parameters:**

<SLatency> integer

Range: 0 to depends on LL connection timeout and connection event interval

*RST: depends on LL connection timeout and connection event interval

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:SLAT 10`
sets the number of consecutive connection events.

**Manual operation:** See "Slave Latency" on page 41

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SSValue** <SsValue>

Sets the sequence number of the packet. This parameter can be set in the first event. From the second event this field is not indicated.

**Parameters:**

<SsValue> integer

Range: 0 to 1

*RST: 0

**Example:** `SOUR:BB:BTO:ECON:PCON:SSV 1`
sets the sequence number of the packet.

**Manual operation:** See "SN Start Value" on page 39

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:SVNumber** <SvNumber>

Sets a unique value for each implementation or revision of an implementation of the Bluetooth Controller. A 16 bit value is set.

**Note:** This parameter is relevant for data frame configuration and for the packet type:

LL_VERSION_IND.

**Parameters:**

<SvNumber> 16 bits

*RST: 0

**Example:** `SOUR:BB:BTO:ECON:PCON:SVN #H0000,16`
sets the sub version number

**Manual operation:** See "Sub Version Number" on page 46

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:UTYPe** <UType>

Enables that an invalid control packet is indicated. The CtrType field indicates the value of the LL control packet that caused the transmission of this packet.

**Parameters:**
<UType> 8 bits

**Example:** `SOUR:BB:BTO:ECON:PCON:UTYP #H8, 00`
enables that an invalid control packet is indicated.

**Manual operation:** See "Unknown Type (hex)" on page 43

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:VNUMber** <VNumber>

Sets the company identifier of the manufacturer of the Bluetooth Controller. A 8 bit value is set.

**Note:** This parameter is relevant for data frame configuration and for the packet type LL_VERSION_IND.

**Parameters:**
<VNumber> 8 bits
*RST: 0

**Example:** `SOUR:BB:BTO:ECON:PCON:VNUM #H00,8`
sets the version number

**Manual operation:** See "Version Number" on page 46

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:WOFFset** <WOffset>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the start point of the transmit window.

**Parameters:**
<WOffset> float
Range: 0 to depends on connection event interval
Increment: 1.25E-3
*RST: 0

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:WOFF 800.0`
sets the start point of the transmit window.

**Manual operation:** See "Transmit Window Offset" on page 45

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PCONfiguration:WSIZe** <WSize>

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Sets the size of the transmit window, regarding to the start point.

**Parameters:**

<WSize> float

Range: 1.25E-3 to depends on connection event interval
Increment: 1.25E-3
*RST: 1.25E-3

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:WSIZ 8.0`
sets the size of the transmit window.

**Manual operation:** See "Transmit Window Size" on page 44

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:PNUMber** <PNumber>

Sets the number of Tx packets per event. Each connection contains at least one data channel packet. The maximum number of packets per event is determined by the duration of the connection event interval.

**Parameters:**

<PNumber> integer

Range: 1 to depends on connection event interval
*RST: 1

**Example:** `SOUR:BB:BTO:ECON:PNUM 2580`
sets the number of Tx packets per event.

**Manual operation:** See "No. of Tx Packets per Event" on page 35

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SINTerval** <SInterval>

Sets the time interval between the starting points of two consecutive windows during which the scanner is operating in an advertising channel.

**Parameters:**

<SInterval> float

Range: 10E-3 to depends on oversampling and the number of advertsing channel table states
Increment: 0.625E-3
*RST: 10E-3

**Example:** `SOUR:BB:BTO:ECON:SINT 3.5`
sets the time interval.

**Manual operation:** See "Scan Interval" on page 34

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:SWINdow** <SWindow>

Sets the length of the window during which the scanner is operating in the advertising channel. Note that the scan window is less or equal to the value of the scan interval.

**Parameters:**

<SWindow> float

Range: 10E-3 to 10240E-3
Increment: 0.625E-3
*RST: 10E-3

**Example:** `SOUR:BB:BTO:ECON:SWIN 2.5`
sets the length of the window.

**Manual operation:** See "Scan Window" on page 33

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WOINfo?**

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Queries the start point of the transmit window.

**Return values:**

<WoInfo> string

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:WOIN?`
queries the start point of the transmit window.

**Usage:** Query only

**Manual operation:** See "Transmit Window Offset" on page 34

---

**[:SOURce<hw>]:BB:BTOoth:ECONfiguration:WSINfo?**

(for data event and advertising frame configuration with the packet type CONNECT_REQ)

Queries the size of the transmit window, regarding to the start point.

**Return values:**

<WsInfo> string

**Example:** `SOUR:BB:BTO:UPTY CREQ`
sets packet type CONNECT_REQ
`SOUR:BB:BTO:ECON:PCON:WSIN?`
queries the size of the transmit window.

**Usage:** Query only

**Manual operation:** See "Transmit Window Size" on page 34

---

**[:SOURce<hw>]:BB:BTOoth:UPTYpe** <UpType>

Selects the packet type. The available packets depend on the selected channel type.

**Parameters:**

<UpType> AIND | ADINd | ANINd | SREQ | SRSP | CREQ | ADCind | DATA | CUReq | CMReq | TIND | EREQ | ERSP | SEReq | SERSp | URSP | FREQ | FRSP | TPACket | PEReq | PERSp | VIND | RIND

*RST: AIND

**Example:**

```
SOUR:BB:BTO:UPTY AIND
SOUR:BB:BTO:UPTY ADIN
SOUR:BB:BTO:UPTY ANIN
SOUR:BB:BTO:UPTY ADC
SOUR:BB:BTO:UPTY SREQ
SOUR:BB:BTO:UPTY SRSP
SOUR:BB:BTO:UPTY CREQ
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Advertising"

```
SOUR:BB:BTO:UPTY DATA
SOUR:BB:BTO:UPTY CUR
SOUR:BB:BTO:UPTY CMR
SOUR:BB:BTO:UPTY TIND
SOUR:BB:BTO:UPTY LER
SOUR:BB:BTO:UPTY SERS
SOUR:BB:BTO:UPTY FREQ
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Data" and Controller role "Master".

```
SOUR:BB:BTO:UPTY DATA
SOUR:BB:BTO:UPTY TIND
SOUR:BB:BTO:UPTY LERS
SOUR:BB:BTO:UPTY SER
SOUR:BB:BTO:UPTY SERS
SOUR:BB:BTO:UPTY URSP
SOUR:BB:BTO:UPTY FRSP
SOUR:BB:BTO:UPTY TPAC
```

Channel type "Data" and Controller role "Slave".

**Manual operation:** See "Packet Type" on page 29

---

**[:SOURce<hw>]:BB:BTOoth:USLength** <UsLength>

Selects the number of frames or events depending on the packet type. The signal repeats after the specified number of frames/events.

For SCAN_REQ and CONNECT_REQ packet, the sequence length is expressed in "Frames".

For TERMINATE_IND packets, a default value according to the specification is given:

Master: 'SlaveLatency + 6'

Slave: '6'

For all other packet types the sequence length is expressed in "Events".

**Parameters:**

<UsLength> integer

Range: depending on the number of states in dirty transmitter test to dynamics

*RST: 1

**Example:** `SOUR:BB:BTO:USL 1000`
selects the number of frames or events.

**Manual operation:** See "Sequence Length" on page 30

# List of Commands

# Index

## F



## G



## H



## I



## L



## M



## N



## O



## P