# Log File Generation
# R&S®SMW-K81
# User Manual


User Manual

Version 03

ROHDE&SCHWARZ

This document describes the following software options:

- R&S®SMW-K81
  1413.4539.0x

This manual describes firmware version FW 4.70.026.xx and later of the R&S®SMW200A.

# Contents

# 1 Preface

## 1.1 Documentation Overview

This section provides an overview of the R&S SMW user documentation. Unless specified otherwise, you find the documents on the R&S SMW product page at:

www.rohde-schwarz.com/manual/smw200a

### 1.1.1 Getting Started Manual

Introduces the R&S SMW 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. A printed version is delivered with the instrument.

### 1.1.2 User Manuals and Help

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

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

The contents of the user manuals are available as help in the R&S SMW. The help offers quick, context-sensitive access to the complete information for the base unit and the software options.

All user manuals are also available for download or for immediate display on the Internet.

### 1.1.3 Tutorials

The R&S SMW provides interactive examples and demonstrations on operating the instrument in form of tutorials. A set of tutorials is available directly on the instrument.

### 1.1.4 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, https://gloris.rohde-schwarz.com).

### 1.1.5 Instrument Security Procedures

Deals with security issues when working with the R&S SMW in secure areas. It is available for download on the Internet.

### 1.1.6 Basic Safety Instructions

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

### 1.1.7 Data Sheets and Brochures

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

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

See www.rohde-schwarz.com/brochure-datasheet/smw200a

### 1.1.8 Release Notes and Open Source Acknowledgment (OSA)

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

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

See www.rohde-schwarz.com/firmware/smw200a

### 1.1.9 Application Notes, Application Cards, White Papers, etc.

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

See www.rohde-schwarz.com/application/smw200a and www.rohde-schwarz.com/manual/smw200a

# 2 Generating Logfiles for Design Cross-Verification

If equipped with the option R&S SMW-K81, your R&S SMW can create logfiles for exchanging intermediate results of different logging points in the signal processing chain.

Analyzing the content of the logfiles can help you to verify the signal processing chain in both the DL and UL direction. The intermediate results provide a basis for enhanced debugging. To verify the FEC implementation of the DUT for instance, you can load the coded bitstream from the instrument into an Rx software module for offline analysis in a simulation environment. You can also compare the coded stream to the bitstreams from a Tx software module. The logfiles generation functionality can also be remote controlled, so that the design flow can be optimized and the process automated.

# 3 LTE/IoT Logfile Generation

## 3.1 Required Options

The generation of logfiles requires:

- Standard or wideband baseband generator (R&S SMW-B10/-B9)
- Option EUTRA/LTE (R&S SMW-K55)
- Option Cellular IoT (R&S SMW-K115)
- Option log files generation (R&S SMW-K81)

  Two options R&S SMW-K81 are required in the fowlloing cases:
  - For generating logfiles for more than one transmission antenna simultaneously.
  - If coupled baseband sources are used.

## 3.2 Output Files

The instrument stores the output logfiles in a user-defined network directory, selected with the parameter Output Path. The logfiles are named according to the naming conventions described in Chapter 3.2.1, "Filenames", on page 8. Description of the available file formats is listed in "File formats" on page 8.

**File formats**

Generally, the logfiles are generated in two file formats:

- Bitstream
  The logfile contains a sequence of "1" and "0"; one value per line
  The logfile of the PHICH contains also the entry "-" that corresponds to DTX.
- IQ samples
  The logfile contains pairs of I and Q samples; the I and Q components alternate at each line

File format "IQ samples" is used for the logfiles generated for the logging points after "Modulation Mapping". The other logfiles are output in a Bitstream format.

Exceptions are the extended DCI/UCI logfiles, and the summary logfile (see Chapter 3.2.2.1, "Extended DCI Logfile", on page 13 and Chapter 3.2.2.2, "Extended UCI Logfile", on page 16).

### 3.2.1 Filenames

The generated logging files are named according to the following naming structure:

```
[<Preamble>_]<Frame#>_<Subframe#>|<TRANSM#>_<Channel>[-<Format>]
[_<User/Allocation#>|<DCI#>|<Group#>]_<Point#>[_<CW#>|<LAY#>|
<ANT#>][_<RV#>]_<PointName>[_<CodeBlock#>].dat
```

Exceptions are the extended DCI/UCI logfiles, and the summary logfile. The filenames of these logfiles are as follows:

- `[<Preamble>_]ExtendedDciLog_<BB#>.txt`
- `[<Preamble>_]ExtendedUciLog_<BB#>.txt`
- `[<Preamble>_]SummaryLogfile_<BB#>.txt`

***Table 3-1: Filename structure***

| | **Description** | **Value range** |
|---|---|---|
| `<Preamble>` | Preamble with default syntax<br>`EUtraLog_<Entity#>` | Entity#: 0 to 7 |
| `<Frame#>` | Frame number | F000 to F873<br>NPBCH*): F000, F064, F128, etc. |
| `<Subframe#>` | Subframe number<br>Starting subframe number (NPBCH/ NPDSCH/NPDCCH) | SF0 to SF9 |
| `<TRANSM#>` | eMTC/NB-IoT transmission | TRANSM01 to TRANSM20 |
| `<Channel>` | Channel name | DL: PBCH \| PCFICH \| PHICH \| PDCCH \| PDSCH \| PMCH<br>DL: NPBCH \| NPDSCH \| NPDCCH \| NPDSCH-SIB1 \| PDSCH-SIB1BR |
| | | UL: PUSCH \| PUCCH PUSCHDRS \| PUCCHDRS \| SRS<br>UL: NPUSCH |
| `<User>` | PDSCH, PUSCH, PUCCH, PUSCH DRS, PUCCH DRS<br>NPUSCH, NPUSCH DRS | USER1 to USER4 |
| `<Allocation#>` | PDSCH allocation only | ALL000 to ALL101 |
| `<DCI#>` | PDCCH allocation only<br>Each PDCCH DCI is logged individu- ally | DCI00 to DCI19 |
| `<Group#>` | PHICH group<br>An individual file is generated for each PHICH group | Group00 \| Group01 |
| `<Format>` | PUCCH format | F1 \| F1A \| F1B \| F2 \| F2A \| F2B \| F3 \| F4 \| F5 |
| `<Point#>` | Logging point number | See Table 3-2. |
| `<CW#>` | PDSCH, PUSCH/NPUSCH allocations only<br>Codeword | CW0 \| CW1 (PDSCH, PUSCH)<br>CW0 (NPUSCH, PDSCH-SIB1BR) |

| | Description | Value range |
|---|---|---|
| <LAY#> | PDSCH, PUSCH, and PUSCH DRS allocations only<br>PDSCH-SIB1BR, NPBCH, NPDSCH, NPUSCH and NPUSCH DRS allocations only<br>Layer number | DL: LAY0 to LAY7<br>DL: LAY0 to LAY1 (NPBCH/NPDCCH/ NPDSCH/NPDSCH-SIB1/PDSCH-SIB1BR)<br>UL: LAY0 to LAY3 (PUSCH and PSUCH DRS)<br>UL: LAY0 (NPUSCH and NPUSCH DRS) |
| <ANT#><br><AP#> | Antenna port number | DL: ANT1 to ANT4<br>DL NB-IoT: ANT1\|ANT2<br>UL: AP10 \| AP100 \| AP20 \| AP21 \| AP40 \| AP41 \| AP42 \| AP43 \| AP200 \| AP201 |
| <RV#> | NPUSCH allocations only<br>Redundancy version | RV00 to RV02 |
| <PointName> | Logging point designation | See Table 3-2 |
| <CodeBlock#> | PDSCH and PUSCH allocations only | CB00 to CB20 |
| <BB#> | Baseband | BBA to BBH |

*) NPBCH lasts 640 ms. One logfile contains 64 frames, starting from the frame number indicated as <Frame#>.

There is a fixed cross-reference between the logging point number and the logging point designation:

- See Table 3-2.
- The PUCCH logging points depend on the PUCCH format, see Table 3-3.
- See Table 3-4.

***Table 3-2: Logging points overview (DL and PUSCH/NPUSCH)***

| <Point#> | <PointName> | DL *) | PUSCH | NPUSCH | Description |
|---|---|---|---|---|---|
| PT00 | TB | X | X | X | Bits of the transport block |
| PT01 | TBCRC | X | X | X | Bits after transport block CRC |
| PT02 | CBCRC | X | X | - | Bits after code block CRC<br>One file per code block is generated |
| PT03 | | | | | Bits after channel coding (one file per code block) |
| | CCSys | X | X | X | Systematic bits |
| | CCPar1 | X | X | X | Parity 1 bits |
| | CCPar2 | X | X | X | Parity 2 bits |
| | CCTotal | X | X | X | (N)PDSCH and PUSCH allocation only<br>Complete bitstream after channel coding, incl. systematic, parity 1 and parity 2 bits |
| PT04 | RM | X | X | X | Bits after rate matcher (one file per code block) |

| <Point#> | <PointName> | DL *) | PUSCH | NPUSCH | Description |
|---|---|---|---|---|---|
| PT05 | CBCON | X | X | - | Bits after code block concatenation |
| PT06 | DL: SCR | X | - | - | Bits after scrambling |
| | UL: MUX | - | X | - | Bits after data and control multiplexing |
| PT07 | DL: MOD | X | - | - | IQ-Samples after modulation |
| | UL: CHI | - | X | X | Bits after channel interleaver |
| PT08 | DL: MAP | X | - | - | IQ-Samples after layer mapping (one file per layer) |
| | UL: SCR | - | X | X | Bits after Scrambling |
| PT09 | DL: PREC | X | - | - | IQ-Samples after precoding (one file per antenna) |
| | UL: MOD | - | X | X | IQ-Samples after modulation |
| PT10 | UL: MAP | - | X | - | IQ-Samples after layer mapping (one file per layer) |
| PT11 | UL: DFT | - | X | X | IQ-Samples after DFT |
| PT12 | UL: PREC | - | X | - | IQ-Samples after precoding (one file per antenna) |

*) PT02 and PT05 not available for NPBCH, NPDSCH and NPDCCH

***Table 3-3: PUCCH logging points overview per PUCCH format***

| PUCCH format | <Point#> | <PointName> | Description |
|---|---|---|---|
| F1/F1a/F1b | PT00 | SCR-BLOCK-WISE-SPREAD | Bits after scrambled block-wise spread operation |
| F2/F2a/F2b | PT00 | UNCODED | Uncoded bits |
| | PT01 | SCR | Scrambled bits |
| | PT02 | CYCLIC-SHIF-TED | Bits after cyclic-shift operation |
| F3 | PT00 | UNCODED | Uncoded bits |
| | PT01 | CODED | Coded bits |
| | PT02 | SCR | Scrambled bits |
| | PT03 | MOD | IQ-Samples after modulation |
| | PT04 | BLOCK-WISE_SPREAD | Bits after block-wise spread operation |
| | PT05 | CYCLIC-SHIF-TED | Bits after cyclic-shift operation |
| | PT06 | DFT-PREC | IQ-Samples after DFT transform precoding |
| F4/F5 | PT00 | UNCODED | Uncoded bits |
| | PT01 | CRC | Bits after block CRC |

| PUCCH format | <Point#> | <PointName> | Description |
|---|---|---|---|
| | PT02 | CCSys<br>CCPar1/CCPar2 | Systematic bits<br>Parity 1 bits/Parity 2 bits |
| | PT03 | RM | Bits after rate matcher |
| | PT04 | SCR | Scrambled bits |
| | PT05 | MOD | IQ-Samples after modulation |

*Table 3-4: PUCCHDRS and PUSCHDRS logging points overview*

| DRS | <Point#> | <PointName> | Description |
|---|---|---|---|
| PUCCHDRS | PT00 | CYCLIC-SHIF-TED | Bits after cyclic-shift operation |
| PUSCHDRS | PT00 | CAZAC | IQ-Samples after CAZAC sequence generation |
| | PT01 | OCC | IQ-Samples of OCC (orthogonal cover code) sequence |
| | PT02 | PREC | IQ-Samples after precoding |

### Example: List of the output logfiles for PDSCH

The following output files are generated for one PDSCH channel, configured on an allocation with index `ALL002` in the third subframe (`SF2`) of the first frame (`F000`). The instrument is configured to generate a MIMO signal with two antennas (`PREC_ANT1` and `PREC_ANT2`). Channel coding and scrambling are enabled (`CCPar1`, `CCPar2`, `CCSys`, `CCTotal` and `SCR`). Two codewords (`CW0` and `CW1`) and two layers (`LAY0` and `LAY1`) are used; three code blocks per code (`CB00`, `CB01`, `CB02`) are generated.

All logging points are enabled and a preamble (`EUtraLog_0`) is selected.

```
<User/
Allocation#>_<Point#>[_<CW#>|<LAY#>|<ANT#>]_<PointName>[_<CodeBlock#>].dat

EUtraLog_0_F000_SF2_PDSCH_ALL002_PT00_CW0_TB.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT00_CW1_TB.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT01_CW0_TBCRC.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT01_CW1_TBCRC.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW0_CBCRC_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW1_CBCRC_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW0_CBCRC_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW1_CBCRC_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW0_CBCRC_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT02_CW1_CBCRC_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar1_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar1_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar1_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar1_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar1_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar1_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar2_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar2_CB00.dat
```

```
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar2_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar2_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCPar2_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCPar2_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCSys_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCSys_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCSys_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCSys_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCSys_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCSys_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCTotal_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCTotal_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCTotal_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCTotal_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW0_CCTotal_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT03_CW1_CCTotal_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW0_RM_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW1_RM_CB00.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW0_RM_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW1_RM_CB01.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW0_RM_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT04_CW1_RM_CB02.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT05_CW0_CBCON.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT05_CW1_CBCON.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT06_CW0_SCR.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT06_CW1_SCR.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT07_CW0_MOD.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT07_CW1_MOD.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT08_LAY0_MAP.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT08_LAY1_MAP.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT09_ANT1_PREC.dat
EUtraLog_0_F000_SF2_PDSCH_ALL002_PT09_ANT2_PREC.dat
```

### 3.2.2 Extended Logfiles Contents

The instrument generates only one logfile with extended information regarding the DCI/UCI mapping.

DCI mapping to the NB-IoT channels in not logged.

#### 3.2.2.1 Extended DCI Logfile

An extended DCI logfile summarizes the information for the whole generated signal. It can contain information for more than one frame. The information is grouped in rows with different syntax.

- PCFICH mapping, i.e the resource elements the PCFICH REGs are mapped to

```
<Frame#>_<Subframe#>_
PCFICH: REG-Idx=<REG#>: Subcarrier=<Subcarrier#>,
Symbol=<OFDMSymbol#>
```

- PHICH mapping, i.e the resource elements the PHICH REGs of the individual PHICH groups are mapped to
  `<Frame#>_<Subframe#>_ PHICH: Group=<Group#>: REG-Idx=<REG#>: Subcarrier=<Subcarrier#> , Symbol=<OFDMSymbol#>`
- PDCCH number of useful REGs
  `<Frame#>_<Subframe#>_PDCCH: Columns:<Columns#>, Rows:<Rows#>, Useful REGs:<REG#>`
  The number of useful REGs corresponds to the value displayed with the parameter "LTE > PDCCH > Number of PDCCH REGs".
- The start CCE-Index of the individual DCIs
  `<Frame#>_<Subframe#>_DCI: DCI Idx=<DCI#>, Start CCE-Idx=<CCE#>`
  The DCI Idx corresponds to the row in the DCI table and the start CCE-Idx is the "CCE Index" for the corresponding DCI.
- PDCCH mapping, i.e the resource elements the PDCCH REGs of the individual PDCCHs are mapped to.
  `<Frame#>_<Subframe#>_ PDCCH: Idx=<Symbol#>: REG-Idx=<REG#>: Subcarrier=<Subcarrier#> , Symbol=<OFDMSymbol#> [--- DTX REG]`
  The additional information DTX REG is assigned to all dummy PDCCH REGs.

Subcarrier with index 0 is the most left subcarrier, i.e. the one belonging to the resource block 0.

**Example: Content of an extended DCI logfile**

The instrument is configured to generate a DL LTE signal with the following configuration:

- 1.4 MHz bandwidth (6 RBs)
- Normal cyclic prefix
- Extended PHICH duration
- Control region for PDCCH of 3 OFDM symbols
- Two antennas, path A generates the signal of antenna 1 and path B, the signal of antenna 2
- PDCCH format variable
- PDCCH is configured as given on the figure bellow.

| | User | UE_ID n_RNTI | Cell Index | DCI Format | Search Space | Content | PDCCH Format | Number CCEs | CCE Index | No.Dummy CCEs | Conflict |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | User1 | 0 | 0 | 0 | Off | Config... | 1 | 2 | 0 | 0 | |
| 1 | SI-RNTI | 65535 | 0 | 1A | Off | Config... | 0 | 1 | 2 | 1 | |

Generation of extended DCI logfile is enabled and the file contains the following information (only the beginning of the file is listed):

```
F00,SF0,PCFICH: REG-Idx=0: Subcarrier=1, Symbol=0
F00,SF0,PCFICH: REG-Idx=0: Subcarrier=2, Symbol=0
F00,SF0,PCFICH: REG-Idx=0: Subcarrier=4, Symbol=0
F00,SF0,PCFICH: REG-Idx=0: Subcarrier=5, Symbol=0
F00,SF0,PCFICH: REG-Idx=1: Subcarrier=19, Symbol=0
F00,SF0,PCFICH: REG-Idx=1: Subcarrier=20, Symbol=0
F00,SF0,PCFICH: REG-Idx=1: Subcarrier=22, Symbol=0
F00,SF0,PCFICH: REG-Idx=1: Subcarrier=23, Symbol=0
F00,SF0,PCFICH: REG-Idx=2: Subcarrier=37, Symbol=0
F00,SF0,PCFICH: REG-Idx=2: Subcarrier=38, Symbol=0
F00,SF0,PCFICH: REG-Idx=2: Subcarrier=40, Symbol=0
F00,SF0,PCFICH: REG-Idx=2: Subcarrier=41, Symbol=0
F00,SF0,PCFICH: REG-Idx=3: Subcarrier=55, Symbol=0
F00,SF0,PCFICH: REG-Idx=3: Subcarrier=56, Symbol=0
F00,SF0,PCFICH: REG-Idx=3: Subcarrier=58, Symbol=0
F00,SF0,PCFICH: REG-Idx=3: Subcarrier=59, Symbol=0

F00,SF0,PHICH: Group=0: REG-Idx=0: Subcarrier=7, Symbol=0
F00,SF0,PHICH: Group=0: REG-Idx=0: Subcarrier=8, Symbol=0
F00,SF0,PHICH: Group=0: REG-Idx=0: Subcarrier=10, Symbol=0
F00,SF0,PHICH: Group=0: REG-Idx=0: Subcarrier=11, Symbol=0
F00,SF0,PHICH: Group=0: REG-Idx=1: Subcarrier=24, Symbol=1
F00,SF0,PHICH: Group=0: REG-Idx=1: Subcarrier=25, Symbol=1
F00,SF0,PHICH: Group=0: REG-Idx=1: Subcarrier=26, Symbol=1
F00,SF0,PHICH: Group=0: REG-Idx=1: Subcarrier=27, Symbol=1
F00,SF0,PHICH: Group=0: REG-Idx=2: Subcarrier=48, Symbol=2
F00,SF0,PHICH: Group=0: REG-Idx=2: Subcarrier=49, Symbol=2
F00,SF0,PHICH: Group=0: REG-Idx=2: Subcarrier=50, Symbol=2
F00,SF0,PHICH: Group=0: REG-Idx=2: Subcarrier=51, Symbol=2

F00,SF0,PDCCH: Columns:32, Rows:2, Useful REGs:41
F00,SF0,DCI: DCI Idx=0: Start CCE-Idx=0
F00,SF0,DCI: DCI Idx=1: Start CCE-Idx=2
F00,SF0,PDCCH: Idx=0: REG-Idx=10: Subcarrier=0, Symbol=1
F00,SF0,PDCCH: Idx=0: REG-Idx=10: Subcarrier=1, Symbol=1
F00,SF0,PDCCH: Idx=0: REG-Idx=10: Subcarrier=2, Symbol=1
F00,SF0,PDCCH: Idx=0: REG-Idx=10: Subcarrier=3, Symbol=1
F00,SF0,PDCCH: Idx=1: REG-Idx=26: Subcarrier=0, Symbol=2
F00,SF0,PDCCH: Idx=1: REG-Idx=26: Subcarrier=1, Symbol=2
F00,SF0,PDCCH: Idx=1: REG-Idx=26: Subcarrier=2, Symbol=2
F00,SF0,PDCCH: Idx=1: REG-Idx=26: Subcarrier=3, Symbol=2
.......
```

The Figure 3-1 shows the resource allocation for this example.

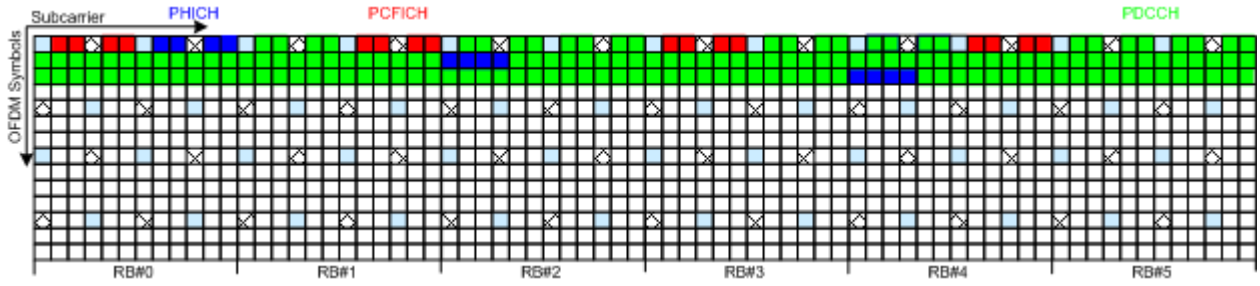



*Figure 3-1: Example of DL control information mapping*

### 3.2.2.2 Extended UCI Logfile

The extended UCI logfile summarizes the information for the whole generated signal and can contain information for more than one frame. The information is grouped in rows with the following syntax:

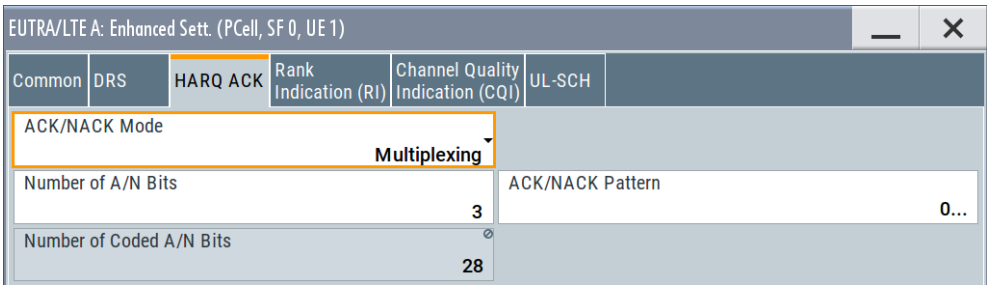
```
PUSCH_<Frame#>_<Subframe#>: <CW#>:

No.HARQ Bits=<HARQ#>,No.RI Bits=<RI#>,No.CQI Bits=<CQI#>,
No.coded A/
N Bits=<CodedHARQ#>,No.coded RI Bits=<CodedRI#>,
No.coded CQI Bits=<CodedCQI#>,No.coded UL-SCH Bits=<UL-SCH#>
```

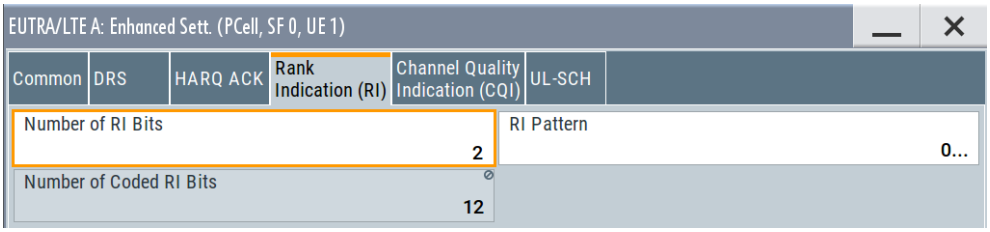
In case of NB-IoT, extended UCI logfile contains the number of bits per transmission, where transmissions are indicated with their starting frame and subframe indexes. The logfile has the following format:

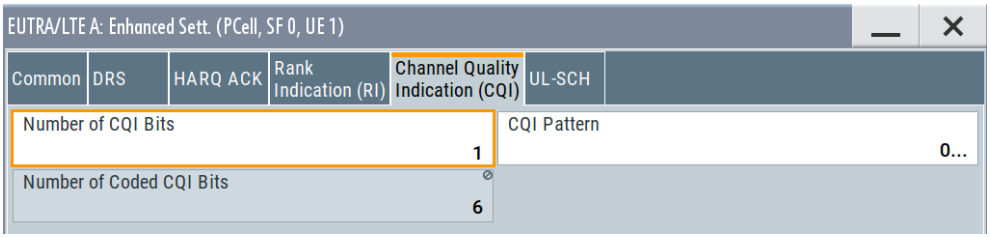
```
NPUSCH_<Frame#>_<Subframe#>: <CW#>:

No.HARQ Bits=<HARQ#>,No.coded UL-SCH Bits=<UL-SCH#>
```

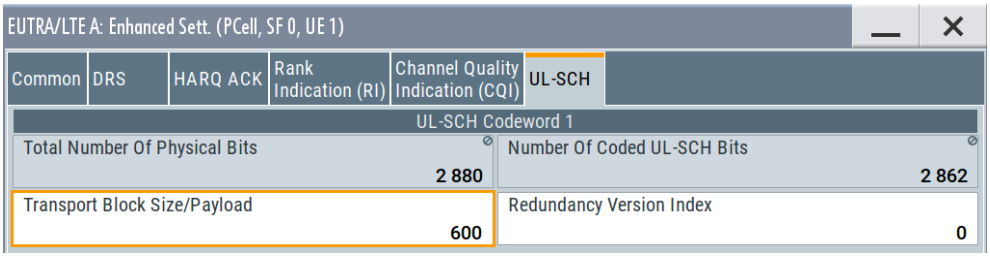
**Example:**

The PUSCH of a release 8/9 UE carries multiplexed control information and data (`UCI+UL-SCH`) and the channel is configured as follows:

- 

- 

- 

- 


The first line of the logfile is:

```
F00,SF0,
PUSCH UCI+UL-SCH Number of Bits: No.HARQ Bits=3,No.RI Bits=2,
No.CQI Bits=1,No.coded HARQ Bits=28,No.coded RI Bits=12,
No.coded CQI Bits=6,No.coded UL-SCH Bits=2862
```

### 3.2.3 Summary Logfiles Contents

The summary logfiles contain general information on the individual signal processing blocks, like the used rate matching parameters or allocation mapping.

In case of NB-IoT, the summary logfile contains also the base sequence index `u` and, if group hopping is used, the group hopping pattern `f_gh` and the sequence shift `f_ss`. The base sequence shift `u` is logged per slot; the `f_gh` and `f_ss` are logged on recalculation.

## 3.3 Signal Processing Chains and Logging Points

Logfile generation can be enabled after at the so called logging point. Logging points (PTxx) are available after each completed processing stage up to the "Precoding". Not logged are the results of the "Resource Element Mapping/OFDM Mapper".

For detailed information about the signal processing of all channels, refer to TS 36.212 and TS 36.211.

### Signal processing in downlink

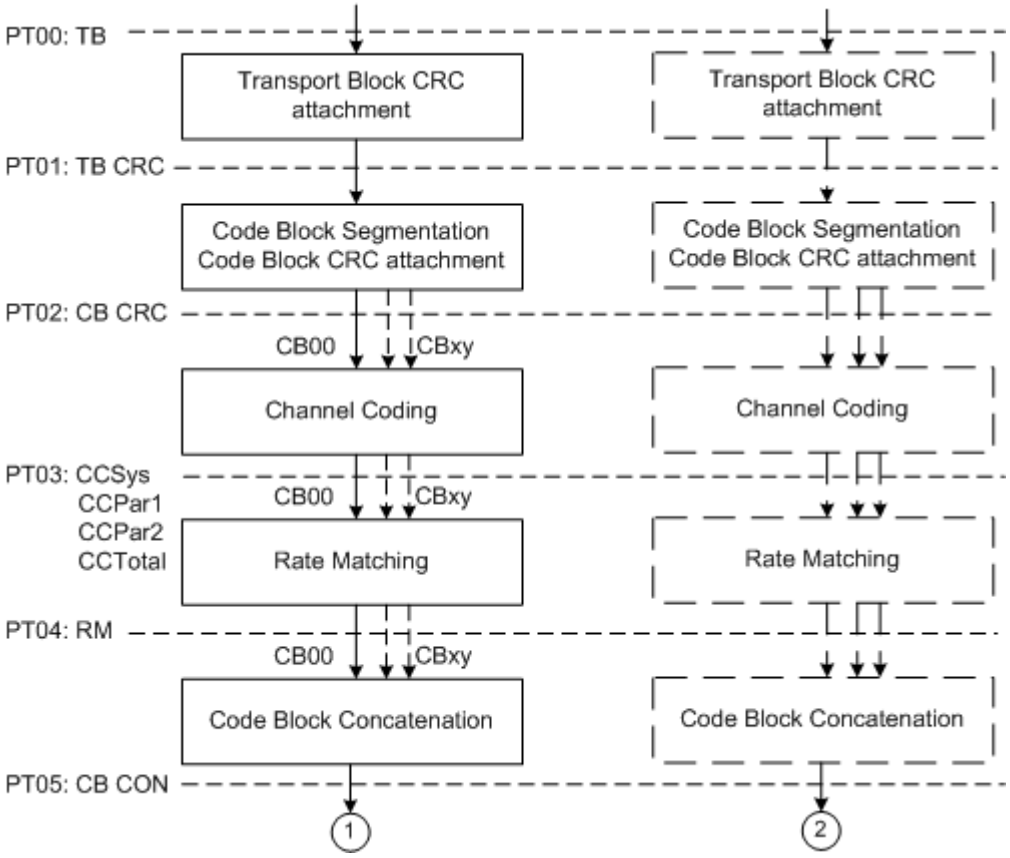



*Figure 3-2: Transport channel processing for DL-SCH*

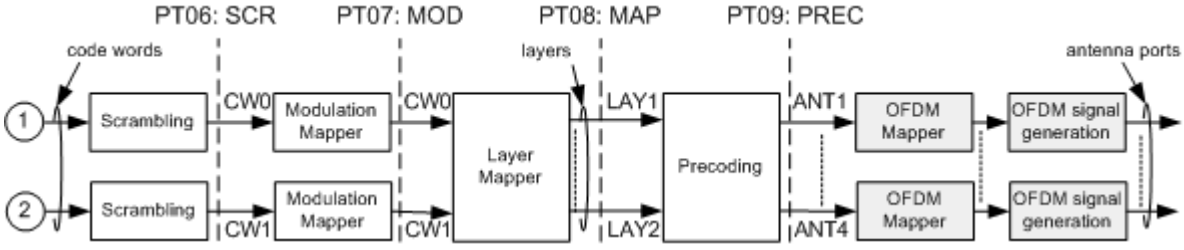



*Figure 3-3: Overview of physical channel processing*

The Table 3-5 gives an overview of the logging point available for the DL channels.

*Table 3-5: Available logging points per DL channel*

| Point/ Channel | PT00 TB | PT01 TBCRC | PT02 CBCRC | PT03 CC | PT04 RM | PT05 CBCON | PT06 SCR | PT07 MOD | PT08 MAP | PT09 PREC |
|---|---|---|---|---|---|---|---|---|---|---|
| PDSCH | X | X | X | X | X | X | X | X | X | X |
| NPDSCH<br>NPDSCH-SIB1 | X | X | | X | X | | X | X | X | X |
| PDSCH-SIB1 | X | X | X | X | X | X | X | X | X | X |
| PBCH [1)] | X | X | | X | X | | X | X | X | X |
| NPBCH | X | X | | X | X | | X | X | X | X |
| PCFICH [1)] | X | | | | | | X | X | X | X |
| PHICH [1)] | X | | | | | | | X [2)] | X [2)] | X [2)] |
| PDCCH [1)] | X [3)] | X [3)] | | X [3)] | X [3)] | X | X | X | X | X |
| NPDCCH | X | X | | X | X | | X | X | X | X |
| PMCH[1)] | X | X | | X | X | | X | X | X | X |

[1)] The channel has one codeword and one code block.

[2)] An individual file is generated per PHICH group.

[3)] An individual file is generated per DCI

**Signal processing in uplink**

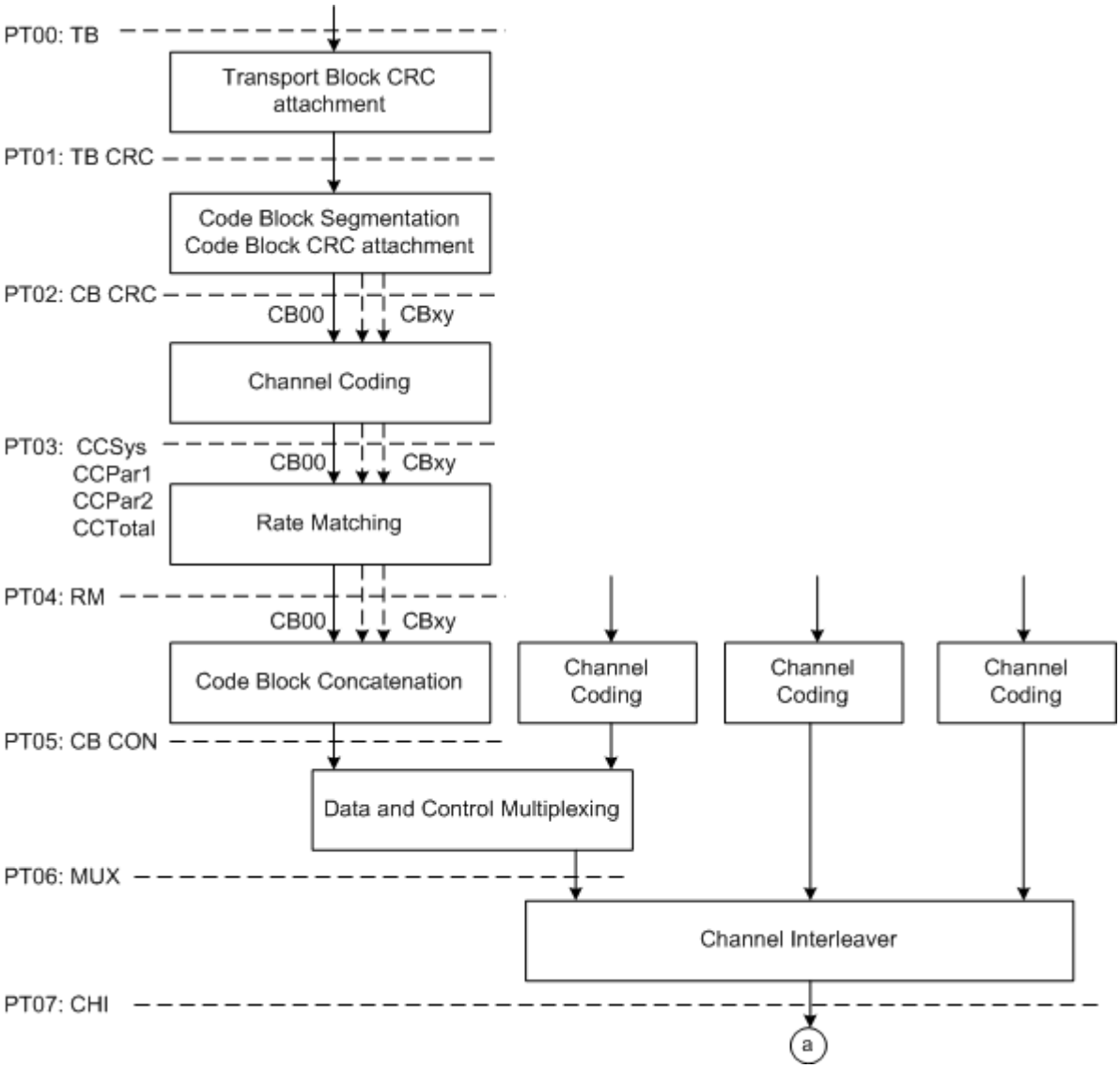



*Figure 3-4: Transport channel processing for UL-SCH [TS 36.212]*

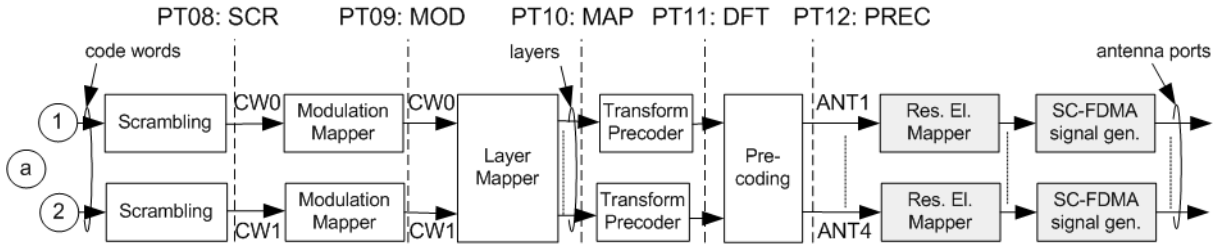



*Figure 3-5: Overview of uplink physical shared channel processing [TS 36.211]*

**Signal processing in NB-IoT uplink**

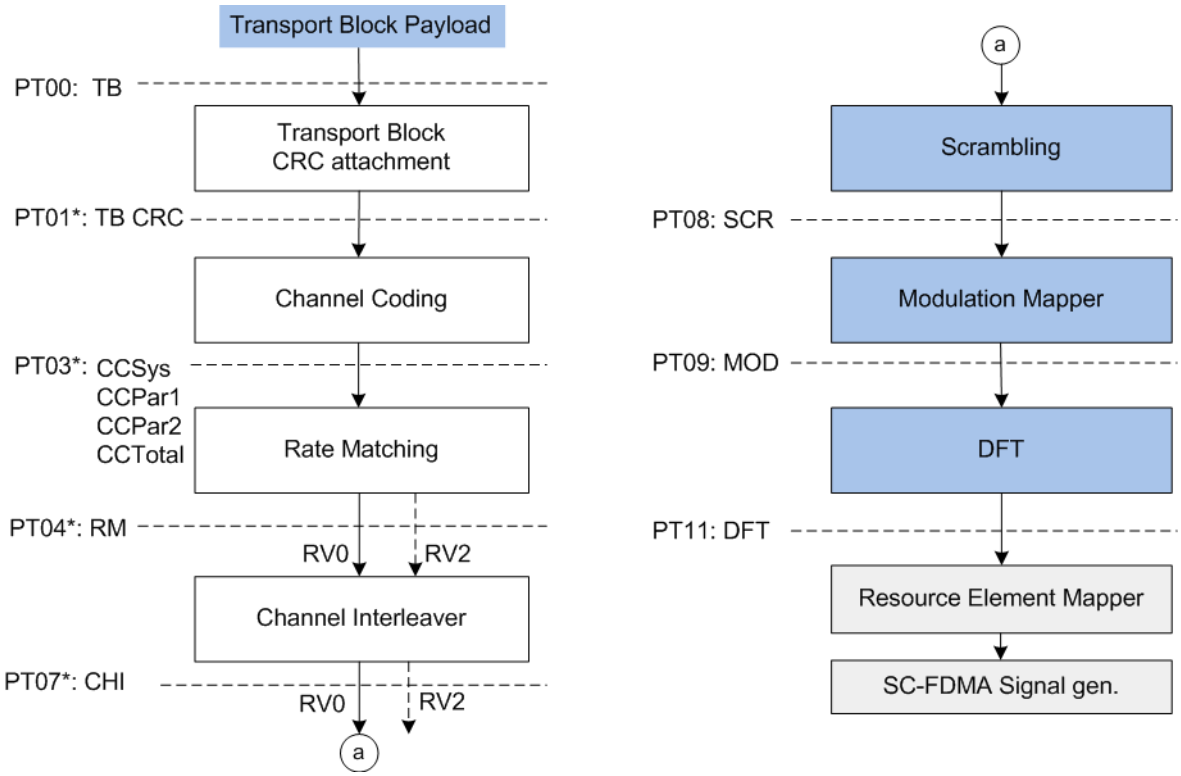



***Figure 3-6: Narrowband uplink physical shared channel NPUSCH processing [TS 36.212]***

* = Logging points that are available only for NPUSCH Format 1
RV = Redundancy version
Transport block payload = For PUSCH format 2, it the payload is 16 zero or 16 one bits

For an overview of logging points available for the UL channels, see:

- Table 3-6
- Table 3-3
- Table 3-4
- There are no specific logging points for the SRS; a logfile with SRS information is always created.

***Table 3-6: Available logging points per PUSCH channel***

| Point/ Channel | PT00 TB | PT01 TBCRC | PT02 CBCRC | PT03 CC | PT04 RM | PT05 CBCON | PT06 MUX | PT07 CHI | PT08 SCR | PT09 MOD | PT10 MAP | PT11 DFT | PT12 PREC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PUSCH | X | X | X | X | X | X | X | X | X | X | X | X | X |
| NPUSCH | X | X | - | X | X | - | - | X | X | X | - | X | - |

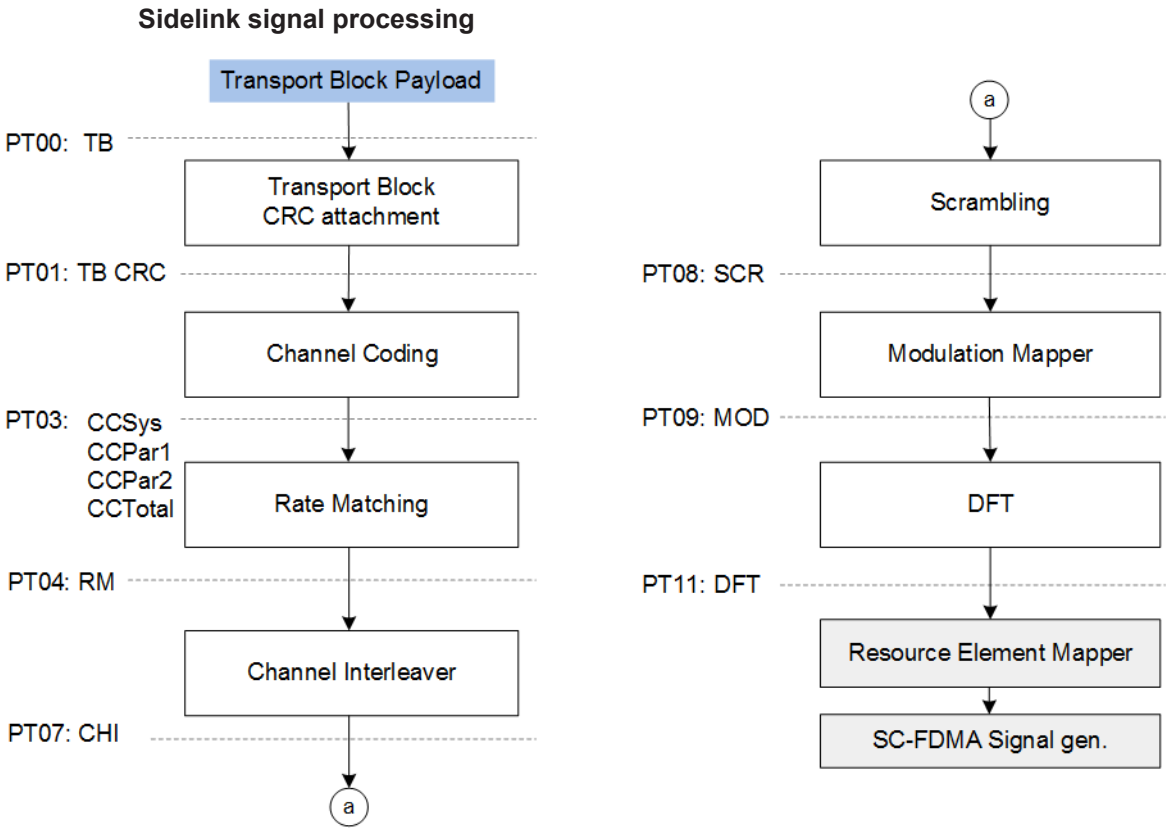



*Figure 3-7: Sidelink processing [TS 36.212]*

## 3.4 How to Use the Logfile Generation Functionality

The R&S SMW generates logfiles only if the logging state is enabled. Adjusting the settings in the "Logfile Generation" dialog does not affect the content of the generated LTE/IoT signal and does not cause a recalculation of the signal. The generation of new logfiles is triggered by changing of a signal relevant LTE/IoT parameter or by enabling/disabling the generation of LTE/IoT signal.

Activation of logfile generation slows down the calculation speed of the instrument.
Enable this function only if logfiles are explicitly requested.

**General workflow**

To enable the generation of logfiles, proceed as follows:

1. In the "EUTRA/LTE > Logfile Generation > Output Path" dialog, select the network directory the logfiles are saved to, e.g. `/var/user/logfiles`.

   **Note:** Select an empty directory. Existing logfiles are overwritten.
   Use different preambles to assure that previous logfiles are not lost.

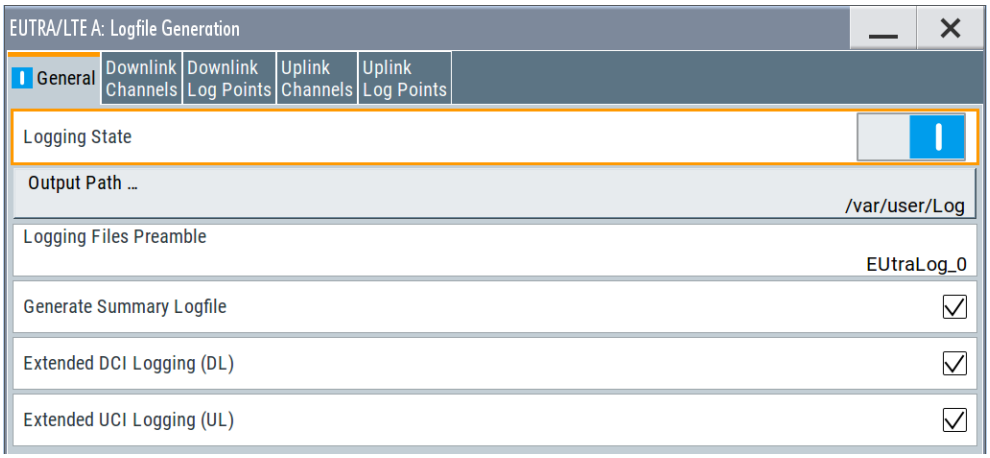
2. If necessary, enable "Extended DCI/UCI Logging".
3. Select the processing chain points for that logfiles are generated, e.g. "Point 3: Channel Coding".

   **Tip:** Not all the available logging points are relevant for all channels. The processing of the PBCH for instance does not include the step "Code block segmentation / CRC", i.e. even if the logging point "Point 2: Code block segmentation / CRC" is enabled, no logfile is generated.
   No logfiles are generated also in case that the corresponding processing step is disabled in the EUTRA/LTE dialog. For example, if for a particular channel coding or scrambling are disabled, no logfiles for points 3, 6 and 8 are available for this channel. Logfiles are created after coding and scrambling are enabled for this particular channel.
4. To enable logfile generation, select "Logging State > On".
5. Adjust the EUTRA/LTE settings as required.
6. Set "EUTRA/LTE > State > On".

## 3.5 Logfile Generation Settings

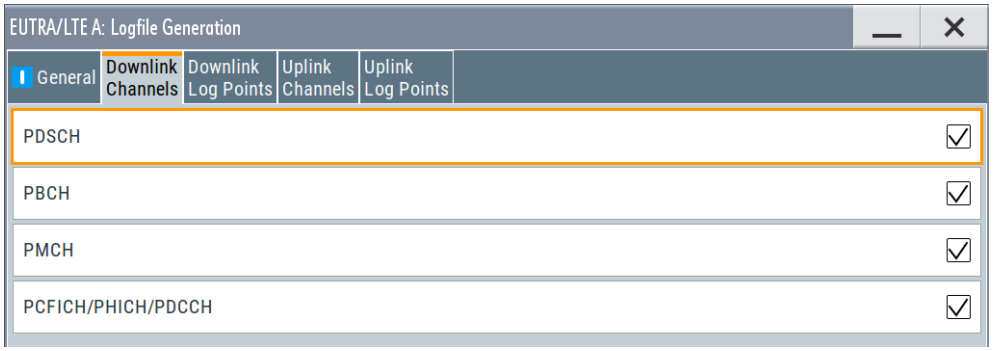
Access:

► Select "EUTRA/LTE > General > Logfile Generation".

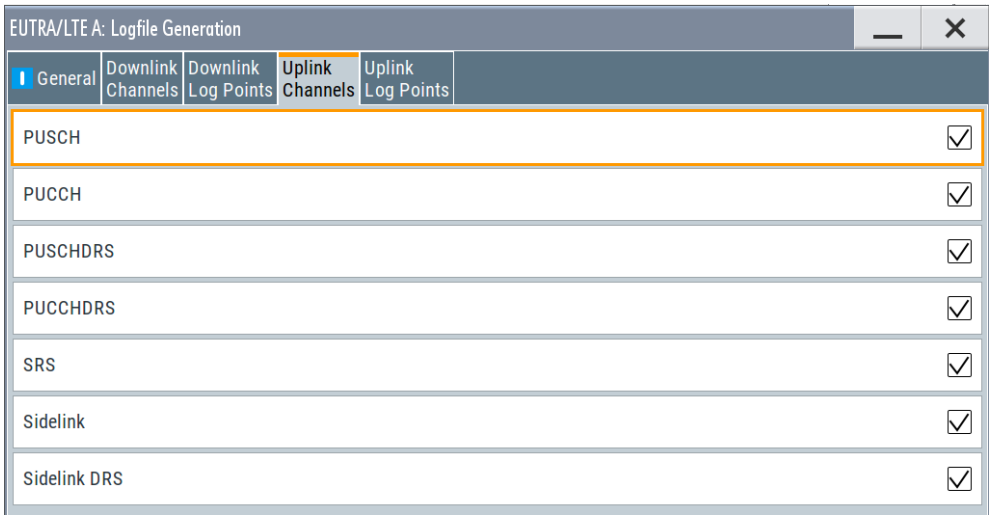



The dialog is divided into several tabs. The general tab comprises the settings necessary to enable the logfile generation and configure the output file. The further tabs group the settings to define the different channels to be logged and the logging points for which logfiles are generated.

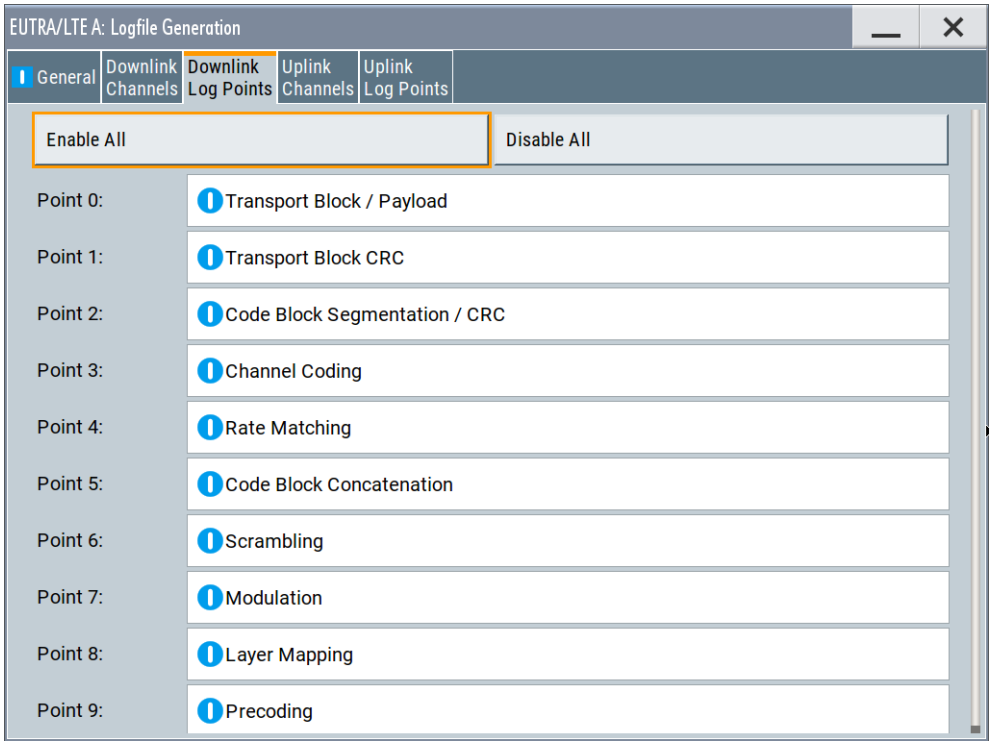
- Downlink Channels



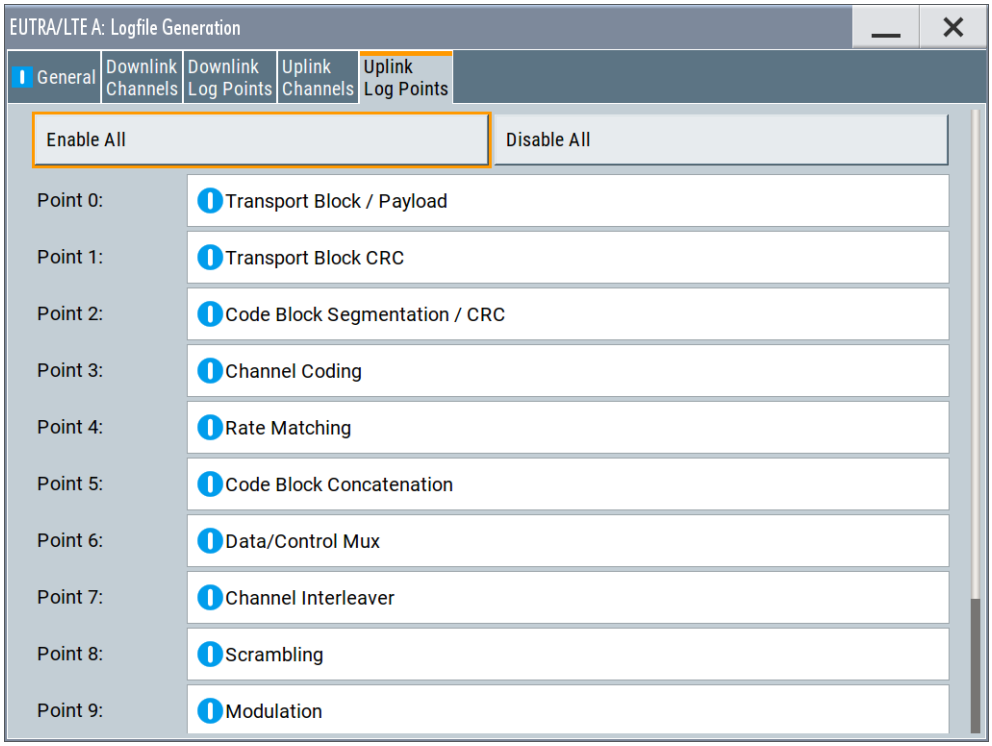

- Uplink Channels




- Downlink Log Pints



- Uplink Log Points




The remote commands required to define these settings are described in Chapter 3.6, "Remote-Control Commands", on page 27.



**Logging State**
Enables/disables logfile generation.

**Note:** Activation of logfile generation slows down the calculation speed of the instrument.
Enable this function only if logfiles are explicitly requested.

See also Chapter 3.4, "How to Use the Logfile Generation Functionality", on page 22.

Remote command:
`[:SOURce<hw>]:BB:EUTRa:LOGGen:STATe` on page 29

**Output Path**
Selects the network directory the logged files are stored in.

Remote command:
`[:SOURce<hw>]:BB:EUTRa:LOGGen:OUTPut` on page 29

**Logging Files Preamble**
Adds a preamble to the filename.

Refer to Chapter 3.2.1, "Filenames", on page 8 for a description of the file naming convention used.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:LFP on page 29

**Generate Summary Log**
Enables the generation of a summary logfile.

The summary logfiles contain general information on the individual signal processing blocks, like the used rate matching parameters or allocation mapping.

In case of NB-IoT, the summary logfile contains also the base sequence index `u` and, if group hopping is used, the group hopping pattern `f_gh` and the sequence shift `f_ss`. The base sequence shift `u` is logged per slot; the `f_gh` and `f_ss` are logged on recalculation.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:GSLogfile on page 29

**Extended DCI/UCI Logging**
Enables the generation of a logfile with extended information regarding the DCI/UCI mapping.

For description of the content of the generated file, see:
- Chapter 3.2.2.1, "Extended DCI Logfile", on page 13
- Chapter 3.2.2.2, "Extended UCI Logfile", on page 16.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:EDLogging on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:EULogging on page 30

**Physical Channels**
Selects the channel for which logfiles are generated.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:ENCC on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PBCH on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PDSCh on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PMCH on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUSCh on page 31
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUSDrs on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUCCh on page 31
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUCDrs on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SRS on page 31
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SL on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SLD on page 31

**Enable/Disable All**
Enables/disables all logging points.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:EALL on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:DALL on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:EALL on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:DALL on page 30

**Logging Point**
Enables/disables one particular logging point.

Refer to Chapter 3.3, "Signal Processing Chains and Logging Points", on page 17 for description on the available logging points.

Remote command:
[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:LOGPoint<ch0> on page 30
[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:LOGPoint<ch0> on page 30

## 3.6 Remote-Control Commands

The following commands are required to generate logfiles with the **Log file generation option R&S SMW-K81** in a remote environment. We assume that the R&S SMW has already been set up for remote operation in a network as described in the R&S SMW 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 SMW user manual.

### Common Suffixes

The following common suffixes are used in remote commands:

| Suffix | Value range | Description |
|---|---|---|
| ENTity<ch> | 1 to 4 | Entity in a multiple entity configuration<br>ENTity3\|4 require option R&S SMW-K76 |
| SOURce<hw> | [1] to 4 | Available baseband signals |
| OUTPut<ch> | 1 to 3 | Available markers |

**Using SCPI command aliases for advanced mode with multiple entities**
You can address multiple entities configurations by using the SCPI commands starting with the keyword SOURce or the alias commands starting with the keyword ENTity.

Note that the meaning of the keyword SOURce<hw> changes in the second case.

For details, see section "SCPI Command Aliases for Advanced Mode with Multiple Entities" in the R&S SMW user manual.

### Programming example

This description provides a simple programming example. The purpose of the example is to present **all** commands for a given task. In real applications, one would rather reduce the examples to an appropriate subset of commands.

The programming examples have been tested with a software tool which provides an environment for the development and execution of remote tests. To keep the example as simple as possible, only the "clean" SCPI syntax elements are reported. Non-executable command lines (e.g. comments) start with two // characters.

At the beginning of the most remote control program, an instrument (p)reset is recommended to set the instrument to a definite state. The commands `*RST` and `SYSTem:PRESet` are equivalent for this purpose. `*CLS` also resets the status registers and clears the output buffer.

### Example: Logfiles generation

```
SOURce1:BB:EUTRa:LOGGen:OUTPut '/var/user/logfiles'
SOURce1:BB:EUTRa:LOGGen:LFP "EUtraLog_0"
SOURce1:BB:EUTRa:LOGGen:DL:EDLoggong ON
SOURce1:BB:EUTRa:LOGGen:DL:PBCH ON
SOURce1:BB:EUTRa:LOGGen:DL:LOGP9 ON
SOURce1:BB:EUTRa:LOGGen:STATe ON
SOURce1:BB:EUTRa:STATe ON
```

The following commands specific to the **Log file generation option R&S SMW-K81** are described here:

**[:SOURce<hw>]:BB:EUTRa:LOGGen:STATe** <LoggingState>

Enables/disables logfile generation.

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

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Logging State" on page 25

**[:SOURce<hw>]:BB:EUTRa:LOGGen:OUTPut** <OutputPath>

Selects the network directory the logged files are stored in.

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Output Path" on page 25

**[:SOURce<hw>]:BB:EUTRa:LOGGen:LFP** <Preamble>

Sets the preamble added to the file name.

See Chapter 3.2.1, "Filenames", on page 8 for a description of the file naming conventions.

**Parameters:**
<Preamble> string
*RST: K55Log

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Logging Files Preamble" on page 26

**[:SOURce<hw>]:BB:EUTRa:LOGGen:GSLogfile** <GenSumLog>

Enables the generation of a summary logfile.

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

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Generate Summary Log" on page 26

**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:EDLogging** <ExtDciLog>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:EULogging** <ExtUciLog>

Enables the generation of a logfile with extended information regarding the DCI/UCI mapping.

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

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Extended DCI/UCI Logging" on page 26

**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:EALL**
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:EALL**
**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:DALL**
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:DALL**

Enables/disables all logging points.

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Enable/Disable All" on page 26

**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:LOGPoint<ch0>** <LogPointState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:LOGPoint<ch0>** <LogPointState>

Enables/disables one particular logging point.

Refer to Chapter 3.3, "Signal Processing Chains and Logging Points", on page 17 for description on the available logging points.

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

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Logging Point" on page 27

**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:ENCC** <EnccLogState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PBCH** <PbchLogState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PDSCh** <PdschLogState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:DL:PMCH** <State>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUSDrs** <PuschDrsLog>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUCDrs** <PuschDrsLog>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SL** <LogSidelink>

**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SLD** <LogSidelinkDrs>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:SRS** <SrsState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUCCh** <PucchLogState>
**[:SOURce<hw>]:BB:EUTRa:LOGGen:UL:PUSCh** <PuschLogState>

Enables the channel or reference signal for that logfiles are generated.

**Parameters:**

<PuschLogState> 0 | 1 | OFF | ON
*RST: ON

**Example:** See Example "Logfiles generation" on page 28.

**Options:** R&S SMW-K81

**Manual operation:** See "Physical Channels" on page 26

# 4 5G New Radio Logfile Generation

## 4.1 Required Options

The generation of logfiles requires:

- Standard or wideband baseband generator (R&S SMW-B10/-B9)
- Baseband main module (R&S SMW-B13) or wideband baseband main module (R&S SMW-B13XT)
- Option 5G New Radio (R&S SMW-K144)
  (per signal path)
- Option log files generation (R&S SMW-K81)
  Two options R&S SMW-K81 are required if system configuration with baseband blocks with more that one output is used.
  For example, as it is in configuration with coupled or coupled per entity baseband sources.

## 4.2 Output Files

Logfiles are generated after each step of signal processing chain defined in TS 38.212.

The R&S SMW saves the output logfiles in a user-defined network directory, selected with the parameter Output Path. A folder structure is automatically created according to the number of configured users, BWPs, channels, etc. The logfiles are named according to the naming convention described in "File format and filenames" on page 32.

### File format and filenames

The log files are files in json format and are named according to the following naming structure `after_<SignalPorcessingStep>.json`. See Table 4-1 for an overview of the signal processign steps and the filenames per channel.

*Table 4-1: Log files*

| File name (signal processing step) | PxSCH (per codeword) | CORESET (PDCCH/DCI) | PUCCH (Format 2, 3, 4) | PRACH | SSPBCH (MIB) |
|---|---|---|---|---|---|
| transport_block | x | - | - | - | - |
| after_block_segmentation | - | 0* | - | - | - |
| after_code_block_crc_attachment | - | 0* | - | - | - |
| after_interleaving | - | 0* | - | - | - |
| after_crc_attachment | 0* | - | - | - | - |
| after_code_block_segmentation | 0* | - | - | - | - |
| after_channel_coding | 0* | 0* | - | - | - |

| File name (signal processing step) | PxSCH (per codeword) | CORESET (PDCCH/DCI) | PUCCH (Format 2, 3, 4) | PRACH | SSPBCH (MIB) |
|---|---|---|---|---|---|
| after_rate_matching | 0* | 0* | - | - | - |
| after_code_block_concatenation | x | 0* | - | - | - |
| after_encoding | x | x | x | | x |
| after_scrambling | x | x | x | | x |

0* = if channel coding is enabled

## 4.3 How to Generate Logfiles

The R&S SMW generates logfiles depending on the current configuration, like active channels, users, BWPs, codewords, channel coding, etc.

Activating the logfile generation does not affect the content of the generated 5G NR signal and does not cause a recalculation of the signal. The generation of new logfiles is triggered by changing of a relevant 5G NR parameter or by enabling/disabling the generation of 6G NR signal.

Activation of logfile generation slows down the calculation speed of the instrument. Enable this function only if logfiles are explicitly requested.

**General workflow**

To enable the generation of logfiles, proceed as follows:

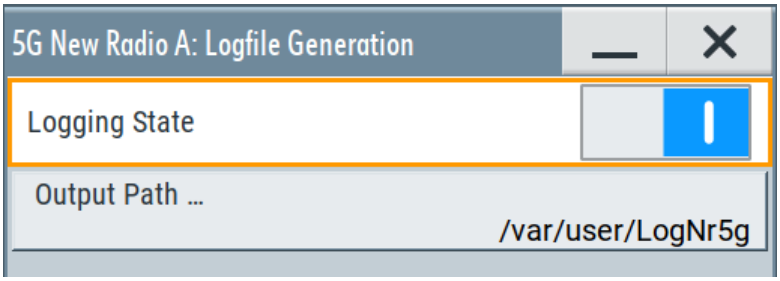
1. Select "5G New Radio > Logfile Generation > Output Path".
   Set the network directory the logfiles are saved to, e.g. `/var/user/logfiles`.

   **Note:** Select an empty directory. Existing logfiles are overwritten.
2. Select "Logging State > On".
3. Adjust the 5G NR settings as required.
4. Set "5G New Radio > State > On".

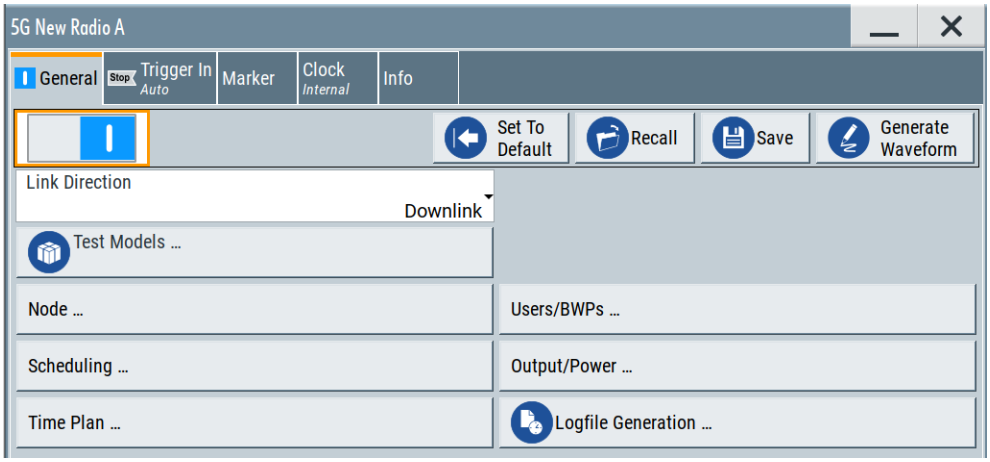
## 4.4 Logfile Generation Settings

Access:

1. Select "5G New Radio > General > Logfile Generation".
2. To enable log file generations, set "Logging State > On".




3. To start generation and logging, set "5G New Radio > State > On".






**Logging State**
Enables/disables logfile generation.

**Note:** Activation of logfile generation slows down the calculation speed of the instrument.
Enable this function only if logfiles are explicitly requested.

See also Chapter 4.3, "How to Generate Logfiles", on page 33.

Remote command:
`[:SOURce<hw>]:BB:NR5G:LOGGen:STATe` on page 35

**Output Path**
Selects the network directory the logged files are saved in.

Per default, files are saved in the user directory of the instrument.

Remote command:
`[:SOURce<hw>]:BB:NR5G:LOGGen:OUTPut` on page 35

## 4.5 Logging Commands

Option: R&S SMW-K81

**Example: Activating logfile generation**

To start logfile generation with the defalut settings:

```
SOURce1:BB:NR5G:PRESet
SOURce1:BB:NR5G:STATe 1
SOURce1:BB:NR5G:LOGGen:OUTPut "/var/user/"
SOURce1:BB:NR5G:LOGGen:STATe 1
```



---

**[:SOURce<hw>]:BB:NR5G:LOGGen:STATe** <LogGenState>

Activates the logfile generation.

**Parameters:**

<LogGenState> 0 | 1 | OFF | ON
*RST: 0

**Example:** See Example "Activating logfile generation" on page 35.

**Manual operation:** See "Logging State" on page 34

---

**[:SOURce<hw>]:BB:NR5G:LOGGen:OUTPut** <LogGenOutPath>

Sets the directory the files are saved in.

**Parameters:**

<LogGenOutPath> string

**Example:** See Example "Activating logfile generation" on page 35.

**Manual operation:** See "Output Path" on page 34

# List of Commands

# Index