

# B-COMM Interface Manual

Reference Manual



# Table of Content

|          |                                                           |           |
|----------|-----------------------------------------------------------|-----------|
| <b>1</b> | <b>About this document</b>                                | <b>6</b>  |
| <b>2</b> | <b>WriteOrder for B-COMM</b>                              | <b>8</b>  |
| 2.1      | Introduction                                              | 8         |
| 2.2      | Options                                                   | 8         |
| 2.3      | Arguments                                                 | 8         |
| 2.4      | Orders for WriteOrder                                     | 9         |
| 2.5      | Further parameters for orders                             | 11        |
| 2.6      | WriteOrder: Examples                                      | 12        |
| 2.6.1    | Calling WriteOrder with argument -h                       | 12        |
| 2.6.2    | Calling WriteOrder with argument -h DateTimeWrite         | 14        |
| 2.6.3    | Writing an order DateTimeWrite                            | 15        |
| <b>3</b> | <b>Communication B-COMM IO/Application</b>                | <b>16</b> |
| 3.1      | Introduction                                              | 16        |
| 3.2      | B-COMM ApplicationDemo                                    | 16        |
| 3.3      | B-COMM Class documentation                                | 16        |
| 3.4      | Data classification                                       | 17        |
| 3.4.1    | Commands                                                  | 17        |
| 3.4.2    | Acknowledgements                                          | 17        |
| 3.4.3    | Data                                                      | 17        |
| 3.5      | Communication examples                                    | 18        |
| 3.5.1    | Error-free online communication                           | 18        |
| 3.5.2    | Error-free offline communication                          | 19        |
| 3.5.3    | Faulty communication                                      | 20        |
| 3.5.4    | Conflict-raising communication                            | 21        |
| 3.6      | Events from gateways or wireless components               | 22        |
| 3.6.1    | All events                                                | 22        |
| 3.6.2    | Event 15 "EvolNewLogIndication"                           | 30        |
| 3.6.3    | Event 18 "EvolDoorEvent"                                  | 38        |
| 3.6.4    | Event 42 "SensorStateChanged"                             | 39        |
| <b>4</b> | <b>B-COMM Helper</b>                                      | <b>40</b> |
| 4.1      | Introduction                                              | 40        |
| 4.2      | Calling the Helper                                        | 40        |
| 4.3      | Helper commands with XML                                  | 40        |
| 4.4      | Helper commands sorting                                   | 41        |
| 4.5      | Helper commands for starting and terminating of functions | 42        |
| 4.5.1    | ShutdownServer                                            | 42        |
| 4.5.2    | ShutdownHelper                                            | 42        |
| 4.5.3    | StartStream                                               | 42        |
| 4.5.4    | StopStream                                                | 43        |
| 4.5.5    | SwitchApplicationInterface                                | 43        |
| 4.6      | File related Helper commands                              | 44        |
| 4.6.1    | DeleteBookingFile                                         | 44        |
| 4.6.2    | DeleteOrderFile                                           | 44        |
| 4.6.3    | FileUpload                                                | 45        |
| 4.6.4    | FileUploadContent                                         | 45        |
| 4.6.5    | FileDownload                                              | 52        |
| 4.6.6    | SendDataRecord                                            | 52        |
| 4.6.7    | RenameCurrentBookingFile                                  | 53        |
| 4.6.8    | ReorganizeOrderFile                                       | 53        |
| 4.7      | Configuration related Helper commands                     | 54        |
| 4.7.1    | ReadConfiguration                                         | 54        |
| 4.7.2    | ReadGlobalConfigValue                                     | 55        |
| 4.7.3    | ReadNetworkAdapter                                        | 55        |
| 4.7.4    | ReadNetworkAdapters                                       | 56        |
| 4.8      | Client related Helper commands                            | 56        |
| 4.8.1    | ReadClient                                                | 56        |

|         |                                               |     |
|---------|-----------------------------------------------|-----|
| 4.8.2   | ReadClients                                   | 57  |
| 4.8.3   | RefreshClients                                | 57  |
| 4.9     | Stream related Helper commands                | 58  |
| 4.9.1   | ReadStream                                    | 58  |
| 4.9.2   | ReadStreams                                   | 58  |
| 4.9.3   | ReadStreamStatus                              | 59  |
| 4.9.4   | ReadStreamConfigValue                         | 59  |
| 4.10    | Terminal related Helper commands              | 60  |
| 4.10.1  | ReadTerminal                                  | 60  |
| 4.10.2  | ReadTerminals                                 | 60  |
| 4.10.3  | ReadTerminalSections                          | 61  |
| 4.11    | Actuator related Helper commands              | 62  |
| 4.11.1  | ReadActuatorConfigStatus                      | 62  |
| 4.11.2  | ReadActuatorEventLog                          | 64  |
| 4.11.3  | ReadActuatorWhitelistEventLog                 | 65  |
| 4.11.4  | ReadActuatorCardLinkEventLog                  | 66  |
| 4.11.5  | ReadActuatorWhitelistConfigStatus             | 67  |
| 4.11.6  | ReadActuatorCardLinkConfigStatus              | 69  |
| 4.12    | Order related Helper commands                 | 71  |
| 4.12.1  | ReadOrders                                    | 71  |
| 4.12.2  | ReadOrdersExecuted                            | 71  |
| 4.12.3  | ReadOrderStatus                               | 72  |
| 4.12.4  | ReadRecentOrderStatus                         | 72  |
| 4.12.5  | WriteOrder                                    | 73  |
| 4.12.6  | WriteOrderOverwrite                           | 74  |
| 4.12.7  | WriteOrderDirect                              | 75  |
| 4.12.8  | WriteOrderOverwriteDirect                     | 76  |
| 4.13    | Booking related Helper commands               | 77  |
| 4.13.1  | ReadBookings                                  | 77  |
| 4.14    | CardLink related Helper commands              | 78  |
| 4.14.1  | ReadCardLinkConfiguration                     | 78  |
| 4.14.2  | WriteCardLinkActuatorConfiguration            | 78  |
| 4.14.3  | UpdateCardLinkActuatorConfiguration           | 79  |
| 4.14.4  | WriteCardLinkAdministrationAreaConfiguration  | 81  |
| 4.14.5  | UpdateCardLinkAdministrationAreaConfiguration | 82  |
| 4.14.6  | WriteCardLinkConfiguration                    | 84  |
| 4.15    | Whitelist related Helper commands             | 85  |
| 4.15.1  | WriteWhitelistPool                            | 85  |
| 4.15.2  | WriteWhitelistConfiguration                   | 86  |
| 4.15.3  | WriteWhitelistActuatorConfiguration           | 88  |
| 4.15.4  | UpdateWhitelistActuatorConfiguration          | 89  |
| 4.15.5  | WriteWhitelistKeyPlanConfiguration            | 91  |
| 4.15.6  | UpdateWhitelistKeyPlanConfiguration           | 92  |
| 4.15.7  | WriteWhitelistActuatorWhitelist               | 94  |
| 4.15.8  | AppendWhitelistActuatorWhitelist              | 95  |
| 4.15.9  | DeleteWhitelistActuatorWhitelist              | 96  |
| 4.15.10 | WriteWhitelistGatewayConfiguration            | 97  |
| 4.15.11 | UpdateWhitelistGatewayConfiguration           | 98  |
| 4.15.12 | ReadKnownGateways                             | 99  |
| 4.15.13 | ReadGateways                                  | 99  |
| 4.15.14 | ReadActuatorJoinStates                        | 99  |
| 4.15.15 | DoorControl                                   | 100 |
| 4.16    | CardLink wireless related Helper commands     | 102 |
| 4.16.1  | WriteCardLinkConfiguration                    | 102 |
| 4.16.2  | WriteCardLinkGatewayConfiguration             | 104 |
| 4.16.3  | UpdateCardLinkGatewayConfiguration            | 105 |
| 4.16.4  | ReadKnownGateways                             | 106 |
| 4.16.5  | ReadGateways                                  | 107 |
| 4.16.6  | ReadActuatorJoinStates                        | 108 |
| 4.16.7  | DoorControl                                   | 109 |
| 4.17    | Biometric related Helper commands             | 109 |
| 4.17.1  | RefreshTemplateList                           | 109 |
| 4.17.2  | RefreshTemplateList Client                    | 109 |
| 4.18    | Programmer related Helper commands            | 110 |
| 4.18.1  | ReadProgrammerStatus                          | 110 |
| 4.19    | elologic related Helper commands              | 111 |
| 4.19.1  | WriteElologicActuatorConfiguration            | 111 |

|          |                                                                   |            |
|----------|-------------------------------------------------------------------|------------|
| 4.19.2   | WriteElolegicKeyPlanConfiguration                                 | 111        |
| 4.20     | Mobile Access related Helper commands                             | 112        |
| 4.20.1   | WriteWhitelistMobileAccessCredentials                             | 112        |
| 4.20.2   | DeleteWhitelistMobileAccessCredentials                            | 113        |
| 4.20.3   | CreateMobileAccessProjectDevices                                  | 114        |
| 4.20.4   | DeleteMobileAccessProjectDevices                                  | 114        |
| 4.20.5   | ListMobileAccessProjectsDevices                                   | 115        |
| 4.20.6   | ReadLegicConnectEvents                                            | 116        |
| 4.20.7   | ListMobileAccessVcps                                              | 117        |
| 4.20.8   | ListMobileAccessProvisionedFiles                                  | 118        |
| 4.20.9   | DeleteMobileAccessVcps                                            | 120        |
| 4.20.10  | WriteMobileAccessVcps                                             | 121        |
| 4.21     | SSH key related Helper commands                                   | 121        |
| 4.21.1   | SSHKeyExport                                                      | 121        |
| <b>5</b> | <b>HTTP Helper</b>                                                | <b>122</b> |
| 5.1      | Introduction                                                      | 122        |
| 5.2      | Implementation of the HTTP helper into the system                 | 123        |
| 5.3      | Overview of the HTTP helper communication                         | 124        |
| 5.3.1    | Components of the HTTP helper                                     | 125        |
| 5.3.2    | Response codes                                                    | 128        |
| 5.4      | Performing the configuration                                      | 129        |
| 5.4.1    | Creating an application interface                                 | 129        |
| 5.4.2    | Provision of the certificates                                     | 132        |
| 5.4.3    | Configuring the HTTP helper                                       | 133        |
| 5.4.4    | Checking the "httphelper.ini" file                                | 135        |
| 5.5      | Supported HTTP helper commands                                    | 136        |
| 5.5.1    | Transmission direction B-COMM to HTTP helper and then to EACM/SWP | 136        |
| 5.5.2    | Transmission direction EACM/SWP to HTTP helper to B-COMM          | 137        |
| <b>6</b> | <b>AMQP Interface</b>                                             | <b>144</b> |
| 6.1      | Introduction                                                      | 144        |
| 6.1.1    | Broker                                                            | 145        |
| 6.1.2    | AMQP messages                                                     | 151        |
| 6.2      | AMQP message structure                                            | 152        |
| 6.2.1    | "header"-structure                                                | 152        |
| 6.2.2    | "data"-structure                                                  | 155        |
| 6.2.3    | "error"-structure                                                 | 160        |
| 6.2.4    | "response"-structure                                              | 162        |
| 6.3      | Commands from application to B-COMM server                        | 162        |
| 6.3.1    | Command structure                                                 | 162        |
| 6.3.2    | Response structure                                                | 163        |
| 6.3.3    | Command "getConfiguration"                                        | 164        |
| 6.3.4    | Command "getClients"                                              | 166        |
| 6.3.5    | Command "getClient"                                               | 168        |
| 6.3.6    | Command "getStreams"                                              | 170        |
| 6.3.7    | Command "getStream"                                               | 172        |
| 6.3.8    | Command "startStream"                                             | 174        |
| 6.3.9    | Command "stopStream"                                              | 175        |
| 6.3.10   | Command "getStreamStatus"                                         | 176        |
| 6.3.11   | Command "getTerminals"                                            | 178        |
| 6.3.12   | Command "getTerminal"                                             | 180        |
| 6.3.13   | Command "getKnownGateways"                                        | 183        |
| 6.3.14   | Command "getGateways"                                             | 185        |
| 6.3.15   | Command "getGateway"                                              | 187        |
| 6.3.16   | Command "setGateways"                                             | 189        |
| 6.3.17   | Command "getAdministrationAreas"                                  | 190        |
| 6.3.18   | Command "getAdministrationArea"                                   | 192        |
| 6.3.19   | Command "setAdministrationAreas"                                  | 194        |
| 6.3.20   | Command "getKeyplans"                                             | 195        |
| 6.3.21   | Command "getKeyplan"                                              | 197        |
| 6.3.22   | Command "setKeyplans"                                             | 199        |
| 6.3.23   | Command "getComponents"                                           | 200        |
| 6.3.24   | Command "getComponent"                                            | 204        |



|        |                                              |            |
|--------|----------------------------------------------|------------|
| 6.3.25 | Command "setComponents"                      | 208        |
| 6.3.26 | Command "getComponentEventLogFile"           | 209        |
| 6.3.27 | Command "getComponentEventLog"               | 213        |
| 6.3.28 | Command "sendRecord"                         | 215        |
| 6.3.29 | Command "reorganizeOrders"                   | 218        |
| 6.3.30 | Command "getOrders"                          | 220        |
| 6.3.31 | Command "getOrder"                           | 222        |
| 6.3.32 | Command "createOrders"                       | 224        |
| 6.3.33 | Command "deleteOrders"                       | 226        |
| 6.3.34 | Command "sendParameters"                     | 227        |
| 6.3.35 | Command "getWhitelist"                       | 231        |
| 6.3.36 | Command "setWhitelist"                       | 234        |
| 6.3.37 | Command "addWhitelist"                       | 236        |
| 6.3.38 | Command "sendWhitelist"                      | 238        |
| 6.3.39 | Command "deleteWhitelist"                    | 239        |
| 6.3.40 | Command "setDoorReleased"                    | 241        |
| 6.3.41 | Command "setDoorLocked"                      | 242        |
| 6.3.42 | Command "setDoorNormalOperation"             | 243        |
| 6.3.43 | Command "getWhitelistPool"                   | 244        |
| 6.3.44 | Command "setWhitelistPool"                   | 246        |
| 6.3.45 | Command "addWhitelistPool"                   | 247        |
| 6.3.46 | Command "shutdownServer"                     | 248        |
| 6.3.47 | Command "restartServer"                      | 249        |
| 6.3.48 | Command "getBookings"                        | 250        |
| 6.3.49 | Command "setMobileAccessCredentials"         | 252        |
| 6.3.50 | Command "writeMobileAccessCredentials"       | 254        |
| 6.3.51 | Command "deleteMobileAccessCredentials"      | 256        |
| 6.3.52 | Command "createProjectDevices"               | 258        |
| 6.3.53 | Command "listProjectDevices"                 | 260        |
| 6.3.54 | Command "deleteProjectDevices"               | 262        |
| 6.3.55 | Command "listProvisionedFiles"               | 264        |
| 6.3.56 | Command "listVcps"                           | 267        |
| 6.3.57 | Command "deleteVcps"                         | 269        |
| 6.3.58 | Command "writeVcps"                          | 271        |
| 6.4    | Examples                                     | 273        |
| 6.4.1  | Data record/event from device to application | 273        |
| 6.5    | Description of B-COMM structures             | 276        |
| 6.5.1  | Client structure                             | 276        |
| 6.5.2  | Stream structure                             | 278        |
| 6.5.3  | NetworkAdapter structure                     | 279        |
| 6.5.4  | Terminal structure                           | 280        |
| 6.5.5  | ApplicationInterface structure               | 281        |
| 6.5.6  | Gateway structure                            | 282        |
| 6.5.7  | Component structure                          | 283        |
| 6.5.8  | AdministrationArea structure                 | 289        |
| 6.5.9  | Keyplan structure                            | 291        |
| 6.5.10 | Stamp structure                              | 293        |
| 6.5.11 | Master structure                             | 294        |
| 6.5.12 | DoorGroup structure                          | 294        |
| 6.5.13 | MaintenanceMediaEntry structure              | 294        |
| 6.5.14 | TimeProfile structure                        | 295        |
| 6.5.15 | Order structure                              | 296        |
| 6.5.16 | WhitelistEntry structure                     | 297        |
| 6.5.17 | EventLogEntry structure                      | 298        |
| 6.5.18 | BlacklistEntry structure                     | 300        |
| 6.5.19 | StaticCredential structure                   | 300        |
| 6.5.20 | ProjectDevice structure                      | 300        |
|        | <b>Index</b>                                 | <b>301</b> |

# 1 About this document

This document describes the use of several program interfaces implemented in the B-COMM software.

**Validity** The content of this document refers to software versions as of version May 2023 (B-COMM 5.4).

**Addressee** This document is intended for IT administrators, system administrators and persons who write own applications for use together with B-COMM.

The descriptions in this document cannot substitute a product training.

**Contents and purpose** The main purpose of this document is to support software partners (SWPs) in programming own parts of software to work together with B-COMM.

The contents of this document include the following sections:

- Information about the utility program "WriteOrder"
- Information about the utility program "B-COMM Helper" including the HTTP Helper
- Information about the AQMP interface

**Passwords and IT-Security**

## NOTICE



### Unauthorized access to data and systems!

The passwords, user names and access data given in this document are factory default settings. This means, all these data is public or freely available.

Dormakaba strongly recommends to adapt these data immediately according to the safety guidelines of the purchaser.

All further data given in the screen shots or examples like names, IP addresses and similar data are dummy data.

**Additional documentation**

- B-COMM Reference Manual, No. 04037205

**Change log** The following section describes the most important changes compared to the previous edition of the document.

- New document layout
- New chapter ( [▶ 3.6](#) ) explaining the gateway and components parameter in XML files transmitted via application interface.
- Additional information in the chapter "Configuring the HTTP Helper"
- New graphic in the chapter "Configuring the HTTP Helper"
- Additional information in the chapter "Checking the httphelper.ini file"

**Orientation in the document**

This document contains the following orientation aids to help you locate specific topics more easily:

- Table of Contents: The table of contents at the beginning of the document gives an overview of all topics.
- Header. Each header contains the respective main chapter.
- Cross references. The cross references indicate the number of the chapter where additional information can be found. Example [ ▶ 5.7].
- Index. An index in alphabetical order is given at the end of the document.

**Meaning of the symbols**

Important information for proper handling of the software.

Non-observance of this information can cause device malfunction, system failure or data loss.



Tips and useful information. This information will help you to best use the product and its functions.



Instruction

A certain operation is described in detail by means of a step-by-step instruction.

## 2 WriteOrder for B-COMM

### 2.1 Introduction

The WriteOrder program is implemented in B-COMM and offers the additional option of creating orders in B-COMM. It is called as described for example in chapter Calling WriteOrder with argument -h [▶ 2.6.1](#).

The orders can be created via:

- direct input in the command line
- batch files
- scripts.

The orders are created via RMI on the corresponding B-COMM Server. The corresponding parameters of the order are shown as arguments when calling the order.

The command syntax for WriteOrder is always

*WriteOrder [Options] Arguments Order*

### 2.2 Options

The following optional parameters can be entered:

| Parameter | Meaning                     |
|-----------|-----------------------------|
| -i        | IP address of B-COMM server |
| -p        | RMI port of B-COMM server.  |

If these parameters are not entered, WriteOrder uses the first IP address on the current computer and the port 1099 for the RMI communication with B-COMM server.

### 2.3 Arguments

The definition of an argument is mandatory. The following arguments can be defined:

| Argument  | Meaning                                                                                                                                                                                        |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -h        | Causes the output of general definitions for WriteOrder as a help, see also the example in 2.6.1 . This includes, for example, the command syntax and the meaning of the parameters.           |
| -h order  | Causes the output of special definitions for the order defined in the syntax, see also the example in 2.6.1 . This includes, for example, value ranges of parameters and possible subcommands. |
| -v        | Displays the WriteOrder version.                                                                                                                                                               |
| -c client | Client for whom an order is to be written.                                                                                                                                                     |
| -s stream | Stream for which an order is to be written.                                                                                                                                                    |

## 2.4 Orders for WriteOrder

The orders which can be executed by WriteOrder are summarized in the following list. A distinction is made between orders for a terminal stream (BCLANxy or BCSErxy) and orders for Card-Link (BCSCWxy). The parameters assigned to the individual order are described in detail in the B-COMM manual in the order file related chapters. The argument **-h <Name of the order>** allows you to call this information from the system.

In the case of terminal streams, the terminal must be specified either via the AlternativeID (-ov) or via GID (-og) and DID (-od). Orders can be sent to an individual terminal by specifying the GID/DID/AlternativeID or to all terminals of a stream by specifying the wildcard "\*" for GID/DID.

The entry of the priority for the order (-oy) is optional. If no entry is made here, priority = 2 (normal) applies.

The entry of the parameters for the scheduler is optional. If the parameter SchedulerActivated (-osa) is not defined, the scheduler is not activated. If the parameter SchedulerActivated (-osa) is transferred, the additional parameters SchedulerStartTime (-oss) and SchedulerIntervall (-osi) are mandatory. More information about the scheduler can be found in the B-COMM manual.

| Parameter | Designation of the order      |                                                 |
|-----------|-------------------------------|-------------------------------------------------|
| -oc       | DataBufferControl             | Orders for terminal stream (BCLANxy or BCSErxy) |
| -oc       | DataRead                      |                                                 |
| -oc       | DateTimeWrite                 |                                                 |
| -oc       | DaylightSavingTime            |                                                 |
| -oc       | DisplayMessage                |                                                 |
| -oc       | EncryptionKeyChange           |                                                 |
| -oc       | LEGICUpdateRecordDownload     |                                                 |
| -oc       | LEGICUpdateRecordUpload       |                                                 |
| -oc       | LEGICValidationRecordDownload |                                                 |
| -oc       | LEGICValidationRecordUpload   |                                                 |
| -oc       | MasterRecordDownload          |                                                 |
| -oc       | MasterRecordUpload            |                                                 |
| -oc       | OperatingModeWrite            |                                                 |
| -oc       | ParameterDownload             |                                                 |
| -oc       | ParameterUpload               |                                                 |
| -oc       | ReadStreamStatus              |                                                 |
| -oc       | RenameCurrentBookingFile      |                                                 |
| -oc       | ReorganizeOrderFile           |                                                 |
| -oc       | ShutdownStream                |                                                 |
| -oc       | StartupStream *)              |                                                 |
| -oc       | TransparentDataDownload       |                                                 |
| -oc       | SSHKeyExport                  |                                                 |
| -oc       | SSHKeyReplacement             |                                                 |

\*) No order. However, it is possible to start the corresponding stream via this command.



| Parameter | Designation of the order  |                                      |
|-----------|---------------------------|--------------------------------------|
| -oc       | LogUploadActuator         | Orders for CardLink stream (BCSCWxy) |
| -oc       | ParameterDownloadActuator |                                      |
| -oc       | StatusUploadActuator      |                                      |
| -oc       | ShutdownStream            |                                      |
| -oc       | StartupStream *)          |                                      |

\*) No order. However, it is possible to start the corresponding stream via this command.

| Parameter | Designation of the order                 |                                           |
|-----------|------------------------------------------|-------------------------------------------|
| -oc       | BlacklistDeleteWirelessActuatorGateway   | Orders for CardLink Wireless or Whitelist |
| -oc       | BlacklistDownloadWirelessActuatorGateway |                                           |
| -oc       | DateTimeGateway                          |                                           |
| -oc       | DateTimeGatewayGet                       |                                           |
| -oc       | DoorControlWirelessActuatorGateway       |                                           |
| -oc       | IdentifyWirelessActuatorGateway          |                                           |
| -oc       | ParameterDownloadGateway                 |                                           |
| -oc       | ParameterDownloadWirelessActuatorGateway |                                           |
| -oc       | ReadDoorWirelessActuatorGateway          |                                           |
| -oc       | ReadLinkQualityWirelessActuatorGateway   |                                           |
| -oc       | StatusUploadWirelessActuatorGateway      |                                           |
| -oc       | TracebackUploadWirelessActuatorGateway   |                                           |
| -oc       | WhitelistDeleteWirelessActuatorGateway   |                                           |
| -oc       | WhitelistDownloadWirelessActuatorGateway |                                           |

| Parameter | Designation of the order   |                  |
|-----------|----------------------------|------------------|
| -oc       | TemplateReplication        | Biometric orders |
| -oc       | TemplateReplicationMRBased |                  |
| -oc       | TemplateIDUpload           |                  |
| -oc       | TemplateDelete             |                  |

## 2.5 Further parameters for orders

Depending on the order and stream type, additional parameters must be entered or are possible which have an effect on the following keys in the order file:

| Parameter | Meaning                       |
|-----------|-------------------------------|
| -os       | Subcommand                    |
| -ox       | Extension                     |
| -of       | File                          |
| -ofe      | FTP enable (True or False)    |
| -ot       | Data                          |
| -oy       | Priority                      |
| -og       | GID                           |
| -od       | DID                           |
| -ov       | Alternative ID                |
| -osa      | Activate scheduler            |
| -oss      | Scheduler start time          |
| -osi      | Scheduler interval in minutes |

## 2.6 WriteOrder: Examples

### 2.6.1 Calling WriteOrder with argument -h

Calling up WriteOrder with the argument -h (for help)

```
java -Djava.security.policy==bcommjava.policy -cp bcommwriteorder.jar  
benzing.bcomm.writeorder.WriteOrder -h
```

creates the following output:

Usages for WriteOrder:

WriteOrder [OPTIONS] ARGUMENTS ORDER

OPTIONS:

-i: IP address of B-COMM Server

-p: RMI port of B-COMM Server

ARGUMENTS:

-h: Displays this help and exits.

-h Order command: e.g. ShutdownStream. Displays help about given order command and exits.

-v: Displays program version and exits.

-c: Client name: e.g. 'dormakaba'

-s: Stream name: e.g. 'bclan08'

ORDER:

-oc: Order command

Valid values for an order command for a Terminal stream with according parameters are:

'DataBufferControl', 'DataRead', 'DateTimeWrite', 'DaylightSavingTime', 'EncryptionKeyChange',  
'SSHKeyReplacement', 'SSHKeyIDRemoval', 'MasterRecordDownload', 'MasterRecordUpload',  
'OperatingModeWrite', 'ParameterDownload', 'ParameterUpload',  
'LEGICUpdateRecordDownload', 'LEGICUpdateRecordUpload',  
'LEGICValidationRecordDownload', 'LEGICValidationRecordUpload', 'UpdateRecordDownload',  
'UpdateRecordUpload', 'ValidationRecordDownload', 'ValidationRecordUpload',  
'ReorganizeOrderFile', 'TransparentDataDownload', 'ShutdownStream' Pseudo orders  
'StartupStream', 'ReadStatusStream', 'RenameCurrentBookingFile'!!!

Valid values for an order command for a FingerTemplate stream with according parameters are:

'TemplateReplication', 'TemplateReplicationMRBased', 'TemplateIDUpload', 'TemplateDelete',  
'ShutdownStream'. Pseudo orders 'StartupStream', 'ReadStatusStream',  
'RefreshTemplateList'!!!

Valid values for an order command for a SmartCom stream with according parameters are:

'LogUploadActuator', 'ParameterDownloadActuator', 'StatusUploadActuator', 'ShutdownStream',  
Pseudo orders 'StartupStream', 'ReadStatusStream'!!!

Valid values for an order command for a Wireless stream (CardLink & Whitelist) with according parameters are:

'DateTimeGateway' 'DateTimeGatewayGet' 'DoorControlWirelessActuatorGateway'  
'IdentifyWirelessActuatorGateway' 'ParameterDownloadGateway'  
'ParameterDownloadWirelessActuatorGateway' 'ReadDoorWirelessActuatorGateway'  
'ReadLinkQualityWirelessActuatorGateway' 'StatusUploadWirelessActuatorGateway'  
'TracebackUploadWirelessActuatorGateway' Pseudo orders 'StartupStream',  
'ReadStatusStream'!!!

Valid values for an order command for a Wireless stream (CardLink only) with according parameters are:

'BlacklistDeleteWirelessActuatorGateway' 'BlacklistDownloadWirelessActuatorGateway'  
Pseudo orders 'StartupStream', 'ReadStatusStream'!!!

Valid values for an order command for a Wireless stream (Whitelist only) with according parameters are:

'WhitelistDeleteWirelessActuatorGateway' 'WhitelistDownloadWirelessActuatorGateway'  
Pseudo orders 'StartupStream', 'ReadStatusStream'!!!

## 2.6.2 Calling WriteOrder with argument -h DateTimeWrite

Calling up WriteOrder with the argument -h DateTimeWrite for help regarding the order DateTimeWrite

```
java -Djava.security.policy==bcommjava.policy -cp bcommwriteorder.jar
benzing.bcomm.writeorder.WriteOrder -h DateTimeWrite
```

creates the following output:

Usages for WriteOrder:

WriteOrder [OPTIONS] ARGUMENTS ORDER

OPTIONS:

-i: IP address of B-COMM Server

-p: RMI port of B-COMM Server

ARGUMENTS:

-h: Displays this help and exits.

-h Order command: e.g. ShutdownStream. Displays help about given order command and exits.

-v: Displays program version and exit.

-c: Client name: e.g. 'dormakaba'

-s: Stream name: e.g. 'bclan08' ORDER:

-oc DateTimeWrite -os SubCommand (-ov AlternativeID -og GID -od DID) [-oy Priority] [- osa SchedulerActivated -oss SchedulerStartTime -osi SchedulerIntervall]

Valid value for SubCommand is:

SystemSeconds: sends system time in the format YYMMDDhhmmss

Valid values for AlternativeID are:

-1: don't use an alternative ID

0001 - 9999: order to terminal with this alternative ID

\*: order to all terminals. Attention: Enclose '\*' in '"', ("\*")

Valid values for GID are:

0 - 29: order to terminal with this group ID

\*: order to all groups of this stream. Attention: Enclose '\*' in '"', ("\*")

Valid values for DID are:

0 - 60: order to terminal with this device ID

\*: order to all terminals with defined group ID. Attention: Enclose '\*' in '"', ("\*") Specify a terminal either by AlternativeID or GID and DID!

Valid values for priority are:

1: high

2: normal

Valid values for SchedulerActivated are:

true: enables the scheduler for this order

false: disables the scheduler for this order

Valid values for SchedulerStartTime are:

any time in format HHMM

Valid values for SchedulerIntervall are:

0 - 1440 minutes: whereas 1440 minutes are 24 hours

0: order is executed one-time at SchedulerStartTime

> 0 < 1440: order is executed at SchedulerStartTime and then repetitive in this intervall

1440: order is executed daily at SchedulerStartTime

SchedulerStartTime and SchedulerIntervall are recognized only if SchedulerActivated is set to 'true'

Further information about orders can be found in B-COMM manual

### 2.6.3 Writing an order DateTimeWrite

```
java -Djava.security.policy=bcommjava.policy -cp bcommwriteorder.jar  
benzing.bcomm.writeorder.WriteOrder -c Kaba -s bclan08 -oc DateTimeWrite -os  
SystemSeconds -og "*" -od "*" -osa true -oss 0600 -osi 1440
```

The call shown writes an order DateTimeWrite for the stream bclan08 of the Kaba client.

This order applies to all terminals connected to this stream (GID and DID = "\*").

The order is executed daily (interval = 1440 minutes) at 6:00 a.m. (start time = 0600).

## 3 Communication B-COMM IO/ Application

### 3.1 Introduction

The communication between B-COMM IO and the application is carried out on the basis of TCP/IP. This allows installing B-COMM IO and the application on different computers. The data format is Unicode Big Endian unmarked. Although TCP/IP is already working connection-oriented, the communication between B-COMM IO and application has been implemented so that the transferred data packets must be acknowledged. Thus, every data packet sent via TCP/IP must first be acknowledged before the next data packet is sent.

This makes sure that the data packets are really received by the application or B-COMM IO. Without this "auxiliary protocol", it would only be ensured by the safety mechanism of TCP/IP that the data packets are lying on the respective IP stack. Whether or when the data packets are really received by the application or by B-COMM IO, would not be defined.

### 3.2 B-COMM ApplicationDemo

During setup, the installation of a demo application can be selected.

Hereby, the file

`bcommappdemoswp.jar` is copied to `../B-COMM/ApplicationDemo/bin`

and the source code of `ApplicationDemo` is copied to `../B-COMM/ApplicationDemo/src`  
the demo application

These files can serve as base for developing the user's own applications.

Calling the demo application:

- Change to the `/bin` directory.
- Entry: `-cp bcommappdemoswp.jar applicationdemo.ApplicationDemo -i<IP address> -p<Port number> -v`

The variables have the following meaning:

- |                                      |                                                                                                                   |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>-i &lt;IP address&gt;</code>   | IP address of the computer on which ApplicationDemo should run. If it is the local host, no entry is necessary.   |
| <code>-p &lt;Port address&gt;</code> | Port address of the computer on which ApplicationDemo should run. If it is the local host, no entry is necessary. |
| <code>-v</code>                      | If the parameter is indicated, the outputs of ApplicationDemo on the console are more detailed.                   |

### 3.3 B-COMM Class documentation

The module "Class documentation" contains an `.html` file with descriptions of the Java classes, methods and properties used in B-COMM Server. They can be called from an application written in Java via RMI on the B-COMM Server.

The file is stored during installation in the directory `Javadoc` of the installation directory.



## 3.4 Data classification

The data packets can be divided into three groups:

- Commands
- Acknowledgements
- Data

### 3.4.1 Commands

There are two types of commands from B-COMM IO to the application:

| Command | Syntax                | Example              | Meaning                                                                                            |
|---------|-----------------------|----------------------|----------------------------------------------------------------------------------------------------|
| Log-on  | <-c client -s stream> | <-c Kaba -s bclan08> | When starting up, B-COMM IO logs on in the application with the name of the client and the stream. |
| Log-off | <-e>                  |                      | B-COMM IO logs off from the application when shutting down                                         |

Commands are marked up in angle brackets ('<' and '>').

### 3.4.2 Acknowledgements

There are three different types of acknowledgements

| Acknowledgement | Syntax | Meaning                                                                                        |
|-----------------|--------|------------------------------------------------------------------------------------------------|
| ACK             | <ACK>  | The transmitted command/data packet has been received and acknowledged with ACK.               |
| NAK             | <NAK>  | The transmitted command/data packet has <b>not</b> been received and declined with NAK or CAN. |
| CAN             | <CAN>  |                                                                                                |

Acknowledgements are marked up in angle brackets ('<' and '>').

### 3.4.3 Data

|                 |                                |
|-----------------|--------------------------------|
| <b>Syntax</b>   | Data                           |
| <b>Example:</b> | QD B21060207171900200050150744 |
| <b>Example:</b> | QD R1Booking accepted!         |

Data is **not** marked up.

## 3.5 Communication examples

### 3.5.1 Error-free online communication

|   | Specification   | Command, acknowledgement, data | Direction |     |     |
|---|-----------------|--------------------------------|-----------|-----|-----|
| 1 | Log-on          | <-c Kaba -s bclan08>           | IO        | --> | App |
| 2 | Acknowledgement | <ACK>                          | App       | --> | IO  |
| 3 | Data            | @@ B21.....44                  | IO        | --> | App |
| 4 | Acknowledgement | <ACK>                          | App       | --> | IO  |
| 5 | Data            | @@ R1Booking accepted!         | App       | --> | IO  |
| 6 | Acknowledgement | <ACK>                          | IO        | --> | App |
| 7 | Log-off         | <-e>                           | IO        | --> | App |
| 8 | Acknowledgement | <ACK>                          | App       | --> | IO  |

#### Explanations

|   |                                                                                                                                                                              |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | When starting up, B-COMM IO logs on in the application with the name of the client and the stream.                                                                           |
| 2 | The application acknowledges with ACK.<br>This finishes the log-on period, and data can be transmitted in both directions.                                                   |
| 3 | B-COMM IO has received a data record from a terminal and sends it to the application.                                                                                        |
| 4 | The application acknowledges with ACK.<br>Thus, this data record is considered as transmitted, and B-COMM IO will acknowledge this data record as positive for the terminal. |
| 5 | The application has a data record for a terminal and sends it to B-COMM IO.<br>B-COMM IO will send this data record to the corresponding terminal.                           |
| 6 | If the data record has been transmitted successfully to the terminal, B-COMM IO acknowledges this data record with ACK.                                                      |
| 7 | B-COMM IO logs off.                                                                                                                                                          |
| 8 | The application acknowledges with ACK.                                                                                                                                       |

### 3.5.2 Error-free offline communication

|   | Specification   | Command, acknowledgement, data | Direction |     |     |
|---|-----------------|--------------------------------|-----------|-----|-----|
| 1 | Log-on          | <-c Kaba -s bclan08>           | IO        | --> | App |
| 2 | Acknowledgement | <ACK>                          | App       | --> | IO  |
| 3 | Data            | @@ B21.....44                  | IO        | --> | App |
| 4 | Acknowledgement | <ACK>                          | App       | --> | IO  |
| 5 | Log-off         | <-e>                           | IO        | --> | App |
| 6 | Acknowledgement | <ACK>                          | App       | --> | IO  |

#### Explanations

|   |                                                                                                                                                                              |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | When starting up, B-COMM IO logs on in the application with the name of the client and the stream.                                                                           |
| 2 | The application acknowledges with ACK.<br>This finishes the log-on period, and data can be transmitted in both directions.                                                   |
| 3 | B-COMM IO has received a data record from a terminal and sends it to the application.                                                                                        |
| 4 | The application acknowledges with ACK.<br>Thus, this data record is considered as transmitted, and B-COMM IO will acknowledge this data record as positive for the terminal. |
| 5 | B-COMM IO logs off.                                                                                                                                                          |
| 6 | The application acknowledges with ACK.                                                                                                                                       |

### 3.5.3 Faulty communication

|                      | Specification   | Command, acknowledgement, data | Direction |     |     |
|----------------------|-----------------|--------------------------------|-----------|-----|-----|
| 1                    | Log-on          | <-c Kaba -s bclan08>           | IO        | --> | App |
| 2                    | Acknowledgement | <ACK>                          | App       | --> | IO  |
| further data traffic |                 |                                |           |     |     |
| n+1                  | Data            | @A B210602....744              | IO        | --> | App |
| n+2                  | Acknowledgement | <NAK>                          | App       | --> | IO  |
| n+3                  | Data            | @A B210602....744              | IO        | --> | App |
| n+4                  | Acknowledgement | <ACK>                          | App       | --> | IO  |
| n+5                  | Data            | @A R1Booking accepted!         | App       | --> | IO  |
| n+6                  | Acknowledgement | <ACK>                          | IO        | --> | App |

#### Explanations

|     |                                                                                                                                                                                                                                                                          |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n+1 | B-COMM IO has received a data record from a terminal and sends it to the application.                                                                                                                                                                                    |
| n+2 | The application acknowledges this with NAK.<br>Thus, this data record is considered as <b>not</b> transmitted, and B-COMM IO will not acknowledge this data record for the terminal. The terminal will send this data record again to B-COMM IO at the next opportunity. |
| n+3 | The terminal will send the data record not acknowledged before again to B-COMM IO at the next opportunity. B-COMM IO will send it then to the application.                                                                                                               |
| n+4 | The application acknowledges with ACK.<br>Thus, this data record is considered as transmitted, and B-COMM IO will acknowledge this data record as positive for the terminal.                                                                                             |
| n+5 | The application has a data record for a terminal and sends it to B-COMM IO. B-COMM IO will send this data record to the corresponding terminal.                                                                                                                          |
| n+6 | If the data record has been transmitted successfully to the terminal, B-COMM IO acknowledges this data record with ACK.                                                                                                                                                  |

### 3.5.4 Conflict-raising communication

|                      | Specification   | Command, acknowledgement, data | Direction |     |     |
|----------------------|-----------------|--------------------------------|-----------|-----|-----|
| 1                    | Log-on          | <-c Kaba -s bclan08>           | IO        | --> | App |
| 2                    | Acknowledgement | <ACK>                          | App       | --> | IO  |
| further data traffic |                 |                                |           |     |     |
| n+1                  | Data            | @@ B21060....888               | IO        | --> | App |
| n+2                  | Acknowledgement | <ACK>                          | App       | --> | IO  |
| n+3                  | Data            | @A B210602....744              | IO        | --> | App |
| n+4                  | Data            | @A R1Booking accepted!         | App       | --> | IO  |
| n+5                  | Acknowledgement | <ACK>                          | IO        | --> | App |
| further data traffic |                 |                                |           |     |     |
| m+1                  | Data            | @A B210602....744              | IO        | --> | App |
| m+2                  | Acknowledgement | <ACK>                          | App       | --> | IO  |
| further data traffic |                 |                                |           |     |     |

#### Explanations

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n+1 | B-COMM IO has received a data record from a terminal and sends it to the application.                                                                                                                                                                                                                                                                                                                                                                  |
| n+2 | The application acknowledges this with ACK.                                                                                                                                                                                                                                                                                                                                                                                                            |
| n+3 | B-COMM IO has received again a data record from a terminal and sends it to the application.                                                                                                                                                                                                                                                                                                                                                            |
| n+4 | <p>Instead of the acknowledgement, the application sends a data record.</p> <p>Thus, this data record is considered as not transmitted, and B-COMM IO will <b>not</b> acknowledge this data record for the terminal. B-COMM IO discards the sent data record and sends the received data record to the corresponding terminal.</p> <p>The application also must refuse the data record received from B-COMM IO and <b>must not acknowledge</b> it.</p> |
| n+5 | If the data record has been transmitted successfully to the terminal, B-COMM IO acknowledges this data record with ACK to the application.                                                                                                                                                                                                                                                                                                             |
| m+1 | The terminal will send the data record not acknowledged before again to B-COMM IO at the next opportunity. B-COMM IO will send it then to the application.                                                                                                                                                                                                                                                                                             |
| m+2 | <p>The application acknowledges this with ACK.</p> <p>Thus, this data record is considered as transmitted, and B-COMM IO will acknowledge this data record as positive for the terminal.</p>                                                                                                                                                                                                                                                           |



### 3.6.1.2 log\_level

This parameter states the log level defined for the event.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="815" counter="35450" event_source="3" groups="100,101" log_level="5"
message_id="15" priority="NORMAL" timestamp="2022-10-17T05:27:04.551">
  <variable-body>
    <entry key="uid" value="000000000000000007F9"/>
    <entry key="guid" value="A5BFE391-991D-40C1-94C5-0A09E6EDCF25"/>
    <entry key="event_code" value="9"/>
    <entry key="index" value="29"/>
    <entry key="media_information" value="6"/>
    <entry key="event_timestamp" value="1665991623000"/>
    <entry key="additional_information" value="0"/>
  </variable-body>
</event>
```

| Value | Log level     |
|-------|---------------|
| 0     | Emergency     |
| 1     | Alert         |
| 2     | Critical      |
| 3     | Error         |
| 4     | Warning       |
| 5     | Notice        |
| 6     | Informational |
| 7     | Debug         |

### 3.6.1.3 message\_id

This parameter gives more detailed information about the event occurred.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="815" counter="35450" event_source="3" groups="100,101"
log_level="5" message_id="15" priority="NORMAL" timestamp="2022-10-17T05:27:04.551">
  <variable-body>
    <entry key="uid" value="0000000000000000007F9"/>
    <entry key="guid" value="A5BFKE391-991D-40C1-94C5-0A09E6EDCF25"/>
    <entry key="event_code" value="9"/>
    <entry key="index" value="29"/>
    <entry key="media_information" value="6"/>
    <entry key="event_timestamp" value="1665991623000"/>
    <entry key="additional_information" value="0"/>
  </variable-body>
</event>
```

| ID | Name                             | Description                                                                                                       |
|----|----------------------------------|-------------------------------------------------------------------------------------------------------------------|
| 1  | ZigBeeDeviceJoinInitializedEvent | Event generated if a component tries to join the gateway.                                                         |
| 2  | ZigBeeLinkQualityEvent           | A link quality event for the specified device was generated.                                                      |
| 3  | ProgressNotificationEvent        | Event generated if an asynchronous process has progressed.                                                        |
| 4  | EvolAssociateEvent               | An evolo component has joined the network, and is now ready to use.                                               |
| 5  | EvolBatteryAlarmEvent            | An evolo component reported its battery state to the gateway.                                                     |
| 6  | ZigBeeRevokePermitJoinEvent      | Event generated if the commissioning mode is no longer active.                                                    |
| 7  | ProcessTimeoutEvent              | An asynchronous process has reached its timeout limit without being successful.                                   |
| 8  | ZigBeeSendDataErrorEvent         | Sending data to a ZigBee device led to an error, the process is aborted.                                          |
| 9  | ProcessErrorEvent                | Event generated if an error occurred while performing an asynchronous process.                                    |
| 10 | EvolLogEvent                     | Event generated as a result of a GET request on the <code>*. {component}.configuration.reporting</code> resource. |
| 11 | DiagnosticEvent                  | This event is returned as a response to a GET request on <code>*.components. {component}.diagnostics</code> .     |
| 12 | ProcessSuccessEvent              | Event generated if an asynchronous process has completed successfully.                                            |
| 13 | ZigBeeReadErrorEvent             | An error happened while reading data from a component.                                                            |
| 14 | EvolPermittedEvent               | An evolo component has changed its state from UNPERMITTED to PERMIT_NOTIFIED.                                     |
| 15 | EvolNewLogIndicationEvent        | A new log entry has been generated on a component.                                                                |



| ID | Name                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16 | ProcessRetryEvent                | Event generated if a process was started again after a failure or timeout.                                                                                                                                                                                                                                                                                                                                                                                          |
| 17 | PersistentErrorEvent             | Something went wrong when saving a resource. Indicates a bug on the gateway.                                                                                                                                                                                                                                                                                                                                                                                        |
| 18 | EvoloDoorEvent                   | <p>There are four different sensors which can be connected to a door lock:</p> <ul style="list-style-type: none"> <li>• Frame contact (FC): door open or not?</li> <li>• Handle contact (HC): handle pressed or not?</li> <li>• Bolt contact (BC): bolt in frame or not?</li> <li>• Cylinder contact (CC): door opened using the cylinder?</li> </ul> <p>State changes of these contacts can generate messages to the system for logging and alarming purposes.</p> |
| 19 | SecondClientConnectedEvent       | Event generated if a second client tries to connect to the webservice of a gateway while another client was already connected.                                                                                                                                                                                                                                                                                                                                      |
| 20 | EvoloDoorStateEvent              | Event received upon issuing a GET request on the <code>*.{component}.door</code> resource.                                                                                                                                                                                                                                                                                                                                                                          |
| 21 | DuplicateGuidErrorEvent          | Event generated if a component is being joined which has the same GUID as a previously joined one.                                                                                                                                                                                                                                                                                                                                                                  |
| 22 | ZigBeeDeviceTimeSetEvent         | Event generated after successfully setting the time on a component.                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23 | MultiProcessErrorEvent           | Error event for an asynchronous process concerning multiple components. This event is currently fired if something goes wrong with the setting of time/getting the link quality on/from all components.                                                                                                                                                                                                                                                             |
| 24 | MultiProcessSuccessEvent         | Success event for an asynchronous process concerning multiple components.                                                                                                                                                                                                                                                                                                                                                                                           |
| 25 | ZigBeeMissingDeviceAnnounceEvent | ZigBee association failed, because no DeviceAnnounce-Message was received.                                                                                                                                                                                                                                                                                                                                                                                          |
| 26 | DatabaseCreatedEvent             | Event generated if the database of the gateway is created during the startup. This may happen after a factory reset.                                                                                                                                                                                                                                                                                                                                                |
| 27 | EvoloComponentDeletedEvent       | Event generated if a previously joined component is removed from the gateway database.                                                                                                                                                                                                                                                                                                                                                                              |

| ID | Name                          | Description                                                                                                                                                                                                           |
|----|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28 | IdentifyEvent                 | Event generated as a response to a GET on <code>*component</code> .<br><code>{component}.identification</code> .                                                                                                      |
| 29 | EvolComponentUnavailableEvent | Event generated if an evolo component is unavailable at a gateway.                                                                                                                                                    |
| 30 | ZigBeeMessageReceivedEvent    | Event generated if the gateway receives a message from a remote device.                                                                                                                                               |
| 31 | RangeMeasurementStartedEvent  | The gateway is equipped with a range measurement mode. This allows to position it optimally dependent of the component positions. When activating this mode a <code>RangeMeasurementStartedEvent</code> is generated. |
| 32 | RangeMeasurementStoppedEvent  | Event generated if the range measurement mode has been stopped                                                                                                                                                        |
| 33 | DoorSensorStateEvent          | Event generated as a result of a GET on <code>*.door.door-sensors</code> with the parameter <code>"getSensorState"</code> .                                                                                           |
| 34 | EvolComponentAvailableEvent   | Event generated if the communication with an evolo component can be established again after a period of time when the component was unavailable.                                                                      |
| 35 | GatewayUpdatedEvent           | Event generated after a successful update of the gateway firmware and the subsequent restart of the device.                                                                                                           |
| 36 | EvolComponentUpdatedEvent     | Event generated after a successful update of an evolo component containing the updated values of the versions of the different firmware on the component.                                                             |
| 37 | EvolLogConfigurationEvent     | Event generated as a response to an async GET on the reporting settings of a component (resource <code>*.{component}.configuration.reporting-settings</code> ).                                                       |
| 38 | AutoDeleteEvent               | Event generated if switching from CardLink-Mode to Whitelist-Mode or vice-versa triggering the deletion of data.                                                                                                      |
| 39 | ZigBeeRegisterEndpointEvent   | Event generated if gateway registers ZigBee endpoints on ZigBee interface.                                                                                                                                            |
| 40 | EvolIncompatibleEvent         | Event generated if an evolo component could not be joined because of its incompatibility.                                                                                                                             |
| 41 | DoorLockModeChangedEvent      | Indicates that the door lock mode of the referenced component has changed.                                                                                                                                            |
| 42 | SensorStateChangedEvent       | Indicates that a sensor state of the evolo component has changed.                                                                                                                                                     |
| 43 | ZigBeeAssociateEvent          | A ZigBee device has joined the network, and is now ready to use.                                                                                                                                                      |

| ID | Name                                     | Description                                                                                                                                                      |
|----|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 44 | ZigBeeAddressUpdateEvent                 | Event generated if the ZigBee address of a device changes.                                                                                                       |
| 45 | ProcessCancelEvent                       | Event generated if an asynchronous process is canceled.                                                                                                          |
| 46 | EvoloBatteryPercentageChangedEvent       | Indicates a change of the battery percentage of one of the commissioned components.                                                                              |
| 47 | GatewayStartedEvent                      | Event generated after a reboot of a gateway, the reason of the reboot is not contained in this event.                                                            |
| 48 | StreamingProcessTimeoutEvent             | Event generated after the timeout time for a streaming process was reached.                                                                                      |
| 49 | ComponentStatusNotificationReceivedEvent | Event triggered by the receiving of a status information from a component.                                                                                       |
| 50 | MediaTracebackNotificationReceivedEvent  | Event triggered by the receiving of media traceback information from a component (through reading the data from a user medium).                                  |
| 51 | CardlinkUpdateNotificationReceivedEvent  | Event triggered by the update of a medium with CardLink information.                                                                                             |
| 52 | CardlinkUpdateFunctionalityStatusEvent   | Event triggered by a change of the status of the CardLink update functionality.                                                                                  |
| 53 | ComponentStatusRemovedEvent              | Event triggered by the removal of a component status entry by the gateway.                                                                                       |
| 54 | MediaTracebackEntryRemovedEvent          | Event triggered by the removal of a media traceback entry by the gateway.                                                                                        |
| 55 | CardlinkUpdateNotificationRemovedEvent   | Event triggered by the removal of a CardLink update notification entry by the gateway.                                                                           |
| 56 | GatewayMemoryFullEvent                   | Event triggered in case the gateway memory is full.                                                                                                              |
| 57 | CardlinkUpdateDataRemovedFromReaderEvent | Event triggered in case the CardLink update data on a reader was deleted.                                                                                        |
| 58 | KeyConfigurationEvent                    | Event triggered by a change of the status of the key configurations list.                                                                                        |
| 59 | KeyConfigurationIndicationEvent          | Event triggered by a change of the status of the key configurations list.                                                                                        |
| 60 | MissedEventsInformationsEvent            | Event generated after a host connection in case the missed_events_mode is set to "DON'T_SEND_ON_CONNECT" (see <code>kate-api.gateway.reporting</code> resource). |
| 61 | EvoloConfigurationLostEvent              | Notifies about the loss of configuration data from a component.                                                                                                  |
| 62 | Reserved for internal use                | Reserved for internal use.                                                                                                                                       |

| ID | Name                                 | Description                                                                                                                                                                                                                                          |
|----|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 63 | RadioPowerNotificationEvent          | This event is only sent if explicitly requested by the <code>kate-api.gateway.zigbee-network.components.{component}.current-radio-power</code> resource.                                                                                             |
| 64 | Reserved for internal use            | Reserved for internal use.                                                                                                                                                                                                                           |
| 65 | RadioPowerReportingEvent             | This event is only sent if it is activated in the <code>kate-api.gateway.zigbee-network.components.{component}.configuration.energy-options</code> resource.                                                                                         |
| 66 | Reserved for internal use            | Reserved for internal use                                                                                                                                                                                                                            |
| 67 | VcpStatusIndicationEvent             | Event generated if the resource <code>kate-api.gateway.zigbee-network.components.{component}.mobile-access.versatile-configuration-parameter.vcp_data</code> is set and the component applied the VCP on the LEGIC chip                              |
| 68 | ProcessAliveEvent                    | Event generated if for asynchronous resources after 80% of their retry-time has passed and the processes are still alive.                                                                                                                            |
| 69 | LinkQualityEvent                     | Event is returned as a response to a GET request on <code>kate-api.gateway.zigbee-network.diagnosis.link-quality</code> .                                                                                                                            |
| 70 | IdsLockDownEvent                     | Event generated if a component has been locked or unlocked by <code>kate-api.gateway.zigbee-network.components.{component}.door.ids-lock-down.mode</code> by the ids user.                                                                           |
| 71 | EvoloAttributeNotSupportedEvent      | Event generated if an attribute is specified but not supported by the component.                                                                                                                                                                     |
| 72 | EvoloAttributeValueNotSupportedEvent | Event generated if a value for an attribute is specified but not supported by the component.                                                                                                                                                         |
| 73 | RouterUpdateAvailableEvent           | Event generated if a newer firmware version is available for the given router. The event can be generated after <ul style="list-style-type: none"> <li>gateway firmware update</li> <li>router firmware upload</li> <li>joining a router.</li> </ul> |
| 74 | RouterUpdateStartedEvent             | Event generated if a router update is started                                                                                                                                                                                                        |
| 75 | RouterUpdateFinishedEvent            | Event generated if a router update is finished.                                                                                                                                                                                                      |

| ID | Name                                | Description                                                                       |
|----|-------------------------------------|-----------------------------------------------------------------------------------|
| 76 | RouterUpdateProcessActivatedEvent   | Event generated if the process for updating one ore multiple routers starts.      |
| 77 | RouterUpdateProcessDeactivatedEvent | Event generated if the process for updating one ore multiple router has finished. |
| 78 | AttributeNotSupportedEvent          | Event generated if an attribute is specified but not supported by the gateway.    |
| 79 | EvoloComponentOutOfSyncEvent        | Event generated if there is data to be transferred to a component.                |
| 80 | EvoloComponentInSyncEvent           | Event generated if there is no more data to be transferred to a component.        |

### 3.6.2 Event 15 "EvolonNewLogIndication"

### 3.6.2.1 event\_code

The following table shows the different events codes possible with the event 15.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="815" counter="35450" event_source="3" groups="100,101"
log_level="5" message_id="15" priority="NORMAL" timestamp="2022-10-17T05:27:04.551">
  <variable-body>
    <entry key="uid" value="000000000000000007F9"/>
    <entry key="guid" value="A5BFE391-991D-40C1-94C5-0A09E6EDCF25"/>
    <entry key="event_code" value="9"/>
    <entry key="index" value="29"/>
    <entry key="media_information" value="6"/>
    <entry key="event_timestamp" value="1665991623000"/>
    <entry key="additional_information" value="0"/>
  </variable-body>
</event>
```

| Value hex (dec) | Event                                                                                  | Source                    |
|-----------------|----------------------------------------------------------------------------------------|---------------------------|
| 0x00 (0)        | Forbidden traceback value                                                              |                           |
| 0x01 (1)        | Whitelist authorised                                                                   | Medium                    |
| 0x02 (2)        | Whitelist not authorised                                                               | Medium                    |
| 0x03 (3)        | Whitelist wrong time                                                                   | Medium                    |
| 0x04 (4)        | Authorised with emergency power supply                                                 | Medium                    |
| 0x05 (5)        | Start media programming                                                                | Medium                    |
| 0x06 (6)        | Start programmer access                                                                | Programmer                |
| 0x07 (7)        | Clock adjusted                                                                         | Programmer                |
| 0x09 (9)        | TimePro Office Open                                                                    | Medium                    |
| 0x0A (10)       | TimePro Office Close (Medium or by S-Module or end of time slot or deactivation timer) | Medium, S-Module, TimePro |
| 0x0F (15)       | Close Hold Open Mode                                                                   | External switches         |
| 0x10 (16)       | Actuator reconfigured (reboot)                                                         | Programmer, AirLink       |
| 0x12 (18)       | Locking Module Error                                                                   | Various                   |
| 0x13 (19)       | Locking Module Error (Door is locked, a locking attempt was successful)                | Various                   |
| 0x14 (20)       | Coupling position incorrect                                                            | Various                   |
| 0x15 (21)       | Coupling position correct                                                              | Various                   |
| 0x19 (25)       | TimePro DayNight Open                                                                  | TimePro                   |
| 0x1A (26)       | TimePro DayNight (end of time slot or by S-Module)                                     | TimePro                   |
| 0x1B (27)       | Pass Mode Open                                                                         | External switches         |
| 0x1C (28)       | Pass Mode Close                                                                        | External switches         |
| 0x20 (32)       | SiteKey Modification (SiteKey added / deleted / changed)                               | Medium                    |
| 0x21 (33)       | SiteKey Modification Error                                                             | Medium                    |
| 0x24 (36)       | PIN not authorised                                                                     | Medium / PIN Pad          |
| 0x25 (37)       | PIN input timeout                                                                      | Medium / PIN Pad          |

| Value hex (dec) | Event                                                                          | Source           |
|-----------------|--------------------------------------------------------------------------------|------------------|
| 0x26 (38)       | No PIN on Media                                                                | Medium / PIN Pad |
| 0x27 (39)       | PIN pad not connected                                                          | Medium / PIN Pad |
| 0x28 (40)       | PIN pad pairing successful                                                     | Programmer       |
| 0x29 (41)       | PIN pad pairing not successful                                                 | Programmer       |
| 0x30 (48)       | Initialisation (Advant/Prime)                                                  | Medium           |
| 0x31 (49)       | Deinitialisation (Advant/Prime)                                                | Medium           |
| 0x35 (53)       | VCP OK, everything saved                                                       | Medium / Mobile  |
| 0x36 (54)       | VCP Error: Generic, not further specifiable                                    | Medium / Mobile  |
| 0x37 (55)       | VCP Error: Wrong password                                                      | Medium / Mobile  |
| 0x38 (56)       | VCP Error: Custom Data Format incorrect (wrong length, format)                 | Medium / Mobile  |
| 0x39 (57)       | VCP Error: Key store full                                                      | Medium / Mobile  |
| 0x3A (58)       | VCP Error: KeySet ProjectID wrong (does not match project ID from database)    | Medium / Mobile  |
| 0x3B (59)       | VCP Error: Wrong VCP key (VCP from other LEGIC Connect Company)                | Medium / Mobile  |
| 0x3D (61)       | VCP Error: Admin is already set                                                | Medium / Mobile  |
| 0x3E (62)       | VCP Error: No admin set                                                        | Medium / Mobile  |
| 0x3F (63)       | VCP Error: Wrong LEGIC Chip ID                                                 | Medium / Mobile  |
| 0x40 (64)       | S-Module: Start "Open" mode                                                    | S-Module         |
| 0x41 (65)       | S-Module: Start "Closed" mode                                                  | S-Module         |
| 0x42 (66)       | S-Module: Start "Any Medium" mode                                              | S-Module         |
| 0x43 (67)       | S-Module: End S-Module action                                                  | S-Module         |
| 0x45 (69)       | S-Module: Not authorised because "Closed" Mode or "Authorised Medium" mode off | Medium           |
| 0x46 (70)       | S-Module: Authorised because in "Any Medium" mode                              | Medium           |
| 0x47 (71)       | S-Module: Start "Authorised Medium" mode (no longer exists in V4)              | S-Module         |
| 0x48 (72)       | S-Module: Start "TimePro Switch Off" mode                                      | S-Module         |
| 0x50 (80)       | Modification: Successful                                                       | Medium           |
| 0x57 (87)       | Modification: Blacklist full                                                   | Medium           |
| 0x58 (88)       | Modification: Not allowed                                                      | Medium           |
| 0x59 (89)       | Validation Update: Successful                                                  | Medium           |
| 0x5A (90)       | Validation Update: Error, not authorised                                       | Medium           |
| 0x5B (91)       | Validation Update: Error, wrong time                                           | Medium           |
| 0x5C (92)       | Validation Update: Error, validation expired                                   | Medium           |
| 0x5D (93)       | CardLink Update: Successful                                                    | Medium           |
| 0x5E (94)       | CardLink Update: Failed                                                        | Medium           |

| Value hex (dec) | Event                                                                  | Source            |
|-----------------|------------------------------------------------------------------------|-------------------|
| 0x60 (96)       | CardLink / AoC Configuration / OSS / MobileLink: not authorised        | Medium            |
| 0x61 (97)       | CardLink / AoC Configuration / OSS / MobileLink: validation invalid    | Medium            |
| 0x62 (98)       | CardLink / AoC Configuration / OSS: Wrong administration area          | Medium            |
| 0x63 (99)       | CardLink / AoC Configuration / OSS / MobileLink: wrong time            | Medium            |
| 0x65 (101)      | CardLink: Blacklisted media with validation terminal: Header deleted   | Medium            |
| 0x6A (106)      | CardLink / AoC Configuration / OSS: authorised (individual permission) | Medium            |
| 0x6B (107)      | CardLink / AoC Configuration / OSS: authorised (door group permission) | Medium            |
| 0x6C (108)      | CardLink / MobileLink: authorised (Reservation)                        | Medium            |
| 0x6D (109)      | Not authorised, wrong binding                                          | Medium            |
| 0x6E (110)      | CardLink / MobileLink: not authorised, corrupted file                  | Medium            |
| 0x71 (113)      | CabinetLock: Opening / Closing by maintenance medium                   | Medium            |
| 0x73 (115)      | CabinetLock: File / segment not available or no space free             | Medium            |
| 0x74 (116)      | CabinetLock: Locking error activation switch                           | Various           |
| 0x75 (117)      | CabinetLock: Alarm triggered external switches                         | External switches |
| 0x76 (118)      | CabinetLock: Manual closing without medium                             | External switches |
| 0x80 (128)      | PairedList: paired RCID medium authorised (UID of the RCID medium)     | Medium (RCID)     |
| 0x81 (129)      | PairedList: RCID medium not authorised (UID of the RCID medium)        | Medium (RCID)     |
| 0x82 (130)      | PairedList: Validation invalid (UID of the RCID medium)                | Medium (RCID)     |
| 0x83 (131)      | PairedList: wrong time (UID of the RCID medium)                        | Medium (RCID)     |
| 0x85 (133)      | PairedList: Medium in blacklist (UID of the RCID medium)               | Medium (RCID)     |
| 0x8A (138)      | Commissioning started (join triggered)                                 | Medium            |
| 0x8B (139)      | Commissioning successful                                               | AirLink           |
| 0x8C (140)      | Commissioning not successful                                           | AirLink           |
| 0x8D (141)      | Leave triggered                                                        | Medium            |
| 0x8E (142)      | Leave executed                                                         | AirLink           |
| 0x90 (144)      | Anti-Amok: Panic mode activated                                        | External switches |
| 0x91 (145)      | Anti-Amok: Panic mode deactivated                                      | Medium            |
| 0x95 (149)      | Escape Return: Escape return activated                                 | External switches |



| Value hex (dec)          | Event                                                                                                                 | Source            |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------------|
| 0x96 (150)               | Escape Return: Close with push-button                                                                                 | External switches |
| 0x97 (151)               | Authorised close                                                                                                      | Medium            |
| 0x98 (152)               | Authorised always open                                                                                                | Medium            |
| 0x99 (153)               | Remote Lock IDS                                                                                                       | AirLink           |
| 0x9A (154)               | Remote Opening (unlock)                                                                                               | AirLink           |
| 0x9B (155)               | Remote Access profile check (access control)                                                                          | AirLink           |
| 0x9C (156)               | Remote Lock (shutdown)                                                                                                | AirLink           |
| 0x9D (157)               | Remote Normal operation (normal)                                                                                      | AirLink           |
| 0x9E (158)               | Remote One time opening (open once)                                                                                   | AirLink           |
| 0x9F (159)               | Remote Not authorised because in "Shutdown" or "IDS" Mode.                                                            | AirLink           |
| 0xA0 (160) to 0xAF (175) | Debug traceback entries of the development, different depending on the version. Must not appear in released versions. |                   |
| 0xB0 (176)               | Door forced open (door status monitoring)                                                                             | AirLink           |
| 0xB1 (177)               | Basic event for MultiOutputReader                                                                                     |                   |
| 0xC4 (196)               | Licence upgrade SL1 SL2                                                                                               | Medium            |
| 0xC5 (197)               | Licence upgrade SL1 SL3                                                                                               | Medium            |
| 0xC6 (198)               | Licence upgrade SL2 SL3                                                                                               | Medium            |
| 0xC7 (199)               | Licence upgrade SL4 SL3                                                                                               | Medium            |
| 0xC8 (200)               | Licence upgrade Bluetooth Low Energy                                                                                  | Medium            |
| 0xD0 (208)               | Changing the battery via special medium / Switching on "Adjusting the battery warning limits"                         |                   |
| 0xD1 (209)               | Changing the battery via PD1460 / Switching on "Adjusting the battery warning limits"                                 |                   |
| 0xD2 (210)               | Changing the battery via internal / Switching on "Adjusting the battery warning limits"                               |                   |
| 0xD3 (211)               | Changing the battery via ZigBee/Wireless / Switching on "Adjustment of the battery warning limits"                    |                   |
| 0xD4 (212)               | RFU / Switching on "Adjusting the battery warning limits"                                                             |                   |
| 0xD5 (213)               | Switch off "Adjustment of battery warning limits"                                                                     |                   |
| 0xD6 (214)               | Activation "Adjustment of the battery warning limits"                                                                 |                   |
| 0xD7 (215)               | Deactivation "Adjustment of battery warning limits"                                                                   |                   |

| Value hex<br>(dec) | Event                                                                   | Source |
|--------------------|-------------------------------------------------------------------------|--------|
| 0xE0 (224)         | Online Access: authorised                                               | Medium |
| 0xE1 (225)         | Online Access: not authorised                                           | Medium |
| 0xE2 (226)         | Online Access: wrong time                                               | Medium |
| 0xE3 (227)         | Online Access: error                                                    | Medium |
| 0xE4 (228)         | Cache Access: authorised                                                | Medium |
| 0xE5 (229)         | Cache Access: wrong time                                                | Medium |
| 0xE6 (230)         | Cache Access: not authorised                                            | Medium |
| 0xF0 (240)         | CardLink / AoC Configuration / OSS / MobileLink:<br>Medium in Blacklist | Medium |

### 3.6.2.2 media\_information

This parameter gives more detailed information about the media used.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="815" counter="35450" event_source="3" groups="100,101"
log_level="5" message_id="15" priority="NORMAL" timestamp="2022-10-17T05:27:04.551">
  <variable-body>
    <entry key="uid" value="0000000000000000007F9"/>
    <entry key="guid" value="A5BF6391-991D-40C1-94C5-0A09E6EDCF25"/>
    <entry key="event_code" value="9"/>
    <entry key="index" value="29"/>
    <entry key="media_information" value="6"/>
    <entry key="event_timestamp" value="1665991623000"/>
    <entry key="additional_information" value="0"/>
  </variable-body>
</event>
```

| Bit 0-3 | Description (media type)           |
|---------|------------------------------------|
| 0       | Unknown                            |
| 1       | Master A                           |
| 2       | Master B                           |
| 3       | User medium                        |
| 4       | S-Module                           |
| 5       | Timer                              |
| 6       | User medium with CID               |
| 7       | Master T (temporary master)        |
| 8       | LEGIC functional medium (SAM63/64) |
| 9       | AIDA (wireless)                    |
| 10      | Battery                            |
| 11      | PIN pad                            |

### Evolvo components with Mifare:

| Bit 4-7 | Description (media technology) |
|---------|--------------------------------|
| 0       | Unknown / irrelevant           |
| 1       | Classic                        |
| 2       | Desfire                        |
| 3       | Ultralight                     |
| 15      | All other ISO media            |

**Evolo components with LEGIC advant:**

| Bit 4-7 | Description (media technology) |
|---------|--------------------------------|
| 0       | Unknown / irrelevant           |
| 5       | LEGIC prime                    |
| 6       | LEGIC advant                   |
| 15      | All other ISO media            |

**Evolo components with Mifare and LEGIC advant:**

| Bit 4-7 | Description (media technology)    |
|---------|-----------------------------------|
| 0       | Unknown / irrelevant              |
| 1       | Classic                           |
| 2       | Desfire                           |
| 3       | Ultralight                        |
| 4       | Felica                            |
| 5       | LEGIC prime                       |
| 6       | LEGIC advant                      |
| 7       | Mifare plus                       |
| 8       | Mobile                            |
| 9       | HID                               |
| 14      | Don't care (is the case with OSS) |
| 15      | All other media                   |

### 3.6.2.3 additional\_information

This parameter gives additional information about various items.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="815" counter="35450" event_source="3" groups="100,101"
log_level="5" message_id="15" priority="NORMAL" timestamp="2022-10-17T05:27:04.551">
  <variable-body>
    <entry key="uid" value="00000000000000007F9"/>
    <entry key="guid" value="A5BFE391-991D-40C1-94C5-0A09E6EDCF25"/>
    <entry key="event_code" value="9"/>
    <entry key="index" value="29"/>
    <entry key="media_information" value="6"/>
    <entry key="event_timestamp" value="1665991623000"/>
    <entry key="additional_information" value="0"/>
  </variable-body>
</event>
```

| Bit | Description                                                                                                                                                                                                                      |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0   | <b>MultiOutputReader event</b><br>Flag to indicate that the following traceback entry should be an Event 0xB1 for the MultiOutputReader to indicate which outputs are involved. This flag is only used in the MultiOutputReader. |
| 1   | <b>With PIN verification</b><br>Flag to indicate that the following traceback entry represents an successful access with a pin validation. This flag is only used in combination with a Pin pad.                                 |
| 2   | <b>PIN Pad low battery</b><br>1 if a battery low was detected during the action, otherwise 0.                                                                                                                                    |
| 3   | <b>Online</b><br>Only for AIDA components:<br>0 if offline (no connection to the gateway),<br>1 if connection to the gateway is available.                                                                                       |
| 4   | <b>Hall sensor state</b><br>Digital cylinder only [Cirrus].<br>1 if the Hall sensor is in the "down" position (coupled in), otherwise 0.                                                                                         |
| 5   | <b>Low battery</b><br>1 if a battery low was detected during the action, otherwise 0.                                                                                                                                            |
| 6   | <b>Clock error</b><br>1 if a clock error was detected during the action, otherwise 0.                                                                                                                                            |
| 7   | <b>Door state</b><br>Function primarily for CabinetLock:<br>1 if the door was open (closing) during the action,<br>0 if closed (opening).                                                                                        |

### 3.6.3 Event 18 "EvolDoorEvent"

#### 3.6.3.1 state

**Legend:**

FC = **F**rame **C**ontact

HC = **H**andle **C**ontact

BC = **B**olt **C**ontact

CC = **C**ylinder **C**ontact

**\*M** = Mandatory seonsor for this message

**\*O** = Optional sensor (supports message identification while not necessarily needed)

Empty: Not needed for the message.

| Value hex (dec) | Name                               | Description                                                                                 | FC | HC | BC | CC |
|-----------------|------------------------------------|---------------------------------------------------------------------------------------------|----|----|----|----|
| 0x00 (0)        | Access allowed                     | Access authorized                                                                           |    |    |    |    |
| 0x01 (1)        | Access denied                      | Access denied                                                                               |    |    |    |    |
| 0x02 (2)        | Access authorized with door handle | Inner door handle pressed                                                                   |    | *M |    |    |
| 0x03 (3)        | Access authorized with key         | Key turned                                                                                  |    |    |    | *M |
| 0x10 (16)       | Passage completed                  | Door closed after an authorized access and passage.                                         | *M |    | *M |    |
| 0x11 (17)       | Passage not completed              | Even there was an authorized access, the door was not opened.                               | *M | *M |    |    |
| 0x20 (32)       | Door open                          | Door was opened                                                                             | *M |    |    |    |
| 0x21 (33)       | Door open too long                 | Alarm time expired, the door has been open too long.                                        | *M | *O |    |    |
| 0x22 (34)       | Door closed                        | Door was closed                                                                             | *M |    |    |    |
| 0x23 (35)       | Door closed again                  | Door was closed after an alarm (door open too long (0x21) or door forced (0x30) was issued. | *M |    |    |    |
| 0x30 (48)       | Door forced                        | Door is open even though it shouldn't be.                                                   | *M | *M | *M | *M |
| 0x31 (49)       | Door forced with bolt in position  | Door was closed even though the bolt was out (bolt position fault (0x34).                   | *M |    | *M |    |
| 0x32 (50)       | Door locked                        | Door is locked using the bolt                                                               | *M |    | *M |    |
| 0x33 (51)       | Door unlocked                      | Door is unlocked (bolt in)                                                                  | *M |    | *M |    |
| 0x34 (52)       | Bolt position fault                | Bolt is not where it should be                                                              | *M |    | *M |    |
| 0x35 (53)       | Bolt in position                   | Bolt is back where it belongs to.                                                           | *M |    | *M |    |

### 3.6.4 Event 42 "SensorStateChanged"

#### 3.6.4.1 sensor

This parameter gives information about the sensor involved.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<event acknowledged_by_host="false" correlation_id="1101" counter="75545" event_source="3" groups="100,101"
log_level="5" message_id="42" priority="NORMAL" timestamp="2022-10-19T07:48:51.334">
  <variable-body>
    <entry key="guid" value="2496216D-A18B-4ABE-A9D5-8403D540378B"/>
    <entry key="sensor" value="1"/>
    <entry key="active" value="false"/>
  </variable-body>
</event>
```

| Value | Sensor           |
|-------|------------------|
| 0     | Frame Contact    |
| 1     | Handle Contact   |
| 2     | Bolt Contact     |
| 3     | Cylinder Contact |

## 4 B-COMM Helper

### 4.1 Introduction

B-COMM Helper is used as interface between B-COMM Server and an application **not written in Java**.

The communication is realized via TCP/IP and the default port 3001. Commands of the application are used that are acknowledged by the Helper with

- <ACK> (Acknowledged = accepted) or
- <NAK> (Not Acknowledged = not accepted)

Single commands are marked up in angle brackets < >.

The commands can be transmitted in XML format.

In the debug files, the characters CR (carriage return) and LF (line feed) are interpreted. The debug information therefore will be presented in a structured view. If all information shall be displayed in **one** line, these characters can be ignored. To do so, please tick the checkbox "Remove CR/LF in debug file(s)" in the "Global settings" menu.

The most frequent parameters are quoted as shown below:

|             |                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------|
| Client=     | Client name as used in B-COMM, e. g. <Customer name>                                                    |
| Stream=     | Stream name as used in B-COMM, e. g. BCLAN01                                                            |
| GID=        | Group address in decimal values, e. g. 01                                                               |
| DID=        | Device address in decimal values, e. g. 01                                                              |
| File=       | Name of a file, e. g. bclan01.booking                                                                   |
| LocalFile=  | Definition of a complete path, e. g. c:/test/testfile.booking                                           |
| DataRecord= | Definition of a complete data string with the GID/DID as ASCII characters in the beginning, e. g. @@... |
| Priority=   | Definition of the priority of the order, e. g. 1 or 2.                                                  |

If the text refers to chapters where more information on a topic can be found, these are chapters in the current B-COMM Reference Manual.

### 4.2 Calling the Helper

```
java -Djava.security.policy=bcommjavapolicy -cp bcommhelper.jar
benzing.bcomm.helper.BCOMMHelper[-a <AI port>] [-i <server>] [-p <RMI port>]
```

The variables have the following meaning:

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| [-a <AI port>]  | Port number for the communication with the application (default 3001)   |
| [-i <server>]   | Name or IP address of B-COMM server                                     |
| [-p <RMI port>] | Port number for the RMI communication with B-COMM server (default 1099) |

### 4.3 Helper commands with XML

The **answer** to an XML helper command always contains the entered command. If necessary, also the client and the channel will be returned in the answer.



## 4.4 Helper commands sorting

The large number of helper commands requires a sorting in order to find the needed helper command and its description quickly.

For this purpose, the single helper commands are summarized in functional units under a certain headline.

The following table gives an overview.

| Helper commands in functional units   | Chapter |
|---------------------------------------|---------|
| Starting and terminating of functions | 4.5     |
| File related                          | 4.6     |
| Configuration related                 | 4.7     |
| Client related                        | 4.8     |
| Stream related                        | 4.9     |
| Terminal related                      | 4.10    |
| Actuator related                      | 4.11    |
| Order related                         | 4.12    |
| Booking related                       | 4.13    |
| CardLink related                      | 4.14    |
| Whitelist related                     | 4.15    |
| CardLink wireless related             | 4.16    |
| Biometric related                     | 4.17    |
| Programmer related                    | 4.18    |
| elologic related                      | 4.19    |
| Mobile Access related                 | 4.20    |
| SSH key related                       | 4.21    |

## 4.5 Helper commands for starting and terminating of functions

### 4.5.1 ShutdownServer

Shuts down the B-COMM Server.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="ShutdownServer" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**or:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

### 4.5.2 ShutdownHelper

Shuts down the B-COMM Helper. After this, communication with the helper is no more possible.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="ShutdownHelper" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**or:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

### 4.5.3 StartStream

Starts the stream <stream> of the client <client>. The character \* can also be used as wildcard for <client> and <stream>. Then, either all streams of the client or all streams of all clients are started.

The response of the Helper is either <ACK> if a stream could be started or had already been started, or <NAK> if a stream could not be started.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="StartStream" Client="..." Stream="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**or**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

#### 4.5.4 StopStream

Ends the stream <stream> of the client <client>. The character \* can also be used as wildcard for <client> and <stream>. Then, either all streams of the client or all streams of all clients are stopped.

The response of the Helper is either <ACK> if the order for finishing the stream could be written, or <NAK> if the order could not be written.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="StopStream" Client="..." Stream="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

or

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

#### 4.5.5 SwitchApplicationInterface

Switches the application interface of the stream <stream> of the client <client> on or off, depending on the definition in the <status> parameter. Additionally, the IP address <ipaddress> of the computer the application runs on can be defined and the communication port number <port>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="SwitchApplicationInterface" Client="..." Stream="..." Status="..." IPAddress="..." Port="..." />
```

**Response, if the application interface could be switched over:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**Response, if the application interface could not be switched over:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" /> User example: <?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="SwitchApplicationInterface" Client="dormakaba" Stream="BCLAN01" Status="ON" IPAddress="192.168.2.1" Port="3005" />
```

The example of use switches ON the application interface for the stream <BCLAN01> of the client <dormakaba>. The application runs on the computer with the IP-address 192.168.2.1. The communication between the stream and the application is carried out via the port 3005.

## 4.6 File related Helper commands

### 4.6.1 DeleteBookingFile

Deletes the booking file <file> of the stream <stream> of the client <client>, where <file> is just a filename without any path definition, e.g. "bclan01.booking".

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="DeleteBookingFile"
Client="...." ...Stream="..."...File="..." />
```

**Response, if the booking file could be deleted:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**Response, if the booking file could not be deleted:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

### 4.6.2 DeleteOrderFile

Deletes the order file of the stream <stream> of the client <client>. The response of the Helper is either <ACK> if the order file could be deleted, or <NAK> if the order file could not be deleted.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="DeleteOrderFile"
Client="..." Stream="..." />
```

**Response, if the order file could be deleted:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**Response, if the order file could not be deleted:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```



#### NOTICE

This command deletes the order file independently from the activity status of the stream and independently from the processing status of the orders. Even orders with activated time scheduler will be deleted.

### 4.6.3 FileUpload

Transmits the file <localfile> from the computer, on which the application is running, as <remotefile> to the computer, on which B-COMM Server is running. For <localfile> and <remotefile>, the complete path must always be indicated.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand
Command="FileUpload"RemoteFile="..."LocalFile="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

or

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

### 4.6.4 FileUploadContent

The helper command FileUploadContent is an extension for the FileUpload command described above. The command FileUploadContent enables the data exchange between EACM and B-COMM, if B-COMM is used as B-COMM IoT 90 70 on a fitlet server. In this case the underlying operating system is not Windows but LINUX (this HTTPS interface can also be used by any other (partner) application).

Since EACM does not have direct file access to this, the data exchange is particularly useful for

- Traceback files (CardLink and Whitelist) and
- Log files (or any other file in the supported file formats, see explanation about the parameter "ContentType" at the end of this chapter).

In both cases, the data to be transferred is not transmitted as a file but as a data stream in XML or binary format.

The following chapters explain the necessary steps in detail.

#### 4.6.4.1 FileUploadContent: Uploading traceback files

EACM uses a 2-step procedure to import traceback files:

Step 1: Generation of a traceback file

EACM first sends a request to B COMM Server via B-COMM Helper (HTTP interface). This request contains all information that B-COMM needs to generate the traceback file. The path/file name of the respective traceback file is determined by EACM and transferred in the request.

The name of the generated traceback file is therefore known in EACM.

The file path for the traceback file to be generated is entered into the order by EACM relative (to the root directory of the B-COMM).

There are two subdirectories that come into question:

**Examples:**

C:\Kaba\B-COMM\Cardlink\Debug\  
C:\Kaba\B-COMM\Whitelist\Debug\

The file name of the traceback file is generated according to a name pattern.

**Example:**

**File name**

Component29W9S029-00001.xml  
ComponentXXYYYYYY-ZZZZZ.xml



- XX = 29 = Keyplan (Whitelist) or administration area (Cardlink)
- YYYYYY = W9S029 = Component's name
- ZZZZZ = 00001 = Consecutive number

Example of a Traceback request (HTTP Helper Call)

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteOrderDirect" Client="Whitelist"
Stream="BCWLS02"> <Order
Command="TracebackUploadWirelessActuatorGateway" AlternativeID="29" File=".
\whitelist\debug\component29W9S029-00001.xml " Extension=" W9S029-00001"
Data="10.10.7.173" />
```

Result in the order file:

```
CreationDate=20200212
CreationTime=104246818
Command=TracebackUploadWirelessActuatorGateway
Subcommand=
Extension=W9S029-00001
Status=FinishedOK
AlternativeID=29
GID=0
DID=0
File=.\whitelist\debug\component29W9S029-00001.xml
Data=10.10.7.173
Priority=2
QuartzTriggerPriority=5
SchedulerActivated=false
SchedulerStartTime=0000
NextSchedulerStartTime=----
NextSchedulerStartDate=20200212
SchedulerInterval=1440
SchedulerExecuted=false
StartDate=20200212
StartTime=104246844
EndDate=20200212
EndTime=104251597
FTPEntabled=false
TerminalsWithError=
Error=
```

## Step 2: Retrieving the traceback file

In the next step EACM requests a generated traceback file. This is done by the B-COMM Helper call *"FileUploadContent"*:

### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="FileUploadContent" RemoteFile=".
\whitelist\debug\component29W9S029-00001.xml" ContentType="*XML" />
```

Since the path/name of the traceback file is known in the EACM, the file name can be given directly with the *"FileUploadContent"*. In contrast to the helper command *"FileUpload"* the relative path can also be used here.

The parameter *"LocalFile="* can optionally be included in the call. This ensures that the XML data stream is stored in the specified file on the B-COMM Helper side.

### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="FileUploadContent" RemoteFile=".
\whitelist\debug\component29W9S029-00001.xml" ContentType="*XML"
LocalFile="data\tracebackdata\component29W9S029-00001.xml" />
```

### Note

The traceback file to be read must contain the element *<Upload>* so that the data can be displayed in a well-formed way. The element *<Upload>* is normally created automatically.

If this element is not in the traceback file, an error message is generated. In this case the parameter *"ContentType=XML"* must be omitted. This transfers the data in binary form.

### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="FileUploadContent" RemoteFile=".
\whitelist\debug\component29W9S029-00001.xml" />
```

#### 4.6.4.2 FileUploadContent: Upload of log or any other files

##### General notes

In addition to traceback files, it should also be possible to transfer any other files or file types. This means that these files can be very large under certain circumstances. Since it is not possible to transfer arbitrarily large amounts of data between applications, a possibility has been created to import files in blocks.

For this purpose the operations

- *"FIRST"* and *"NEXT"* for blockwise reading in forward direction and
- *"LAST"* and *"PREV"* for blockwise reading in reverse direction

have been implemented.

Blockwise reading requires several calls from the application program (e.g. the EACM client) to B-COMM Helper and B-COMM Server. The calls are made in a sequence. Starting with the second block (the second read access) the file position for the next block must be known. Since B-COMM Helper and B-COMM Server do not remember the file position for the next access, the information about the file position for the next block access must come from the calling application.

For this purpose, the XML attributes

- *"FilePosition"* and
- *"ResultPosition"*

have been implemented.

For text files (e.g. log files or .ini files), it can be useful to let the imported blocks end at the end of the line and to wrap to a new line.

This facilitates processing and display in the calling application.

. For this purpose, the option

- *"LINEBOUNDARY"*

has been implemented.



## Syntax

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<HelperCommand="FileUploadContent" RemoteFile="<RemoteFilePath>"
[LocalFile="<LocalFilePath>"] [ContentType="*XML|*HEX"] [Operation="*FIRST|
*NEXT|*LAST|*PREV" [FilePosition="<nnn>"] [Option="*LINEBOUNDARY"] />
```

## Description of the parameters

|                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>RemoteFile="&lt;RemoteFilePath&gt;"</i>    | Relative or absolute path name of the remote file to be transferred. If the path is specified relatively, the B-COMM data directory is used as root.                                                                                                                                                                                                                                                                                           |
| <i>[LocalFile="&lt;LocalFilePath&gt;"]</i>    | Optional. Relative or absolute path name of the file to be stored. If the path is specified relatively, the file is written into the "current" directory of the B-COMM Helper application. Same functionality as "FileUpload".                                                                                                                                                                                                                 |
| <i>[ContentType="*XML *HEX"]</i>              | Optional. Data type of the requested result data.<br>*XML = Data is delivered in XML format (currently only works for XML data with the tag "<Upload>").<br>*HEX = Data is delivered in HEX format (default setting).                                                                                                                                                                                                                          |
| <i>[Operation="*FIRST *NEXT *LAST *PREV"]</i> | Optional. File operation, if the file is to be read in blockwise.                                                                                                                                                                                                                                                                                                                                                                              |
| <i>[FilePosition="&lt;nnn&gt;"]</i>           | Optional: File position, for the next file access. Only relevant for *NEXT and *PREV. The numerical value can be taken from the response attribute <i>Resultposition="&lt;nnn&gt;"</i> of the previous call.                                                                                                                                                                                                                                   |
| <i>[Option="*LINEBOUNDARY"]</i>               | Optional. If specified, the read data is displayed within the line boundary.<br><br>Default setting: Independent of a line limit, the system reads with a defined block size. The block size is defined in the file:<br><ul style="list-style-type: none"> <li>- bcommserver.ini in the section</li> <li>- [Global] in the parameter</li> <li>- MaxFileChunkBufferSize=.</li> </ul> The default value is MaxFileChunkBufferSize=8192 (8 kbyte) |

**Response: Syntax**

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<HelperAnswer Command="FileUploadContent" RemoteFile="<RemoteFilePath>"
[LocalFile="<LocalFilePath>"] [ContentType="*XML|*HEX"] [Operation="*FIRST|
*NEXT|*LAST|*PREV" [FilePosition="<nnn>"] [Option="*LINEBOUNDARY"]
Status="ACK|NAK" [Resultposition="<nnn>"] [MoreData="<nnn>"]
[ChunkLen="<nnn>"]FileContent="<Data>" />
```

**Description of the parameters**

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Status="ACK/NAK"</i>               | Status information whether the command was completed successfully (ACK) or with error (NAK).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Status="ACK"</i>                   | Further attributes with the result data are only delivered if the command is successfully answered with ACK.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Status="NAK"</i>                   | No response data. The cause of the error can be checked in the B-COMM Helper log file (helper_YYMMDD.debug). See also paragraph "Error causes" at the end of this chapter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>[Resultposition="&lt;nnn&gt;"]</i> | Optional for blockwise reading.<br>Meaning:<br>File position after one of the accesses with <i>"*FIRST/*NEXT/*LAST/*PREV"</i> . This value must be passed on the next blockwise access with *NEXT or *PREV in the attribute <i>[FilePosition="&lt;nnn&gt;"]</i> .<br>Special case <i>*NEXT</i> :<br>If the <i>FilePos</i> attribute is not passed here, the start value is assumed to be "0" (begin of file).<br>As with <i>*FIRST</i> , it is read from the beginning of the file.<br>Special case <i>*PREV</i> :<br>If the <i>FilePos</i> attribute is not transferred here, the start value is assumed to be "-1" (end of file).<br>As with <i>*LAST</i> , it is read from the end of the file. |
| <i>[MoreData="&lt;nnn&gt;"]</i>       | Optional for blockwise reading.<br>This value specifies how many bytes of the file have not yet been read.<br>If the value <i>MoreData="0"</i> , no more data is returned with the next read access. This value should be evaluated by the application program ( $\leq 0$ ) to avoid further unnecessary read accesses.<br>If another read call (*NEXT or *PREV) is nevertheless made, the value <i>MoreData="-1"</i> is returned.                                                                                                                                                                                                                                                                 |
| <i>[ChunkLen="&lt;nnn&gt;"]</i>       | Optional for blockwise reading.<br>This value specifies how many bytes of the file were read during the current read access.<br>The value <i>ChunkLen="0"</i> or <i>ChunkLen="-1"</i> indicates that no data was returned.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>FileContent="&lt;Data&gt;"</i>     | With status "ACK" this value is always delivered. The read data is displayed here, regardless of whether an entire file or only a block (chunk) was read in.<br>With the values <i>ChunkLen="0"</i> or <i>ChunkLen="-1"</i> the result is empty: <i>FileContent=""</i>                                                                                                                                                                                                                                                                                                                                                                                                                             |

### Notes on data format

The format of the read data depends on the attribute "ContentType":

- *ContentType="\*XML"* = data is returned as XML tag <upload>
- *ContentType="\*HEX"* (or no specification of the "ContentType") = data is returned as hex values
- *ContentType="\*UTF16BE"* = data to be read is transferred in UTF\_16BE format (usable for files within the B-COMM installation)
- *ContentType="\*UTF8"* = data to be read is transferred in UTF\_8 format (usable for files outside the B-COMM installation)

The *ContentType="\*HEX"* always works.

Even in this case, the caller must know in which format the data is available in the source file (XML, ASCII, UTF8, UTF16, ...) to be able to interpret it correctly. All text-based files in the B-COMM directory are available in the format "UTF\_16BE" (also called "UCS-2 BE-BOM" depending on the program).

### Notes on the character set

It is the responsibility of the calling program to request and interpret the data in the correct character set.

For example, if data is requested with *ContentType="\*XML"*, the source file must also be an XML traceback file. If not, the response is "NAK".

#### 4.6.4.3 Error causes at NAK

If the status information "NAK" is returned, an error has occurred. This can have the following causes:

- Source file (RemoteFile) not found
- Target file (LocalFile) cannot be written
- invalid attribute combinations: The attribute *LocalFile="<LocalFilePath>"* must not be combined with *operation="\*FIRST|\*NEXT|\*LAST|\*PREV"*.

If a file is read in blocks, it cannot be saved: In this case the attribute "LocalFile" must not be specified.

- Exceptions during XML processing
- Exceptions during character set conversion
- Exceptions during HTTP or TCP/IP communication

### 4.6.5 FileDownload

Transmits the file <remoteFile> from the computer, on which B-COMM server is running, as <localFile> to the computer, on which the application is running. For <localFile> and <remoteFile>, the complete path must always be indicated.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="FileDownload"
LocalFile="..." RemoteFile="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**or**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

### 4.6.6 SendDataRecord

Sends the data record <dataRecord> to the stream <stream> of the client <client>. The data record is immediately sent to the corresponding terminal, independently if an order is carried out presently or not.

The response of the Helper is either **<ACK>** or **<NAK>**.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="SendDataRecord"
Client="..." Stream="..." DataRecord="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

**or**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK"> </HelperAnswer>
```

### 4.6.7 RenameCurrentBookingFile

This order gives the ability to rename the current booking file. The "Stream" parameter must be defined only if a separate booking file is created for every single stream, see chapter "Configuring the selected client" submenu "Advanced settings" in the B-COMM reference manual. The "File" parameter defines the new booking file name.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand
Command="RenameCurrentBookingFile" Client=".." Stream=".." File=" " />
```

**Response, if the order file could be renamed:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**Response, if the order file could not be renamed:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

**User example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand
Command="RenameCurrentBookingFile" Client="Kaba" Stream="BCLAN01"
File="Bookings_Monday"/>
```

In this example of use, the current booking file will be renamed as "Bookings\_Monday".

### 4.6.8 ReorganizeOrderFile

Reorganizes the order file of the stream <stream> of the client <client>. In this process, orders will be deleted which have a status defined in the <OrderStates> parameter. Several order states can be combined by the ; character. If in the GID parameter a value unlike "0" is defined, time scheduled orders will be deleted as well. The GID value is used as an indication flag only, not as a group address.

The variable DID can be used to specify the deleting tasks more precisely. To do so, a value is to be assigned to the variable.

This value defines, how old (days before) an order must be before deleting.

Example: DID=5. Deletes all orders older than 5 days.

Example: DID=0 (Default value). Deletes all orders from now on.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="ReorganizeOrderFile"
Client="..." Stream="..." GID="..." DID="..." OrderStates="..." />
```

**Response, if the order file could be reorganized:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="ACK" />
```

**Response, if the order file could not be reorganized:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

**User example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand Command="ReorganizeOrderFile"
Client="Kaba" Stream="BCLAN01" GID="1" DID="5" OrderStates="FinishedOK" />
```

In this example of use, all orders with the "FinishedOK" status will be deleted. The time scheduled orders will be deleted as well. This is valid for orders older than 5 days.

## 4.7 Configuration related Helper commands

### 4.7.1 ReadConfiguration

The complete configuration can be read out by means of this command only valid for XML.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?> <HelperCommand
Command="ReadConfiguration" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?> <HelperAnswer Status="NAK" />
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Client ClientName="..." StreamCount="5" ...>
    <Stream StreamName="..." Description="..." ...>
      <NetworkAdapter AdapterIPAddress="..."
AdapterType="..." ...>
        <Terminal Description="..." AlternativeID="..." GID="..."
...>
      </Terminal>
    </NetworkAdapter>
  </Stream>
</Client>
<Client ClientName="..." StreamCount="5" ...>
  <Stream StreamName="..." Description="..." ...>
    <NetworkAdapter AdapterIPAddress="..."
AdapterType="..." ...>
      <Terminal Description="..." AlternativeID="..." GID="..."
...>
    </Terminal>
  </NetworkAdapter>
</Stream>
</Client>
</HelperAnswer>
```

### 4.7.2 ReadGlobalConfigValue

Reads the value of the key <key> of the section <section> from the global configuration file bcommjava.ini.

The response of the Helper is either the value of the specified key in the specified section or <NAK> if section or key are not existing.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadGlobalConfigValue" Section="..." Key="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <GlobalConfigValue Key="Value" />
</HelperAnswer>
```

### 4.7.3 ReadNetworkAdapter

Reads the complete section of the network adapter with the IP address <IP address> of the stream <stream> of the client <client>. The response from the Helper is either a list of all entries of the section of the network adapter or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadNetworkAdapter" Client="..." Stream="..."
IPAddress="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <NetworkAdapter AdapterIPAddress="x.x.x.x"
AdapterType="..." />
</HelperAnswer>
```

### 4.7.4 ReadNetworkAdapters

Reads the IP addresses of all network adapters of the stream <stream> of the client <client>. The response from the Helper is either a list of all IP addresses, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadNetworkAdapters" Client="..."
Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <NetworkAdapter IPAddress="x.x.x.x" />
  <NetworkAdapter IPAddress="y.y.y.y" />
</HelperAnswer>
```

## 4.8 Client related Helper commands

### 4.8.1 ReadClient

Reads the complete section of the client <client> from the bcommjava.ini file. The response from the Helper is either a list of all entries in the section, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadClient" Client="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Client ClientName="..." StreamCount="5" StreamName1="..." ... />
</HelperAnswer>
```



### 4.8.2 ReadClients

Gives a list of all configured clients, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadClients"/>
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Client ClientName="..." />
  <Client ClientName="..." />
</HelperAnswer>
```

### 4.8.3 RefreshClients

Imports the internal list of the clients of the Helper again.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="RefreshClients" />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

## 4.9 Stream related Helper commands

### 4.9.1 ReadStream

Reads the complete section of the stream <stream> of the client <client>. The response from the Helper is either a list of all entries in the section, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadStream" Client="..." Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Stream StreamName="bclan04">
    <Section SectionName="[GLOBAL]" Description="..." ... />
    <Section SectionName="[HOST]" HostDNSName="..." ... />
    <Section SectionName="[TERMINAL_...]" Description="..." ... />
  </Stream>
</HelperAnswer>
```

### 4.9.2 ReadStreams

Reads the names of all configured streams of the client <client>. The response from the Helper is either a list of all names of the streams, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadStreams" Client="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Stream StreamName="bclan01">
  <Stream StreamName="bclan04">
</HelperAnswer>
```

### 4.9.3 ReadStreamStatus

Requests the status of the stream <stream> of the client <client>. The response of the Helper is either <ACK> or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadStreamStatus" Client="..."
Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
```

or

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

### 4.9.4 ReadStreamConfigValue

Reads the value of the key <key> of the section <section> from the stream configuration file of the stream <stream> of the client <client>.

The response of the Helper is either the value of the specified key in the specified section or <NAK> if section or key are not existing.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadStreamConfigValue" Client="..." Stream="..."
Section="..." Key="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <StreamConfigValue Key="Value" />
</HelperAnswer>
```

## 4.10 Terminal related Helper commands

### 4.10.1 ReadTerminal

Reads the complete section of the terminal with the address <address> of the stream <stream> of the client <client>. The variable <address> must have either the form GGDD or GG/DD (GG = two-digit GID, DD = two-digit DID).

The response from the Helper is either a list of all entries of the section of the terminal or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadTerminal" Client="..." Stream="..."
TerminalAddress="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Terminal Description="..." AlternativeID="..." GID="..." DID="..." ... />
</HelperAnswer>
```

### 4.10.2 ReadTerminals

Reads the alternative ID and GID/DID of all terminals of the stream <stream> of the client <client>. The response from the Helper is either a list with all terminals in the format <Alternative ID>/<GID>/<DID>, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadTerminals" Client="..." Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Terminal AlternativeID="..." GID="..." DID="..." />
  <Terminal AlternativeID="..." GID="..." DID="..." />
</HelperAnswer>
```

### 4.10.3 ReadTerminalSections

Reads the section names of all terminals of the stream <stream> of the client <client>. The response from the Helper is either a list of all sections, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadTerminalSections" Client="..."
Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Terminal SectionName="[TERMINAL_...]" AlternativeID="..." GID="..." />
  <Terminal SectionName="[TERMINAL_...]" AlternativeID="..." GID="..." />
</HelperAnswer>
```

## 4.11 Actuator related Helper commands

### 4.11.1 ReadActuatorConfigStatus

This command which is only valid for XML reads the status of one component or all components of an administration area of the stream <stream> of the client <client>. The response is the current status of the component or NAK.

**Example 1 (status of 1 component):**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorConfigStatus" Client="..." Stream="..."
AdministrationArea="..." LogicalName="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
<ActuatorConfigStatus LogicalName="..." Status="ActuatorNotFound"/> or
<ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="ProgrammerUpdateNecessary" FirmwareVersion="..." SerialNumber="..." />
or
<ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="ComponentUpdateNecessary" FirmwareVersion="..." SerialNumber="..." />
or
<ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="NormalOperation" FirmwareVersion="..." SerialNumber="..." />
</HelperAnswer>
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**Example 2 (status of all components):**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorConfigStatus" Client="..." Stream="..."
AdministrationArea="..." />
```

**Response for 3 components A, B, and C:**

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  Component A:
  <ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="ProgrammerUpdateNecessary" FirmwareVersion="..." SerialNumber="..." />
  Component B:
  <ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="ComponentUpdateNecessary" FirmwareVersion="..." SerialNumber="..." />
  Component C:
  <ActuatorConfigStatus LogicalName="..." DoorNumber="..."
Status="NormalOperation" FirmwareVersion="..." SerialNumber="..." />
</HelperAnswer>
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

### 4.11.2 ReadActuatorEventLog

This command which is only valid for XML reads the event memory of a component of the stream <stream> of the client <client>. The response is the content of the memory of the component or NAK.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorEventLog" Client="..." Stream="..."
AdministrationArea="KBS" LogicalName="CLEVER1"/>
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="ACK">
<ActuatorTraceback AdministrationArea="KBS" LogicalName="CLEVER1" Family="Elologic">
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:10:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:54:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:51:00.0000000+01:00" EventTypeId="7" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
</ActuatorTraceback>
</HelperAnswer>
```



### 4.11.3 ReadActuatorWhitelistEventLog

This command which is only valid for XML reads the event memory of a whitelist component of the stream <stream> of the client <client>. The response is the content of the memory of the whitelist component or NAK.

Also possible for wireless stream BCWLS02. If LogicalName is omitted, get event log of all actuators of the specified keyplan.

#### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorWhitelistEventLog" Client="..."
Stream="..." KeyPlan="KBS" LogicalName="CLEVER1"/>
```

#### Response:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="ACK">
<ActuatorTraceback Keyplan="KEYPLAN0" LogicalName="CLEVER1" Family="Elologic">
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:10:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:54:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:51:00.0000000+01:00" EventTypeId="7" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
</ActuatorTraceback>
</HelperAnswer>
```

#### 4.11.4 ReadActuatorCardLinkEventLog

This command which is only valid for XML reads the event memory of a component of the stream <stream> of the client <client>. The response is the content of the memory of the component or NAK.

Also possible for wireless stream BCWLS02. If LogicalName is omitted, get event log of all actuators of the specified administration area.

##### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorEventLog" Client="..." Stream="..."
AdministrationArea="KBS" LogicalName="CLEVER1"/>
```

##### Response:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="ACK">
<ActuatorTraceback AdministrationArea="KBS" LogicalName="CLEVER1" Family="Elologic">
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:32:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="5728962E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:33:00.0000000+01:00" EventTypeId="98" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-11-30T16:34:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:10:00.0000000+01:00" EventTypeId="97" Type="User"
UID="57267D2E" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:54:00.0000000+01:00" EventTypeId="6" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
<Entry DateTime="2006-12-01T09:51:00.0000000+01:00" EventTypeId="7" Type="MasterB"
UID="6E8F8B30" Technology="LegicPrime" />
</ActuatorTraceback>
</HelperAnswer>
```

### 4.11.5 ReadActuatorWhitelistConfigStatus

This command reads the configuration status of a Whitelist actuator.

#### Example 1: Standalone actuator at BCSCW01

```
<?xml version="1.0"?>
<HelperCommand Command="ReadActuatorWhitelistConfigStatus" Client="..."
Stream="BCSCW01" KeyPlan="..." LogicalName="..." />
```

|             |                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------|
| KeyPlan     | Number or name of the corresponding keyplan                                                     |
| LogicalName | Logical name of the requested actuator. If omitted, all actuators of the keyplan are requested. |

#### Response:

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="ReadActuatorWhitelistConfigStatus" Client="..."
Stream="BCSCW01" Status="ACK">
<ActuatorConfigStatus LogicalName="..." ConfigurationStatus="..."
WhitelistStatus="..."
SerialNumber="..." FirmwareVersion="..." BatteryLevel="..." />
</HelperAnswer>
```

|                     |                                                                      |
|---------------------|----------------------------------------------------------------------|
| ConfigurationStatus | Configuration status of the actuator like shown in the B-COMM GUI    |
| WhitelistStatus     | Status of the whitelist of the actuator like shown in the B-COMM GUI |
| SerialNumber        | Serial number of the actuator                                        |
| FirmwareVersion     | Firmware version of the actuator                                     |
| BatteryLevel        | Battery level in %                                                   |

#### Example 2: Wireless actuator at BCWLS02

```
<?xml version="1.0"?>
<HelperCommand Command="ReadActuatorWhitelistConfigStatus" Client="..."
Stream="BCWLS02" KeyPlan="..." LogicalName="..." />
```

|             |                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------|
| KeyPlan     | Number or name of the corresponding keyplan                                                     |
| LogicalName | Logical name of the requested actuator. If omitted, all actuators of the keyplan are requested. |

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="ReadActuatorWhitelistConfigStatus" Client="..."
Stream="BCWLS02" Status="ACK">
<ActuatorConfigStatus LogicalName="..." ConfigurationStatus="..."
WhitelistStatus="..." SerialNumber="..."
FirmwareVersion="..." HardwareVersion="..." ZCLVersion="..." ZigBeeVersion="..."
NetworkAddress="..."
LegicChipsetInfo="..." BatteryLevel="..." LinkQuality="..." LockState="..."
LockMode="..." />
</HelperAnswer>
```

|                     |                                                                      |
|---------------------|----------------------------------------------------------------------|
| ConfigurationStatus | Configuration status of the actuator like shown in the B-COMM GUI    |
| WhitelistStatus     | Status of the whitelist of the actuator like shown in the B-COMM GUI |
| SerialNumber        | Serial number of the actuator                                        |
| FirmwareVersion     | Firmware version of the actuator                                     |
| HardwareVersion     | Hardware version of the actuator                                     |
| ZCLVersion          | ZCL (ZigBee Cluster Library) version of the actuator                 |
| ZigBeeVersion       | ZigBee version of the actuator                                       |
| NetworkAddress      | ZigBee network address of the actuator                               |
| LegicChipsetInfo    | Some information about the LEGIC chipset of the actuator             |
| BatteryLevel        | Battery level in %                                                   |
| LinkQuality         | Link quality between the actuator and its gateway in dBm             |
| LockState           | The lock state of the actuator                                       |
|                     | -1 = unknown (not known in B-COMM)                                   |
|                     | 0 = undefined (error state)                                          |
|                     | 1 = locked                                                           |
|                     | 2 = unlocked                                                         |
| LockMode            | The lock mode of the actuator                                        |
|                     | -1 = unknown (not known in B-COMM)                                   |
|                     | 0 = normal operation                                                 |
|                     | 1 = unlock                                                           |
|                     | 2 = access control                                                   |
|                     | 3 = shutdown                                                         |

### 4.11.6 ReadActuatorCardLinkConfigStatus

This command reads the configuration status of a CardLink actuator.

#### Example 1: Standalone actuator at BCSCW01

```
<?xml version="1.0"?>
<HelperCommand Command="ReadActuatorCardLinkConfigStatus" Client="..."
Stream="BCSCW01" AdministrationArea="..." LogicalName="..." />
```

|                    |                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------|
| AdministrationArea | Number or name of the corresponding administration area                                         |
| LogicalName        | Logical name of the requested actuator. If omitted, all actuators of the keyplan are requested. |

#### Response:

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="ReadActuatorCardLinkConfigStatus" Client="..."
Stream="BCSCW01" Status="ACK">
<ActuatorConfigStatus LogicalName="..." ConfigurationStatus="..."
WhitelistStatus="..."
SerialNumber="..." FirmwareVersion="..." BatteryLevel="..." />
</HelperAnswer>
```

|                     |                                                                      |
|---------------------|----------------------------------------------------------------------|
| ConfigurationStatus | Configuration status of the actuator like shown in the B-COMM GUI    |
| WhitelistStatus     | Status of the whitelist of the actuator like shown in the B-COMM GUI |
| SerialNumber        | Serial number of the actuator                                        |
| FirmwareVersion     | Firmware version of the actuator                                     |
| BatteryLevel        | Battery level in %                                                   |

#### Example 2: Wireless actuator at BCWLS02

```
<?xml version="1.0"?>
<HelperCommand Command="ReadActuatorCardLinkConfigStatus" Client="..."
Stream="BCWLS02" AdministrationArea="..." LogicalName="..." />
```

|                    |                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------|
| AdministrationArea | Number or name of the corresponding administration area                                         |
| LogicalName        | Logical name of the requested actuator. If omitted, all actuators of the keyplan are requested. |

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="ReadActuatorCardLinkConfigStatus" Client="..."
Stream="BCWLS02" Status="ACK">
<ActuatorConfigStatus LogicalName="..." ConfigurationStatus="..."
WhitelistStatus="..." SerialNumber="..."
FirmwareVersion="..." HardwareVersion="..." ZCLVersion="..." ZigBeeVersion="..."
NetworkAddress="..."
LegicChipsetInfo="..." BatteryLevel="..." LinkQuality="..." LockState="..."
LockMode="..." />
</HelperAnswer>
```

|                     |                                                                      |
|---------------------|----------------------------------------------------------------------|
| ConfigurationStatus | Configuration status of the actuator like shown in the B-COMM GUI    |
| WhitelistStatus     | Status of the whitelist of the actuator like shown in the B-COMM GUI |
| SerialNumber        | Serial number of the actuator                                        |
| FirmwareVersion     | Firmware version of the actuator                                     |
| HardwareVersion     | Hardware version of the actuator                                     |
| ZCLVersion          | ZCL (ZigBee Cluster Library) version of the actuator                 |
| ZigBeeVersion       | ZigBee version of the actuator                                       |
| NetworkAddress      | ZigBee network address of the actuator                               |
| LegicChipsetInfo    | Some information about the LEGIC chipset of the actuator             |
| BatteryLevel        | Battery level in %                                                   |
| LinkQuality         | Link quality between the actuator and its gateway in dBm             |
| LockState           | The lock state of the actuator                                       |
|                     | -1 = unknown (not known in B-COMM)                                   |
|                     | 0 = undefined (error state)                                          |
|                     | 1 = locked                                                           |
|                     | 2 = unlocked                                                         |
| LockMode            | The lock mode of the actuator                                        |
|                     | -1 = unknown (not known in B-COMM)                                   |
|                     | 0 = normal operation                                                 |
|                     | 1 = unlock                                                           |
|                     | 2 = access control                                                   |
|                     | 3 = shutdown                                                         |

## 4.12 Order related Helper commands

### 4.12.1 ReadOrders

Lists all sections of the order file of the stream <stream> of the client <client>. The response from the Helper is a list of the sections of the corresponding order file.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadOrders" Client="..." Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order SectionName="ORDER_....." />
  <Order SectionName="ORDER_....." />
</HelperAnswer>
```

### 4.12.2 ReadOrdersExecuted

This command which is only valid for XML allows to verify whether all orders of the stream <stream> of the client <client> are executed not.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadOrdersExecuted" Client="..." Stream="..." />
```

**Response, if all orders are executed:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Orders AllExecuted="true" />
</HelperAnswer>
```

**Response, if not all orders are executed:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Orders AllExecuted="false" />
</HelperAnswer>
```

### 4.12.3 ReadOrderStatus

Reads the execution status of the order with the section name (= order name) <section> of the stream <stream> of the client <client>. The response of the Helper is either <execution status> of the order or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadOrderStatus" Client="..." Stream="..."
Section="ORDER..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order Status="FinishedOK" />
</HelperAnswer>
```

### 4.12.4 ReadRecentOrderStatus

This command which is only valid for XML reads the execution status of the last order sent of the stream <stream> of the client <client>. The response of the Helper is either <execution status> of the order or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadRecentOrderStatus" Client="..." Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order Status="Running" />
</HelperAnswer>
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order Status="FinishedError" Error="error text"/>
</HelperAnswer>
```



### 4.12.5 WriteOrder

The order file <localfile> on the computer, on which the application is running, is added together with its complete path to the orders in the existing order file of the stream <stream> of the client <client>.

The structure of the local file must be identical to the structure of the B-COMM order file.

The response from the Helper is either a list with the section names of the orders, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteOrder" Client="..." Stream="..." LocalFile="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order SectionName="ORDER_....." />
</HelperAnswer>
```

### 4.12.6 WriteOrderOverwrite

The order file <localfile> on the computer, on which the application is running, together with its complete path, overwrites the orders in the existing order file of the stream <stream> of the client <client>.

The structure of the local file must be identical to the structure of the B-COMM order file.

The response from the Helper is either a list with the section names of the orders, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteOrderOverwrite" Client="..." Stream="..."
LocalFile="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order SectionName="ORDER_....." />
</HelperAnswer>
```



The Helper commands

- FileUpload
- FileDownload
- WriteOrder
- WriteOrderOverwrite

work only if <localfile> is on the same computer as B-COMM Helper.

---

### 4.12.7 WriteOrderDirect

Adds the orders whose parameters are defined in the XML structure to the order file of the stream <stream> of the client <client>. Only the parameters relevant for the respective order must be defined in the XML structure. These parameters are shown in the chapter "List of general orders".

The response from the Helper is either a list with the section names of the orders, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteOrderDirect"
Client="<Clientname>" Stream="<Streamname>"
<OrderCommand="<DisplayMessage>"
Subcommand="<D1>"
GID="<01>"
DID="<02>"
Data="<Good Morning>"
</HelperCommand>
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order SectionName="ORDER_....." />
</HelperAnswer>
```

### 4.12.8 WriteOrderOverwriteDirect

Overwrites the order file of the stream <stream> of the client <client> with the orders defined in the XML structure. Only the parameters relevant for the respective order must be defined in the XML structure. These parameters are shown in the chapter "List of general orders".

The response from the Helper is either a list with the section names of the orders, separated by CR/LF, or <NAK>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteOrderOverwriteDirect"
Client="<Clientname>" Stream="<Streamname>"
<OrderCommand="<DisplayMessage>"
Subcommand="<D1>"
GID="<01>"
DID="<02>"
Data="<Good Morning>"
</HelperCommand>
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

**or**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Order SectionName="ORDER_....." />
</HelperAnswer>
```

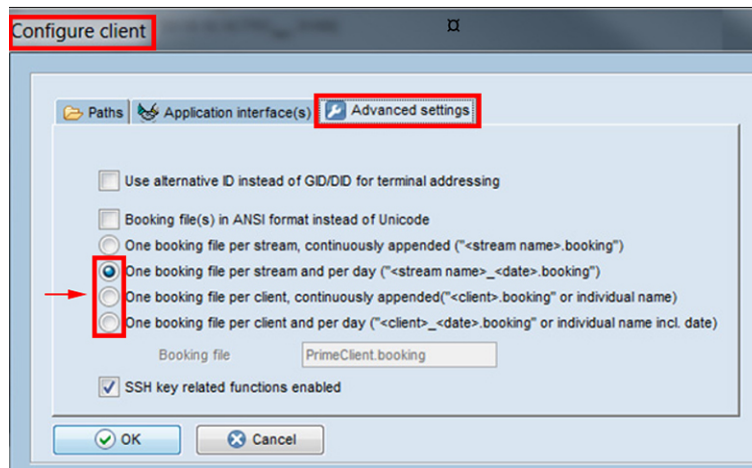
## 4.13 Booking related Helper commands

### 4.13.1 ReadBookings

Reads all bookings of the stream <stream> of the client <client> that were carried out in the period between <startdate> and <enddate>. The response of the Helper is either a list of all bookings or <NAK>.

**Precondition:**

Select one of the marked booking file options in the "Configure Client" menu in the "Advanced settings" tab. For detailed information please check the B-COMM manual.



**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadBookings" Client="..." Stream="..."
StartDate="YYMMDD" EndDate="YYMMDD"/>
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

or

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <Booking Data="..." />
  <Booking Data="..." />
  <Booking Data="..." />
</HelperAnswer>
```

## 4.14 CardLink related Helper commands

### 4.14.1 ReadCardLinkConfiguration

This command which is only valid for XML returns the CardLink configuration of the stream <stream> of the client <client>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadCardLinkConfiguration" Client="..." Stream="..." />
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="ACK">
  <CardLinkConfiguration Client="..." Stream="...">
    <AdministrationArea Index="...">
      ....
    </AdministrationArea>
  </CardLinkConfiguration>
</HelperAnswer>
```

### 4.14.2 WriteCardLinkActuatorConfiguration

Writes a CardLink component configuration to the <stream> of the <client> and sets the defined parameters.

Any possibly existing configuration will be overwritten.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkActuatorConfiguration"
  Client="..." Stream="...">
  <CardLinkConfiguration Client="..." Stream="...">
    <Actuator Changed="..." ...>
      ...
    </Actuator>
  </CardLinkConfiguration>
</HelperCommand>
```

### 4.14.3 UpdateCardLinkActuatorConfiguration

Updates existing CardLink actuators under an administration area or under a gateway.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<LegicAdvantStampConfiguration />` will delete all existing LegicAdvantStampConfiguration elements. That can be used to first delete all existing elements and then add some new elements. For elements, that can appear multiple times (like e.g. SpecialDay1), the new element will be added if it not already exists. The criteria is **Date** for SpecialDays, **StartDate** for Holidays, **Index** for DoorGroup, **Stamp** for LegicAdvantStampConfiguration and **CardId** and **AppDefId** for Blacklist entries. The TimePro element must be completely specified to replace the existing one.

**Example1: Update an existing CardLink actuator under an administration area.**

```
<?xml version="1.0"?>
<HelperCommand
Command="UpdateCardLinkActuatorConfiguration"
Client="..." Stream="BCSCW01" >
<CardLinkConfiguration Client="..." Stream="BCSCW01" >
  <Actuator LogicalName="..." >
    [<Gateway Index="1" Name="..." />]
    [<CabinetLock ... />]
    <OiFParameter ... />
    <EnergyOptions ... />
    <DoorSensors ... />
    <Master ... />
    <DoorGroup Index="..." ... />
    ...
    <LegicAdvantStampConfiguration Stamp="..." ...
    ...
    <SpecialDay1 Date="..." />
    ...
    <SpecialDay2 Date="..." />
    ...
    <Holidays StartDate="..." EndDate="..." />
    ...
    <TimePro ...>
      <TimeWindow ... />
      ...
    </TimePro>
    <Blacklist>
      <Entry CardId="..." AppDefId="..." ... />
      ...
    </Blacklist>
  </Actuator>
  ...
</CardLinkConfiguration >
</HelperCommand>
```

**Example 2: Update an existing CardLink actuator under a gateway.**

```

<?xml version="1.0"?>
<HelperCommand
Command="UpdateCardLinkActuatorConfiguration"
Client="..." Stream="BCWLS02" >
<WirelessConfiguration Client="..." Stream="BCWLS02" >
  <Actuator LogicalName="..." ...>
    [<Gateway Index="1" Name="..." />]
    [<CabinetLock ... />]
    <OiFParameter ... />
    <EnergyOptions ... />
    <DoorSensors ... />
    <Master ... />
    <DoorGroup Index="..." ... />
    ...
    <LegicAdvantStampConfiguration Stamp="..." ...
    ...
    <SpecialDay1 Date="..." />
    ...
    <SpecialDay2 Date="..." />
    ...
    <Holidays StartDate="..." EndDate="..." />
    ...
    <TimePro ...>
      <TimeWindow ... />
      ...
    </TimePro>
    <Blacklist>
      <Entry CardId="..." AppDefId="..." ... />
      ...
    </Blacklist>
  </Actuator>
  ...
</WirelessConfiguration >
</HelperCommand>

```

**Response:**

```

<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="UpdateCardLinkActuatorConfiguration" Client="..."
Stream="BCSCW01 | BCWLS02" Status="ACK" />

```



#### 4.14.4 WriteCardLinkAdministrationAreaConfiguration

Writes a CardLink administration area configuration to the <stream> of the <client> and sets the defined parameters.

Any possibly existing configuration will be overwritten.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkAdministrationAreaConfiguration" Client="..." Stream="...">
  <CardLinkConfiguration Client="..." Stream="...">
    <AdministrationArea Changed="..." Index="...">
      ...
    </AdministrationArea>
  </CardLinkConfiguration>
</HelperCommand>
```

#### 4.14.5 UpdateCardLinkAdministrationAreaConfiguration

Update an existing CardLink administration area.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<LegicAdvantStampConfiguration />` will delete all existing LegicAdvantStampConfiguration elements. That can be used to first delete all existing elements and then add some new elements. For elements, that can appear multiple times (like e.g. SpecialDay1), the new element will be added if it not already exists. The criteria is **Date** for SpecialDays, **StartDate** for Holidays, **Stamp** for LegicAdvantStampConfiguration and **CardId** and **AppDefId** for Blacklist entries. The TimeProfile/TimePro element(s) must be completely specified to replace the existing one.

**Example:**

```
<?xml version="1.0" ?>
<HelperCommand
Command="UpdateCardLinkAdministrationAreaConfigur
ation" Client="..." Stream="BCSCW01" >
<CardLinkConfiguration Client="..." Stream="BCSCW01" >
  <AdministrationArea Index="..." ...>
    <OiFParameter ... />
    <ValidationDuration .../>
    ...
    <Master ... />
    ...
    <LegicAdvantStampConfiguration Stamp="..." ...
    ...
    <SpecialDay1 Date="..." />
    ...
    <SpecialDay2 Date="..." />
    ...
    <Holidays StartDate="..." EndDate="..." />
    ...
    <TimePro ...>
      <TimeWindow ... />
      ...
    </TimePro ...>
    ...
    <TimeProfile ...>
      <TimeWindow ... />
      ...
    </TimeProfile ...>
    ...
    <Actuator LogicalName="..." ...>
      <Gateway Index="1" Name="..." />
      <OiFParameter ... />
      <EnergyOptions ... />
      <DoorSensors ... />
```

```

<Master ... />
<DoorGroup>
...
<LegicAdvantStampConfiguration Stamp="..." ... />
...
<SpecialDay1 Date="..." />
...
<SpecialDay2 Date="..." />
...
<Holidays StartDate="..." EndDate="..." />
...
<TimePro>
  <TimeWindow ... />
  ...
</TimePro>
<Blacklist>
  <Entry CardId="..." AppDefId="..." ... />
  ...
</Blacklist>
</Actuator>
...
</AdministrationArea>
...
</CardLinkConfiguration>
</HelperCommand>

```

**Response:**

```

<?xml version="1.0" encoding="iso-8859-1"?>

<HelperAnswer Command="UpdateCardLinkAdministrationAreaConfiguration" Client="..."
Stream="BCSCW01 | BCWLS02" Status="ACK" />

```

#### 4.14.6 WriteCardLinkConfiguration



##### NOTICE

When using the reader technologies LEGIC advant or Mifare, a maximum of 512 components per management area can be loaded to the programmer PD1460.

---

This command which is only valid for XML writes a CardLink configuration in the stream <stream> of the client <client> and sets the defined parameters.

Any possibly existing configuration is overwritten.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkConfiguration" Client="..."
  Stream="...">
  <CardLinkConfiguration Client="..." Stream="...">
    <AdministrationArea Index="...">
      ...
    </AdministrationArea>
  </CardLinkConfiguration>
</HelperCommand>
```

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Status="NAK">
```

## 4.15 Whitelist related Helper commands

### 4.15.1 WriteWhitelistPool

Writes a list of whitelist media for a client <client>.

**Example:**

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperCommand Command="WriteWhitelistPool" Client="...">
<WirelessWhitelistPool Client="...">
<WhitelistLegicAdvant>
  <Entry AppDefId="..." CardId="..." Description="..."
    OfficeIndividual="..." SafeUID="..." TimeProfileIndex="..." />
  ...more LEGIC advant media...
</WhitelistLegicAdvant>
<WhitelistMifare>
  <Entry AppDefId="..." CardId="..." Description="..." OfficeIndividual="..."
    SafeUID="..." TimeProfileIndex="..." />
  ...more Mifare media...
</WhitelistMifare>
</WirelessWhitelistPool>
</HelperCommand>
```

**Explanation of the parameters**

|                  |                   |                                                             |
|------------------|-------------------|-------------------------------------------------------------|
| AppDefId         | 0...7             | Index of the LEGIC advant stamp or the sitekey Index => CID |
|                  | 8                 | UID                                                         |
|                  | 14                | Don't care                                                  |
|                  | 15                | RCID UID (TouchGo Components only)                          |
| CardId           |                   | CID/UID of the media                                        |
| Description      | n                 | B-COMM internal description                                 |
| OfficeIndividual | "true" or "false" | Defines, whether "OfficeIndividual" is to be used.          |
| SafeUID          | "true" or "false" | Defines, whether "SafeUID" is to be used.                   |
| TimeProfileIndex |                   | Index of the belonging time profile.                        |

### 4.15.2 WriteWhitelistConfiguration

Writes a whitelist configuration for a client <client> and an (optional) stream <stream>.

**a) Example for a complete configuration for both streams BCSCW01 and BCWLS02**

```
<?xml version="1.0" encoding="utf-8"
<HelperCommand Command="WriteWhitelistConfiguration" Client="...">
<WhitelistConfiguration Client="..." Stream="BCSCW01">
  <KeyPlan Index="..." ...>
    ...
    <Actuator ...>
      ...
    </Actuator>
    ...
  </KeyPlan>
</WhitelistConfiguration>
<WirelessConfiguration Client="..." Stream="BCWLS02" Mode="Whitelist">
  <Gateway Index="..." ...>
    <Actuator ...>
      ...
    </Actuator>
    ...
  </Gateway>
  ...
</WirelessConfiguration>
</HelperCommand>
```

**b) Example for a configuration for stream BCSCW01**

```
<?xml version="1.0" encoding="utf-8"
<HelperCommand Command="WriteWhitelistConfiguration" Client="..."
Stream="BCSCW01">
<WhitelistConfiguration Client="..." Stream="BCSCW01">
  <KeyPlan Index="..." ...>
  ...
  <Actuator ...>
  ...
  </Actuator>
  ...
</KeyPlan>
</WhitelistConfiguration>
</HelperCommand>
```

**c) Example for a configuration for stream BCWLS02**

```
<?xml version="1.0" encoding="utf-8"
<HelperCommand Command="WriteWhitelistConfiguration" Client="..."
Stream="BCWLS02">
<WirelessConfiguration Client="..." Stream="BCWLS02">
  <Gateway Index="..." ...>
    <Actuator ...>
    ...
    </Actuator>
    ...
  </Gateway>
  ...
</WirelessConfiguration>
</HelperCommand>
```

### 4.15.3 WriteWhitelistActuatorConfiguration

Writes a configuration for one or more actuators for a stream <stream> of a client <client>

**Example:**

```
<?xml version="1.0" encoding="utf-8"
<HelperCommand Command="WriteWhitelistActuatorConfiguration" Client="..."
Stream="...">
<WhitelistConfiguration Client="..." Stream="...">
  <Actuator ...>
    </Actuator>
  </WhitelistConfiguration>
</HelperCommand>
```



#### 4.15.4 UpdateWhitelistActuatorConfiguration

Updates existing whitelist actuators under a keyplan or under a gateway.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<LegicAdvantStampConfiguration />` will delete all existing LegicAdvantStampConfiguration elements. That can be used to first delete all existing elements and then add some new elements. For elements, that can appear multiple times (like e.g. SpecialDay1), the new element will be added if it not already exists. The criteria is Date for SpecialDays, StartDate for Holidays, Stamp for LegicAdvantStampConfiguration and CardId and AppDefId for Whitelist entries. The TimePro element must be completely specified to replace the existing one.

**Example 1: Update an existing Whitelist actuator under a keyplan.**

```
<?xml version="1.0"?>
<HelperCommand
Command="UpdateWhitelistActuatorConfiguration"
Client="..." Stream="BCSCW01" >
<WhitelistConfiguration Client="..." Stream="BCSCW01" >
  <Actuator LogicalName="..." >
    [<Gateway Index="1" Name="..." />]
    <OiFParameter ... />
    <EnergyOptions ... />
    <DoorSensors ... />
    <ProgrammingMaster ... />
    <LegicAdvantStampConfiguration Stamp="..." ...
    ...
    <SpecialDay1 Date="..." />
    ...
    <SpecialDay2 Date="..." />
    ...
    <Holidays StartDate="..." EndDate="..." />
    ...
    <TimePro ...>
      <TimeWindow ... />
      ...
    </TimePro>
    <Whitelist>
      <Entry CardId="..." AppDefId="..." ... />
      ...
    </Whitelist>
  </Actuator>
  ...
</WhitelistConfiguration >
</HelperCommand>
```

**Example 2: Update an existing Whitelist actuator under a gateway.**

```

<?xml version="1.0"?>
<HelperCommand
Command="UpdateWhitelistActuatorConfiguration"
Client="..." Stream="BCWLS02" >
<WirelessConfiguration Client="..." Stream="BCWLS02" >
  <Actuator LogicalName="..." ...>
    [<Gateway Index="1" Name="..." />]
    <OiFParameter ... />
    <EnergyOptions ... />
    <DoorSensors ... />
    <ProgrammingMaster ... />
    <LegicAdvantStampConfiguration Stamp="..." ...
    ...
    <SpecialDay1 Date="..." />
    ...
    <SpecialDay2 Date="..." />
    ...
    <Holidays StartDate="..." EndDate="..." />
    ...
    <TimePro ...>
      <TimeWindow ... />
      ...
    </TimePro>
    <Whitelist>
      <Entry CardId="..." AppDefId="..." ... />
      ...
    </Whitelist>
  </Actuator>
  ...
</WirelessConfiguration >
</HelperCommand>

```

**Response:**

```

<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="UpdateWhitelistActuatorConfiguration" Client="..."
Stream="BCSCW01 | BCWLS02" Status="ACK" />

```

### 4.15.5 WriteWhitelistKeyPlanConfiguration

Writes a configuration for one or more key plans for a stream <stream> of a client <client>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"
<HelperCommand Command="WriteWhitelistKeyPlanConfiguration" Client="..."
Stream="...">
<WhitelistConfiguration Client="..." Stream="...">
  <KeyPlan ...>
    <Actuator ...>
      ...
    </Actuator>
    ...
  </KeyPlan>
  ...
</WhitelistConfiguration>
</HelperCommand>
```

### 4.15.6 UpdateWhitelistKeyPlanConfiguration

Update an existing Whitelist keyplan.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<LegicAdvantStampConfiguration />` will delete all existing LegicAdvantStampConfiguration elements. That can be used to first delete all existing elements and then add some new elements. For elements, that can appear multiple times (like e.g. SpecialDay1), the new element will be added if it not already exists. The criteria is **Date** for SpecialDays, **StartDate** for Holidays, **Stamp** for LegicAdvantStampConfiguration and **CardId** and **AppDefId** for Whitelist entries. The TimeProfile/TimePro element(s) must be completely specified to replace the existing one.

**Example:**

```
<?xml version="1.0" ?>

<HelperCommand
Command="UpdateWhitelistKeyPlanConfiguration"
Client="..." Stream="BCSCW01" >

<WhitelistConfiguration Client="..." Stream="BCSCW01" >

  <KeyPlan Index="..." />
  <OiFParameter ... />
  <Master ... />
  ...
  <LegicAdvantStampConfiguration Stamp="..." />
  ...
  <SpecialDay1 Date="..." />
  ...
  <SpecialDay2 Date="..." />
  ...
  <Holidays StartDate="..." EndDate="..." />
  ...
  <TimePro ...>
    <TimeWindow ... />
    ...
  </TimePro ...>
  ...
  <TimeProfile ...>
    <TimeWindow ... />
    ...
  </TimeProfile ...>
  ...
  <Actuator LogicalName="...">
    <Gateway Index="1" Name="..." />
    <OiFParameter ... />
    <EnergyOptions ... />
    <DoorSensors ... />
    <ProgrammingMaster ... />
    <LegicAdvantStampConfiguration Stamp="..." />
```

```

...
<SpecialDay1 Date="..." />
...
<SpecialDay2 Date="..." />
...
<Holidays StartDate="..." EndDate="..." />
...
<TimePro>
  <TimeWindow ... />
  ...
</TimePro>
<Whitelist>
  <Entry CardId="..." AppDefId="..." ... />
  ...
</Whitelist>
</Actuator>
...
</KeyPlan>
...
</WhitelistConfiguration>
</HelperCommand>

```

**Response:**

```

<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="UpdateWhitelistKeyPlanConfiguration" Client="..."
Stream="BCSCW01 | BCWLS02" Status="ACK" />

```

### 4.15.7 WriteWhitelistActuatorWhitelist

Writes a list of whitelist media for a stream <stream> of a client <client>.

#### a) Example for an actuator connected to stream BCSCW01

```
<xml version="1.0" encoding="iso-8859-1">
<HelperCommand Command="WriteWhitelistActuatorWhitelist" Client="..."
Stream="BCSCW01">
<ActuatorWhitelist Client="...">
  <Actuator LogicalName="..." KeyplanIndex="...">
    <Whitelist>
      <Entry AppDefId="..." CardId="..." Description="..."
        OfficeIndividual="..." SafeUID="..." TimeProfileIndex="..." />
      ...
    </Whitelist>
  </Actuator>
</ActuatorWhitelist>
</HelperCommand>
```

#### b) Example for an actuator connected to stream BCWLS02

```
<xml version="1.0" encoding="iso-8859-1">
<HelperCommand Command="WriteWhitelistActuatorWhitelist" Client="..."
Stream="BCWLS02">
<ActuatorWhitelist Client="...">
  <Actuator LogicalName="..." GatewayIndex="...">
    <Whitelist>
      <Entry AppDefId="..." CardId="..." Description="..."
        OfficeIndividual="..." SafeUID="..." TimeProfileIndex="..." />
      ...
    </Whitelist>
  </Actuator>
</ActuatorWhitelist>
</HelperCommand>
```

### 4.15.8 AppendWhitelistActuatorWhitelist

Adds one or several whitelist entries to a list of whitelist media in a <Actuator LogicalName="..." GatewayIndex="..."> component of a <stream> stream of a <client> client. Entries available in the component are not overwritten, but are supplemented.

If several entries are supplemented, a separate WhitelistDownloadWirelessActuatorGateway order with the "SingleEntry" subcommand for the transmission of the entry to the component is created for each individual entry, see chapter "Whitelist via gateway order: WhitelistDownloadWirelessActuatorGateway".

#### Example for two actuators on the BCWLS02 stream

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="AppendWhitelistActuatorWhitelist" Client="Whitelist"
Stream="BCWLS02">
  <ActuatorWhitelist Client="Whitelist">
    <Actuator LogicalName="ADVANT-WIRELESS-CLEVER-1" GatewayIndex="1">
      <Whitelist>
        <Entry Technology="LegicAdvant" AppDefId="2" CardId="2041" Description="Meiner"
IdentificationStandard="Identification" TimeProfileIndex="0" OfficeIndividual="true"
SafeUID="false" />
        <Entry Technology="LegicAdvant" AppDefId="2" CardId="8193" Description="Schulung"
IdentificationStandard="Identification" TimeProfileIndex="0" OfficeIndividual="false"
SafeUID="false"/>
      </Whitelist>
    </Actuator>
    <Actuator LogicalName="ADVANT-WIRELESS-CLEVER-2" GatewayIndex="1">
      <Whitelist>
        <Entry Technology="LegicAdvant" AppDefId="2" CardId="2041" Description="Meiner"
IdentificationStandard="Identification" TimeProfileIndex="0" OfficeIndividual="true"
SafeUID="false"/>
        <Entry Technology="LegicAdvant" AppDefId="2" CardId="8193" Description="Schulung"
IdentificationStandard="Identification" TimeProfileIndex="0" OfficeIndividual="false"
SafeUID="false"/>
      </Whitelist>
    </Actuator>
  </ActuatorWhitelist>
</HelperCommand>
```

### 4.15.9 DeleteWhitelistActuatorWhitelist

Deletes one or several whitelist entries from a list of whitelist media in a <Actuator LogicalName="..." GatewayIndex="..."> component of a <stream> stream of a <client> client. There is no need to transmit the complete whitelist.

If several entries are deleted, a separate WhitelistDeleteWirelessActuatorGateway order with the "SingleEntry" subcommand for the transmission of the entry to the component is created for each individual entry, see chapter "Whitelist order via gateway: WhitelistDeleteWirelessActuatorGateway".

#### Example for two actuators on the BCWLS02 stream

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="DeleteWhitelistActuatorWhitelist" Client="Whitelist"
Stream="BCWLS02">
  <ActuatorWhitelist Client="Whitelist">
    <Actuator LogicalName="ADVANT-WIRELESS-CLEVER-1" GatewayIndex="1">
      <Whitelist>
        <Entry AppDefId="2" CardId="8193"
        <Entry AppDefId="2" CardId="2041"
      </Whitelist>
    </Actuator>
    <Actuator LogicalName="ADVANT-WIRELESS-CLEVER-2" GatewayIndex="1">
      <Whitelist>
        <Entry AppDefId="2" CardId="8193"
        <Entry AppDefId="2" CardId="2041"
      </Whitelist>
    </Actuator>
  </ActuatorWhitelist>
</HelperCommand>
```



#### 4.15.10 WriteWhitelistGatewayConfiguration

Writes a configuration for one or more gateways for a stream <stream> of a client <client>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="WriteWhitelistGatewayConfiguration" Client="..."
Stream="BCWLS02">
  <WirelessConfiguration Client="..." Stream="BCWLS02">
    <Gateway Name="..." Index="...">
      <NetworkSettings ... />
      <WebServer ... />
      <User ... />
      <Actuator ...>
        ...
      </Actuator>
      ...
    </Gateway>
    ...
  </WirelessConfiguration>
</HelperCommand>
```

#### 4.15.11 UpdateWhitelistGatewayConfiguration

Update an existing Whitelist gateway.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<Actuator />` will delete all existing Actuator elements. That can be used to first delete all existing elements and then add some new elements.

**Example:**

```
<?xml version="1.0" ?>
<HelperCommand
  Command="UpdateWhitelistGatewayConfiguration"
  Client="..." Stream="BCWLS02" >
  <WirelessConfiguration Client="..." Stream="BCWLS02" >
    <Gateway Index="..." Name="..." ...>
      <NetworkSettings ... />
      <Actuator LogicalName="..." GUID="..." ... >
        ...
      </Actuator>
      ...
    </Gateway>
    ...
  </WirelessConfiguration>
</HelperCommand>
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="UpdateWhitelistGatewayConfiguration" Client="..."
  Stream="BCWLS02" Status="ACK" />
```

#### 4.15.12 ReadKnownGateways

See description in chapter ReadKnownGateways [▶ 4.16.4](#)].

#### 4.15.13 ReadGateways

See description in chapter ReadGateways [▶ 4.16.5](#)].

#### 4.15.14 ReadActuatorJoinStates

See description in chapter ReadActuatorJoinStates [▶ 4.16.6](#)].

### 4.15.15 DoorControl

Transmits the door control commands to the wireless components of the client <client>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="DoorControl" Client="..." Stream="..."
GatewayIPAddress="..." LogicalName="..." DID="..." GID="..." />
```

**Explanation of the parameters**

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Client="..."           | Name of the client                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Stream="..."           | Name of the stream, e.g. BCWLS02.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| GatewayIPAddress="..." | IP address of the gateway the component is joined to.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| LogicalName="..."      | Logical name of the component or * for all components.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| DID="..."              | <p>Numeric value (0 to 8), that determines the command to be send.</p> <p>These are:</p> <p>0 = Release permanently</p> <p>1 = Release time limited</p> <p>2 = Unlock once time limited</p> <p>3 = Access profile check</p> <p>4 = Block (Prevent access)</p> <p>5 = Block with unlock once allowed</p> <p>6 = Normal operation</p> <p>7 = Block time limited</p> <p>8 = Block time limited, unlock once allowed</p> <p>A detailed description of these values can be found below.</p> |
| GID="..."              | <p>Time in seconds for commands that contain time parameter.</p> <p>These are the commands 1, 2, 7, and 8.</p>                                                                                                                                                                                                                                                                                                                                                                         |

**0 = Release permanently**

Opens the doors at the highlighted door components directly after processing the order. The door remains permanently open.

**1 = Release time limited**

Opens the doors at the highlighted door components directly after processing the order. The door remains open until the indicated timeout has expired.

**2 = Unlock once time limited**

Opens the doors at the marked door components for the specified time once although the doors are locked by order with parameter 4 or 7. The opening time ends at the latest when the specified guard time has elapsed.

**3 = Access profile check**

Closes the doors at the highlighted door components directly after processing the order. The door remains permanently closed. Authorized media can open the door. This order cancels the "Open" order (see above). The access profile check contains **no** office mode.

**4 = Block (Prevent access)**

Closes the doors at the highlighted door components directly after processing the order. The door remains permanently closed. Authorized media cannot open the door.

To enable authorized media to open the door again, this order must be transmitted with parameter 3 or 6.

**5 = Block with unlock once allowed**

The door can be opened once within a guard time according to 4. via an order with the parameter 2.

**6 = Normal operation**

This order cancels the orders "Block" and "Block temporally" (see above). After that, authorized media can open the door again. The normal operation includes also the office mode.

**7 = Block time limited**

Closes the doors at the highlighted door components directly after processing the order. The door remains closed until the indicated timeout has expired. Authorized media cannot open the door. To enable authorized media to open the door again within the timeout, this order must be transmitted with parameter 3 or 2.

**8 = Block time limited, unlock once allowed**

The door can be opened once within a guard time according to 7. via an order with the parameter 2.

## 4.16 CardLink wireless related Helper commands

### 4.16.1 WriteCardLinkConfiguration

Writes a CardLink configuration for a client <client> and an (optional) stream <stream>.

#### a) Example for a complete configuration for both BCSCW01 and BCWLS02

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
Command="WriteCardLinkConfiguration" Client="...">
<CardLinkConfiguration Client="..." Stream="BCSCW01">
  <AdministrationArea Index="..." ...>
    ...
    <Actuator ...>
      ...
    </Actuator>
    ...
  </AdministrationArea>
</CardLinkConfiguration>
<WirelessConfiguration Client="..." Stream="BCWLS02"
Mode="CardLink">
  <Gateway Index="..." ...>
    <Actuator ...>
      ...
    </Actuator>
    ...
  </Gateway>
  ...
</WirelessConfiguration>
</HelperCommand>
```

**b) Example for a BCSCW01 configuration**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkConfiguration" Client="..."
  Stream="BCSCW01">
  <CardLinkConfiguration Client="..." Stream="BCSCW01">
    <AdministrationArea Index="..." ...>
      ...
      <Actuator ...>
        ...
      </Actuator>
      ...
    </AdministrationArea>
  </CardLinkConfiguration>
</HelperCommand>
```

**c) Example for a BCWLS02 configuration**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkConfiguration" Client="..."
  Stream="BCWLS02">
  <WirelessConfiguration Client="..." Stream="BCWLS02"
  Mode="CardLink">
    <Gateway Index="..." ...>
      <Actuator ...>
        ...
      </Actuator>
      ...
    </Gateway>
    ...
  </WirelessConfiguration>
</HelperCommand>
```

### 4.16.2 WriteCardLinkGatewayConfiguration

Writes a configuration for one or more gateways for a stream <stream> of a client <client>.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
  Command="WriteCardLinkGatewayConfiguration"
  Client="..." Stream="BCWLS02">
  <WirelessConfiguration Client="..." Stream="BCWLS02">
    <Gateway Name="..." Index="...">
      <NetworkSettings ... />
    <WebServer ... />
    <User ... />
    <Actuator ...>
      ...
    </Actuator>
  </Gateway>
  ...
</WirelessConfiguration>
</HelperCommand>
```



### 4.16.3 UpdateCardLinkGatewayConfiguration

Update an existing CardLink gateway.

All specified attributes of an element will be replaced if they already exist or created if they don't exist. By specifying an empty element, the existing element(s) can be deleted, e.g. specifying `<Actuator />` will delete all existing Actuator elements. That can be used to first delete all existing elements and then add some new elements.

**Example:**

```
<?xml version="1.0" ?>
<HelperCommand
  Command="UpdateCardLinkGatewayConfiguration"
  Client="..." Stream="BCWLS02" >
  <WirelessConfiguration Client="..." Stream="BCWLS02" >
    <Gateway Index="..." Name="..." ...>
      <NetworkSettings ... />
      <Actuator LogicalName="..." GUID="..." ... >
        ...
      </Actuator>
      ...
    </Gateway>
    ...
  </WirelessConfiguration>
</HelperCommand>
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="UpdateCardLinkGatewayConfiguration" Client="..."
  Stream="BCWLS02" Status="ACK" />
```

#### 4.16.4 ReadKnownGateways

Reads the "gateways.ini" file, in which all gateways that have transmitted an UPnP datagram with MAC and IP address have been entered.

The reply is a list of gateways that have been found.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadKnownGateways" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="ReadKnownGateways"
Status="ACK">
  <Gateway MACAddress="00:15:BC:11:22:33"
IPAddress="10.10.10.3" />
  <Gateway MACAddress="00:15:BC:11:22:38"
IPAddress="10.10.10.8" />
</HelperAnswer>
```

### 4.16.5 ReadGateways

Lists all gateways of the stream "BCWLS02" of the <client> with their defined parameters.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadGateways"
Client="<client>" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="ReadGateways"
Client="Whitelist" Stream="BCWLS02" Status="ACK">

  <Gateway Index="1" Name="GW1-Whitelist"
IPAddress="<ip address>" MACAddress="<mac
address>" Subnet="20" StandardGateway="<ip
address>" PrimaryDNS="<ip address>"
SecondaryDNS="<ip address>" DHCPActive="true"
CommissioningOpenTime="3600"
RequestRetryTime="120" LinkQualityEvents="false"
RangeMeasurementActive="false" TimeZoneld="429"
TimeZone="GMT+01:00 (Europe/Berlin)"
TimeServerActive="false" KateServerActive="true"
KateServerPort="9000" WebServerActive="true"
WebServerPort="443" FirmwareVersion="4.2.1.9616
2017-03-07" OSVersion="Linux 3.12.18"
RootFSVersion="1.2.101-1.2.15-kaba20160610"
ZigBeeVersion="3.9.100" ZigBeeChannel="24"
HardwareVersion="3.0.1" />

</HelperAnswer>
```

### 4.16.6 ReadActuatorJoinStates

Lists the "join status" of all components of the <gateway IP address> of the <client>

**or**

the "join status" of all components of all gateways of the <client>.

#### a) Example "join status" of all components of all gateways of the <client>

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorJoinStates"
Client="Whitelist" />
```

#### Response:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="ReadActuatorJoinStates"
Client="Whitelist" Stream="BCWLS02" Status="ACK">

  <Gateway IPAddress="10.10.12.22" Name="GW1-
Whitelist" Index="1">

    <Actuator LogicalName="ADVANT-WIRELESS-CLEVER"
GUID="78CFD79C-9C08-49C8-8B6B-DD8E6BD9AAE1"
JoinState="joined"
JoinGatewayIPAddress="10.10.12.22"
JoinGatewayName="GW1-Whitelist" />

    <Actuator LogicalName="TEST"
GUID="0F695914-26F7-4CAD-BC67-1528F6983C5B"
JoinState="disjoined" />

  </Gateway>
</HelperAnswer>
```

#### b) Example "join status" of all components of the <gateway IP address> of the <client>.

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadActuatorJoinStates"
Client="Whitelist" GatewayIPAddress="10.10.12.22" />
```

#### Response:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="ReadActuatorJoinStates"
Client="Whitelist" Stream="BCWLS02"
Gateway=10.10.12.22" Status="ACK">

  <Actuator LogicalName="ADVANT-WIRELESS-CLEVER"
GUID="78CFD79C-9C08-49C8-8B6B-DD8E6BD9AAE1"
JoinState="joined" JoinGatewayIPAddress="10.10.12.22"
JoinGatewayName="GW1-Whitelist" />

  <Actuator LogicalName="TEST"
GUID="0F695914-26F7-4CAD-BC67-1528F6983C5B"
JoinState="disjoined" />

</HelperAnswer>
```

### 4.16.7 DoorControl

See description in chapter DoorControl [► 4.15.15].

## 4.17 Biometric related Helper commands

### 4.17.1 RefreshTemplateList

Imports the common template list of **all** clients, if this is configured in the B-COMM Helper. This command shall be used, if an overall template list exists for all clients.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="RefreshTemplateList" />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="ACK" />
```

**or**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="NAK" />
```

### 4.17.2 RefreshTemplateList Client

Imports the template list of **one** client, if this is configured in the B-COMM Helper. This command shall be used, if every client has got its own template list.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="RefreshTemplateList" Client="..." />
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="ACK" />
```

**or**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Status="NAK" />
```

## 4.18 Programmer related Helper commands

### 4.18.1 ReadProgrammerStatus

This command makes it possible to query the currently connected programmer. The query may refer to the PD1364 and PD 1460 programmers.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand Command="ReadProgrammerStatus"/>
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer Command="ReadProgrammerStatus" PD1364Connected="true"
PD1460Connected="false" Status="ACK" />
<HelperAnswer Status="ACK">
<HelperAnswer>
```

Possible responses are:

PDxxxxConnected="true".

**Meaning:** Programmer is connected.

PDxxxxConnected="false".

**Meaning:** Programmer is not connected.

## 4.19 elologic related Helper commands

### 4.19.1 WriteElolegicActuatorConfiguration

Writes an elologic component configuration to the <stream> of the <client> and sets the defined parameters.

Any possibly existing configuration will be overwritten.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
Command="WriteElolegicActuatorConfiguration"
Client="..." Stream="...">
  <ElolegicConfiguration Client="..." Stream="...">
    <Actuator Changed="..." ...>
  </Actuator>
</ElolegicConfiguration>
</HelperCommand>
```

### 4.19.2 WriteElolegicKeyPlanConfiguration

Writes an elologic locking plan configuration to the <stream> of the <client> and sets the defined parameters.

Any possibly existing configuration will be overwritten.

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<HelperCommand
Command="WriteElolegicKeyPlanConfiguration"
Client="..." Stream="...">
  <ElolegicConfiguration Client="..." Stream="...">
    <KeyPlan Changed="..." Index="...">
    ...
  </KeyPlan>
</ElolegicConfiguration>
</HelperCommand>
```

## 4.20 Mobile Access related Helper commands

### 4.20.1 WriteWhitelistMobileAccessCredentials

Sends a list of Whitelist static credentials to the appropriate smartphones via LEGIC Connect.

**Example:**

```
<?xml version="1.0"?>
<HelperCommand Command="WriteWhitelistMobileAccessCredentials" Client="..." >
<StaticCredentials>
  <Credential FileDefinitionName="staticCred" PhoneNo="+491519876543210"
    CardId="2041" IsHex="false" BleAccess="true" NfcAccess="true" />
  ...
</StaticCredentials>
</HelperCommand>
```

|                    |                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Client             | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used.                                 |
| FileDefinitionName | Defines the file definition name that is to be used for this credential.                                                                      |
| PhoneNo            | Phone number of the smartphone in format +xyyyyyyyyyy (xx = country code, yyyyyyyyyy = phone number without leading 0 or phone#+xyyyyyyyyyy). |
| CardId             | CID/UID of the media                                                                                                                          |
| IsHex              | "True" or "False", if true, the CID is interpreted as a hexadecimal value. Default: false                                                     |
| BleAccess          | "True" or "False", if true user can make an access with BLE. Default: true                                                                    |
| NfcAccess          | "True" or "False", if true user can make an access with NFC. Default: true                                                                    |



## 4.20.2 DeleteWhitelistMobileAccessCredentials

Delete a list of Whitelist static credentials from the appropriate smartphones via LEGIC Connect.

**Example:**

```
<?xml version="1.0"?>
<HelperCommand Command="DeleteWhitelistMobileAccessCredentials" Client="...">
  <StaticCredentials>
    <Credential PhoneNo="+491519876543210" />
    ...
  </StaticCredentials>
</HelperCommand>
```

|         |                                                                                                                                               |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Client  | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used.                                 |
| PhoneNo | Phone number of the smartphone in format +xyyyyyyyyyy (xx = country code, yyyyyyyyyy = phone number without leading 0 or phone#+xyyyyyyyyyy). |

### 4.20.3 CreateMobileAccessProjectDevices

Create one or more project devices (smartphones) in LEGIC Connect for a project.

**Example:**

```
<?xml version="1.0"?>
<HelperCommand Command="CreateMobileAccessProjectDevices" Client="..." >
<Devices>
  <Device PhoneNo="+491519876543210" Name="Test Device 1" />
  ...
</Devices>
</HelperCommand>
```

|         |                                                                                                                                                 |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Client  | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used.                                   |
| PhoneNo | Phone number of the mobile phone in format +xyyyyyyyyyy (xx = country code, yyyyyyyyyy = phone number without leading 0 or phone#+xyyyyyyyyyy). |
| Name    | A name for the project device.                                                                                                                  |

### 4.20.4 DeleteMobileAccessProjectDevices

Remove one or more project devices (smartphones) from LEGIC Connect for a project.

**Example:**

```
<?xml version="1.0"?>
<HelperCommand Command="DeleteMobileAccessProjectDevices" Client="..." >
<Devices>
  <Device PhoneNo="+491519876543210" />
  ...
</Devices>
</HelperCommand>
```

|         |                                                                                                                                                 |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Client  | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used.                                   |
| PhoneNo | Phone number of the mobile phone in format +xyyyyyyyyyy (xx = country code, yyyyyyyyyy = phone number without leading 0 or phone#+xyyyyyyyyyy). |

### 4.20.5 ListMobileAccessProjectsDevices

Lists all project devices (smartphosen) from LEGIC Connect.

**Example:**

```
<?xml version="1.0"?>
<HelperCommand Command="ListMobileAccessProjectsDevices" Client="..." />
</HelperCommand>
```

|        |                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------|
| Client | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used. |
|--------|---------------------------------------------------------------------------------------------------------------|

## 4.20.6 ReadLegicConnectEvents

Reads the LEGIC Connect events from the FIFO queues of the B-COMM Server.

### Example:

```
<?xml version="1.0"?>
<HelperCommand Command="ReadLegicConnectEvents" Client="..." />
</HelperCommand>
```

|        |                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------|
| Client | The name of the client for the according LEGIC Connect project. If omitted, the default project will be used. |
|--------|---------------------------------------------------------------------------------------------------------------|

### Response:

```
<?xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="ReadLegicConnectEvents" Status="ACK">
<Event Name="ProvisionedFileChange" EventId="1099517888411"
Timestamp="2020-12-14T11:07:01.547+01:00"
ProjectId="42384508" MobileAppId="37584157"
RequestId="CorrId_5227829501607685184_434781"
ChangeType="CREATE" FileState="adding" FileDefinitionName="staticCred"
PhoneNo="phone#+491718583971"
IsVcp="false" /> <Event Name="ProvisionedFileChange" EventId="1099517888414"
Timestamp="2020-12-14T11:08:00.174+01:00" ProjectId="42384508"
MobileAppId="37584157"
RequestId="CorrId_5227829501607685184_434781" ChangeType="MODIFY"
FileState="deployed"
FileDefinitionName="staticCred" PhoneNo="phone#+491718583971" IsVcp="false" />
</HelperAnswer>
```

|                    |                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EventName          | Name of the event: <b>ProvisionedFileChange</b> , <b>ProjectDeviceChange</b> , <b>VcpChange</b> or <b>Error</b>                                                                                                                                                      |
| EventId            | ID of the event                                                                                                                                                                                                                                                      |
| TimeStamp          | Time/date when the event occurred                                                                                                                                                                                                                                    |
| ProjectId          | ID of the corresponding LEGIC Connect project                                                                                                                                                                                                                        |
| MobileAppId        | ID of the corresponding mobile App                                                                                                                                                                                                                                   |
| RequestId          | ID of the request which caused the event                                                                                                                                                                                                                             |
| ChangeType         | Change type of the event: <b>CREATE</b> , <b>MODIFY</b> or <b>DELETE</b>                                                                                                                                                                                             |
| FileState          | State of the corresponding file: <b>requestAdd</b> , <b>rejected</b> , <b>approved</b> , <b>requestRemove</b> , <b>adding</b> , <b>updating</b> , <b>deployed</b> , <b>addFailed</b> , <b>updateFailed</b> , <b>removeFailed</b> , <b>removing</b> or <b>removed</b> |
| FileDefinitionName | Name of the corresponding file definition, usually <b>staticCred</b>                                                                                                                                                                                                 |
| PhoneNo            | Corresponding phone number                                                                                                                                                                                                                                           |
| IsVcp              | <b>True</b> if it's a VCP, <b>false</b> if it's a credential file                                                                                                                                                                                                    |

## 4.20.7 ListMobileAccessVcps

Gets a list of the VCPs of a LEGIC Connect project.

```
<?xml version="1.0"?>
```

```
<HelperCommand Command="ListMobileAccessVcps" [Client="..."] />
```

If the attribute Client is omitted, the VCPs of the default LEGIC Connect project will be listed, else the VCPs of the project assigned to the client will be listed.

### Response:

```
<xml version="1.0" encoding="iso-8859-1">
```

```
<HelperAnswer Command="ListMobileAccessVcps" [Client="..."] Status="ACK">
```

```
<Vcps>
```

```
<Vcp Name="..." Id="..." ProjectId="..." ReaderSeries4000="true|false"  
ReaderSeries6000="true|false" />
```

```
</Vcps>
```

```
</HelperAnswer>
```

## 4.20.8 ListMobileAccessProvisionedFiles

Gets a list of the files (VCPs and credentials) of a LEGIC Connect project

### a) list of all files of the default project

```
<?xml version="1.0"?>
<HelperCommand Command="ListMobileAccessProvisionedFiles" />
```

### b) list of all files of the project assigned to the specified client

```
<?xml version="1.0"?>
<HelperCommand Command="ListMobileAccessProvisionedFiles" Client="..." />
```

### c) list of the files of the specified smartphone(s) of the default project

```
<?xml version="1.0"?>
<HelperCommand Command="ListMobileAccessProvisionedFiles" >
  <ProvisionedFiles>
    <ProvisionedFile PhoneNo="+49151123456789" />
    ...
  </ProvisionedFiles>
</HelperCommand>
```

### d) list of the files of the specified smartphone(s) of the project assigned to the specified client

```
<?xml version="1.0"?>
<HelperCommand Command="ListMobileAccessProvisionedFiles" Client="..." >
  <ProvisionedFiles>
    <ProvisionedFile PhoneNo="+49151123456789" />
    ...
  </ProvisionedFiles>
</HelperCommand>
```

**Response:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<HelperAnswer
Command="ListMobileAccessProvisionedFiles"
[Client="..."] Status="ACK">
<RegisteredMobileApps>
  <RegisteredMobileApp PhoneNo="..." MobileAppId="..." >
    <ProvisionedFiles>
      <ProvisionedFile FileId="..." IsVcp="true|false"
ModificationDate="..." FileState="..."
RfInterfaces="..." />
      ...
    </ProvisionedFiles>
  </RegisteredMobileApp>
  ...
</RegisteredMobileApps>
</HelperAnswer>
```

### 4.20.9 DeleteMobileAccessVcps

Removes all VCPs from specified smartphone(s) of a LEGIC Connect project.

```
<?xml version="1.0"?>
<HelperCommand Command="DeleteMobileAccessVcps" [Client="..."] >
  <Vcps>
    <Vcp PhoneNo="+49151123456789" />
    ...
  </Vcps>
</HelperCommand>
```

If the attribute Client is omitted, the VCPs of the default LEGIC Connect project will be removed from the specified smartphone(s), else the VCPs of the project assigned to the client will be removed from the specified smartphone(s).

**Response:**

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="DeleteMobileAccessVcps" [Client="..."] Status="ACK">
  <Vcps>
    <Vcp Name="..." Id="..." ProjectId="..." ReaderSeries4000="true|false"
      ReaderSeries6000="true|false" />
  </Vcps>
</HelperAnswer>
```



## 4.20.10 WriteMobileAccessVcps

Deploys all VCPs to specified smartphone(s) of a LEGIC Connect project.

```
<?xml version="1.0"?>
<HelperCommand Command="WriteMobileAccessVcps" [Client="..."] >
  <Vcps>
    <Vcp PhoneNo="+49151123456789" />
    ...
  </Vcps>
</HelperCommand>
```

If the attribute Client is omitted, the VCPs of the default LEGIC Connect project will be deployed to the specified smartphone(s), else the VCPs of the project assigned to the client will be deployed to the specified smartphone(s).

### Response:

```
<xml version="1.0" encoding="iso-8859-1">
<HelperAnswer Command="WriteMobileAccessVcps" [Client="..."] Status="ACK">
  <Vcps>
    <Vcp PhoneNo="+49151123456789" />
    ...
  </Vcps>
</HelperAnswer>
```

## 4.21 SSH key related Helper commands

### 4.21.1 SSHKeyExport

Exports SSH key of specified terminal(s) into a file.

```
<?xml version="1.0"?>
<HelperCommand Command="SSHKeyExport" Client="..." Stream="..." GID="..."
DID="..." Alternativeld="..." File="..." />
```

### Response:

```
<?xml version="1.0" encoding="iso-8859-1"
<HelperAnswer Command="SSHKeyExport" Client="..." Stream="..." Status="ACK" />
```

# 5 HTTP Helper

## 5.1 Introduction

The HTTP helper is programmed in Java and is run on a server together with B-COMM Helper (see chapter [▶ 4.1](#)) under B-COMM. The version numbering of HTTP helper follows the B-COMM main application numbering. The current version is 5.4.0 as of manufacturing date May 2023.

The main task of the HTTP helper is the processing of orders from a third system and their transmission to B-COMM as well as the transfer of received BPA data records from the terminals to a third party system.

Applications from a software partner (SWP) or the dormakaba solution EACM come into questions as a third party system. In the following description, both possibilities are therefore mentioned together (EACM/SWP).

In this documentation, dormakaba EACM is used as a third party system.

It is also possible to transmit the orders and BPA data records in an encrypted form via HTTPS if the required certificates are available and the interfaces are parameterized accordingly.

In addition, the HTTP helper offers the possibility to transmit biometric templates to B-COMM. For this reason, terminals with biometric readers can also be supplied with templates.

The orders can be transmitted as individual orders to the I/O and are acknowledged by pressing <ACK> on the terminal.

It is also possible to combine and transmit several orders in one order file.

The orders are acknowledged by an XML document with the corresponding response codes, see chapter [▶ 5.3.2](#).

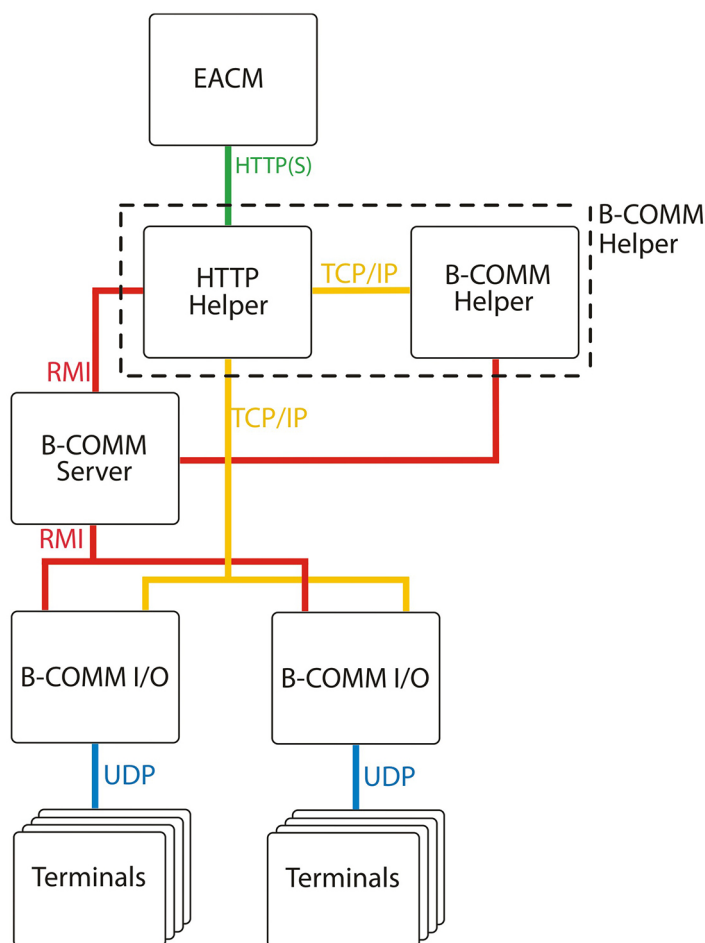
The HTTP helper is able to communicate internally with the B-COMM helper that is integrated in B-COMM. Using this communication way commands and orders used in the B-COMM helper can also be executed via the HTTP helper. For this purpose the B-COMM helper must have been started.

An example on this is given in chapter [▶ 5.5.2.8](#).

HTTP helper and B-COMM helper use the same log files.

## 5.2 Implementation of the HTTP helper into the system

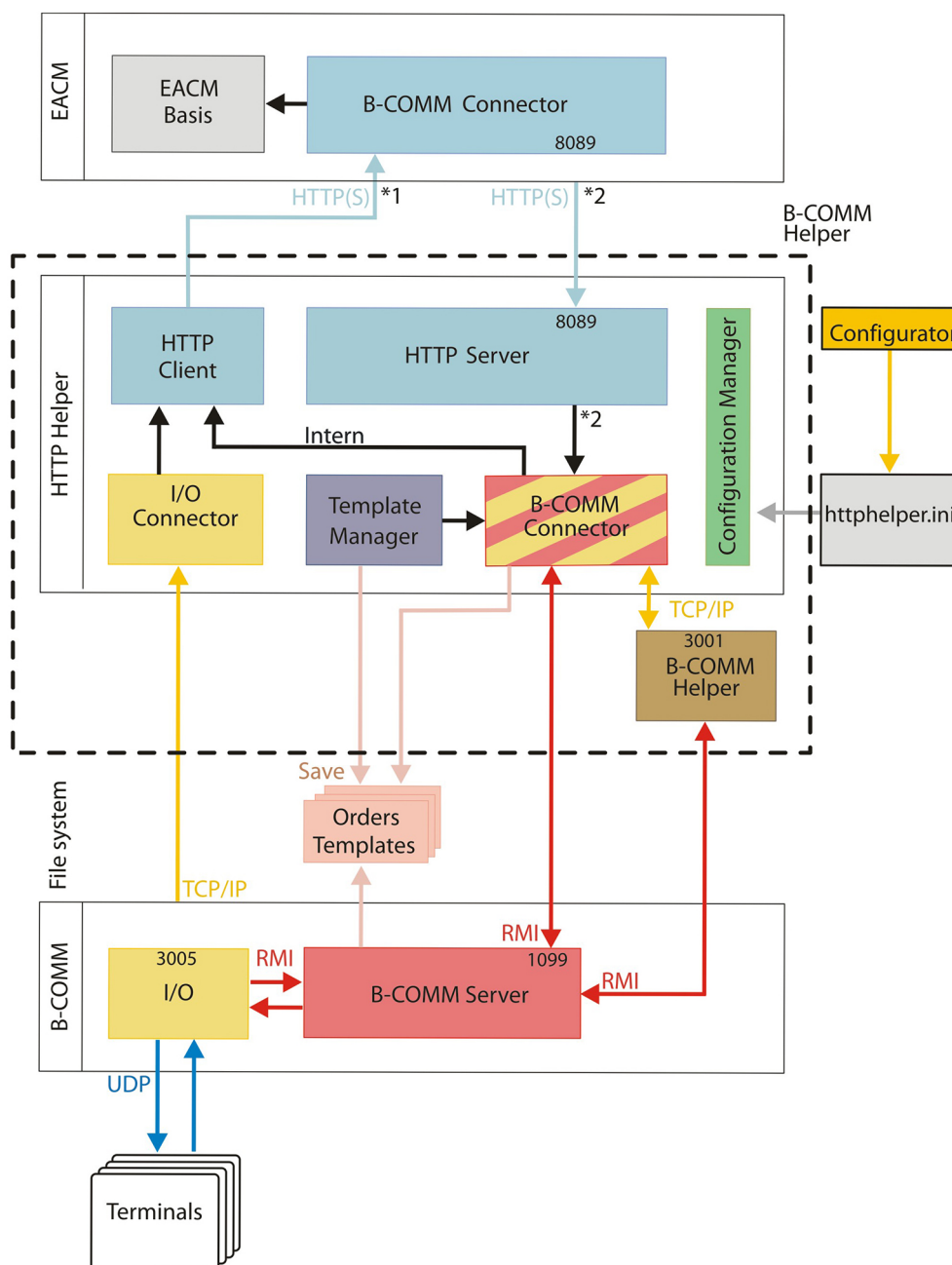
The HTTP helper is implemented into a system environment as follows:



| Application   | Communication        | Task                                                                                |
|---------------|----------------------|-------------------------------------------------------------------------------------|
| EACM or SWP   | HTTP(S)              | Generation of master records, processing of bookings, enrolling of templates.       |
| HTTP Helper   | HTTP(S), RMI, TCP/IP | Data transfer from EACM/SWP or to EACM/SWP and, among others, generation of orders. |
| B-COMM Helper | TCP/IP, RMI          | Commands and orders of the B-COMM helper can be used via HTTP helper.               |
| B-COMM server | RMI                  | Management of the terminals.                                                        |
| B-COMM GUI    | RMI                  | Configuration of the terminals.                                                     |
| B-COMM I/O    | TCP/IP, UDP/HTTP     | Communication with the terminals.                                                   |
| Terminals     | UDP/HTTP             | Generation of bookings, enrolling of templates.                                     |

## 5.3 Overview of the HTTP helper communication

The following diagram shows the communication paths and the protocols used. The application used was the dormakaba solution EACM.



\* See following description.

|    |                                                                                                                                                                                                                                                                                                                                                               |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1* | Transfer of terminal data (bookings) and biometric templates (see chapter <a href="#">▶ 5.5.1.1</a> )                                                                                                                                                                                                                                                         |
| 2* | Transmission of the following helper commands (see chapter <a href="#">▶ 5.5.2.1</a> and the following) <ul style="list-style-type: none"> <li>• get_bc_config</li> <li>• send_data</li> <li>• start_stream</li> <li>• stop_stream</li> <li>• add_bio_template</li> <li>• delete_bio_template</li> <li>• get_all_bio_templates</li> <li>• send_xml</li> </ul> |

### 5.3.1 Components of the HTTP helper

In this chapter, the individual components of the HTTP helper are listed and their function is explained.

#### Overview

| Component             | Task                                                                          |
|-----------------------|-------------------------------------------------------------------------------|
| Configuration Manager | Reads the configuration of the HTTP helper.                                   |
| HTTP Server           | Waits for commands from EACM/SWP.                                             |
| HTTP Client           | Transfers the BPA data records to EACM/SWP.                                   |
| B-COMM Connector      | Communicates via RMI with B-COMM. Communicates via TCP/IP with B-COMM Helper. |
| I/O Connector         | Communicates via TCP/IP with the I/Os.                                        |
| Template Manager      | Monitors whether the newly enrolled templates are saved in B COMM.            |

#### 5.3.1.1 Configuration Manager

The Configuration Manager reads the information which has been set in the configurator. They are stored in the "httphelper.ini" file. This information is only read during program start. Further information regarding the Configuration Manager can be found in chapter [▶ 5.4.3](#).



The HTTP helper will start only if the "httphelper.ini" file is stored in the same directory as the helper itself.

### 5.3.1.2 HTTP Server

The HTTP server is activated when the program is started. EACM/SWP sends orders to the HTTP server with the following functions:

- Inquiry of the B-COMM terminals with all relevant information
- Sending of master records to all terminals via B-COMM
- Starting/stopping of streams
- Addition/deletion of biometric templates in the B-COMM server
- Upload of all biometric templates from B-COMM to EACM/SWP

Commands that have been received are prepared and transmitted to program parts involved. EACM/SWP always receives the answer in form of an XML file.

### 5.3.1.3 HTTP Client

The HTTP client sends the data records of the IO connector to the address of EACM/SWP via HTTP(S).

### 5.3.1.4 B-COMM Connector

The B-COMM connector communicates with B-COMM via RMI.

The B-COMM connector communicates with the B-COMM helper via TCP/IP.

The B-COMM connector has the following tasks:

1. Determination of the terminal configuration in B-COMM and representation of the information in form of an XML file.

The response to a configuration inquiry includes:

- Clients
- Channels (streams)
- Network adapters
- Terminals

2. Transfer of orders from EACM/SWP to B-COMM for distribution to the terminals.

The BPA data records are stored in order files and the file path is transmitted to B-COMM. As an alternative, the individual master records can be sent directly to the terminals. The order files are deleted by the HTTP helper after having been processed by B-COMM.

3. Starting and terminating of individual communication channels (streams) in B-COMM.
4. Sending and saving of templates.
5. Transfer of XML helper commands to the B-COMM Helper via TCP/IP.

### 5.3.1.5 I/O Connector

The I/O connector offers a port which can be used for the connection of the I/Os via TCP/IP.

If an I/O is connected to the I/O connector, the connection is kept permanently open.

The I/O connection represents a channel (stream) of a client.

The BPA data records received are transmitted to the HTTP client by the I/O connector and the stream and the client of the I/O are added to the information so that the data record can be assigned unambiguously to a terminal.

### 5.3.1.6 Template Manager

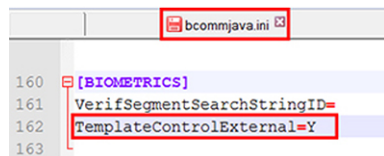
The HTTP helper also supports biometrically relevant orders. For this, B-COMM signals to EACM/SWP whether a terminal is capable to support biometric functions. In the response to the command `get_bc_config` (see chapter [5.5.2.1](#)) the value `Biometric="X"` (= true) can be found.

All templates available for EACM/SWP must be transmitted to the B-COMM server. All templates newly enrolled via EACM/SWP at a later stage must also be transmitted to the B-COMM server.

EACM/SWP indicates to the B-COMM server via the HTTP helper which templates must be distributed to the terminals.

For this, the parameter "TemplateControlExternal" must be set in the section [BIOMETRICS] of the "bcommjava.ini" file.

If the parameter is not available, it can also be generated manually via an editor.



```
bcommjava.ini
160 [BIOMETRICS]
161 VerifSegmentSearchStringID=
162 TemplateControlExternal=Y
163
```

B-COMM will transmit it to the terminals.



If the parameter "TemplateControlExternal=Y" is set, the biometric menu in B-COMM is put out of function. This is necessary to avoid a double management of the templates (in B-COMM and EACM/SWP). The template distribution in B-COMM is not affected by this.

If a new template is enrolled at a terminal, it is automatically transmitted to EACM/SWP through the HTTP helper. Afterwards it is deleted from the intermediate directory in B-COMM.

The template must then be transmitted from EACM/SWP to the B-COMM server to ensure that it can be used.

### 5.3.2 Response codes

After the processing, EACM/SWP will receive an answer from the HTTP helper with a response code giving a statement whether the order has been processed successfully or faultily.

Instead of a response code (only for `get_bc_config`), an XML document with the inquired data can also be sent.

#### Overview

| Response code | Description                               |
|---------------|-------------------------------------------|
| 200           | Processing successful                     |
| 300           | Error and data having triggered the error |
| 310           | Function not implemented                  |
| 311           | Biometric functions not activated         |
| 320           | Incomplete/missing data                   |
| 400           | Error during processing                   |
| 401           | Error during basic authorization          |
| 500           | Error during communication                |



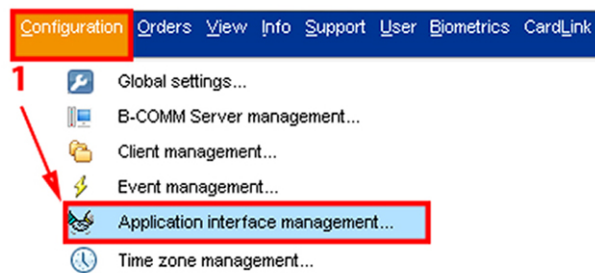
## 5.4 Performing the configuration

1. Generate the application interface in B-COMM.
2. Make the certificates and corresponding passwords available.
3. Configure the HTTP helper with the configurator.
4. Configure the interface in EACM or for SWP (not described in this manual).  
The configuration in EACM can be found in the respective EACM installation manual.
5. Send the order get\_bc\_config to request the terminal configuration of B-COMM, see chapter [▶ 5.5.2.1](#).

### 5.4.1 Creating an application interface

For the data transmission between B-COMM I/O and the I/O connector in the HTTP helper, a defined application interface is required. This interface is generated in B-COMM as follows. Further information on the application interface can be found in the related chapter in the B-COMM manual.

1. In the menu "Configuration", select the menu item "Application interface management".



2. In this screen, define the application interface(s) and add it (them) to the list.

Please note, that the default port number for HTTP connection is 4005.

**Application interface management**

**2** Application interface

Name: EACM-SWP

☒ IP address: 10.10.10.160

☐ DNS name: 10.10.10.160

Port: 4005 (Default: 4005)

Timeout: 60 Sec.

Connect interval: 10 Sec.

Connect repeat: 3

Connect delay: 0 Sec.

RT/RTM list: A1; (Separate several RT/RTM by semicolon)

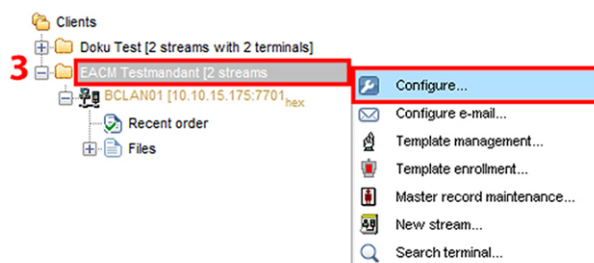
Application interface list

|          |
|----------|
| EACM-SWP |
|----------|

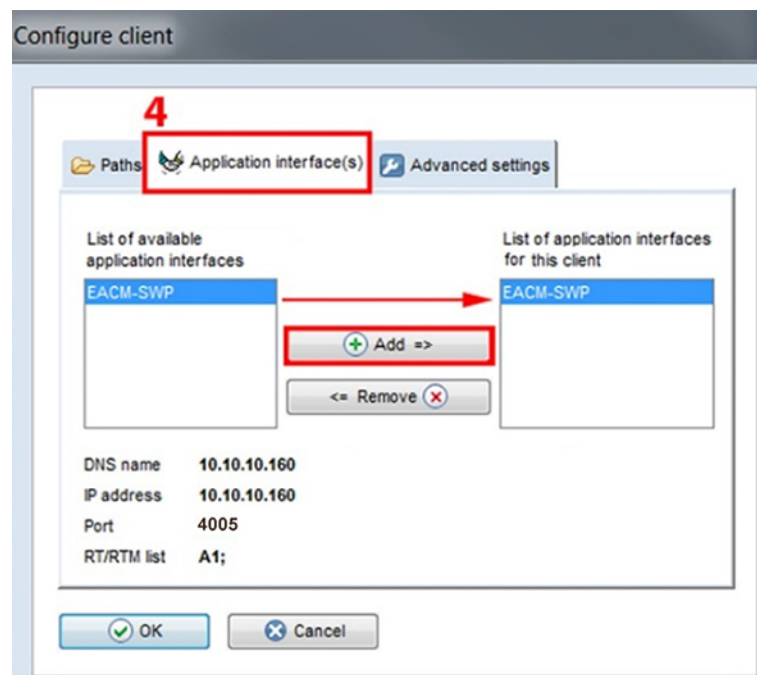
Buttons: Add, Save, Delete, Rename

Cancel

3. Open the context menu for the desired client and select the menu item "Configure".

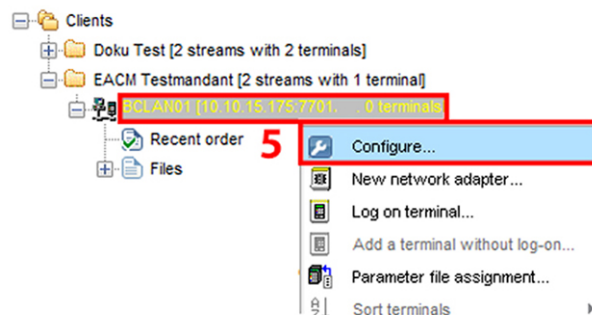


4. Select in the "Application interface" tab the desired application interface and add it to the list of the application interfaces by pressing the "Add" button.

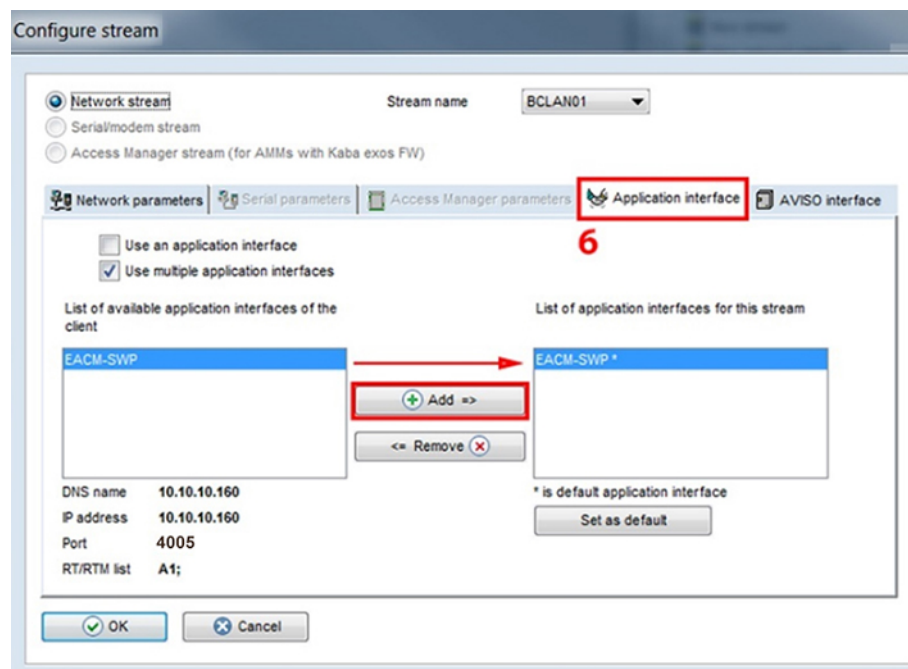


Save the selection by pressing "OK".

5. The application interface can also be assigned to an individual stream. Open the context menu for the desired stream and select the menu item "Configure".



6. Select the desired application interface and add it to the list of the application interfaces for this stream by pressing the "Add" button.



Save the selection by pressing "OK".

### 5.4.2 Provision of the certificates

The communication between EACM/SWP and the HTTP helper can also be performed via HTTPS.

The certificates required for this must be made available by the user in an appropriate way and specified during configuration, see chapter [5.4.3](#) item 4 and 8.

The procedures regarding the creation and provision of certificates are not described in this manual.

If a helper certificate (PKCS12) has been stated during configuration, a test is made before saving whether the certificate can be opened with the specified password.

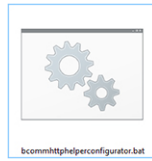
The respective passwords must be entered again if the helper certificate, the EACM/SWP server or the EACM/SWP user has been changed.

It is not allowed to change the paths for the certificates during operation since the certificates are read during runtime.

### 5.4.3 Configuring the HTTP helper

Double-click the "bcommhttphelperconfigurator.bat" file.

The call of the batch file can be extended with a <-f> parameter to define the storage location of the "httphelper.ini" file, depending on the used installation mode. Please see the notes at the end of this chapter for more information.



The configurator is started. The information described below must be entered (for the numbers 3, 4, 6 and 8, the information can be provided optionally).

#### Calling the "bcommhttphelperconfigurator.bat"

If in a B-COMM installation customer-specific directories are used (meant: not the default directories), the call of the "bcommhttphelperconfigurator.bat" can be extended with the <-f> parameter and the used storage path of the "httphelper.ini" file.

#### Example:

```
bcommhttphelperconfigurator.bat -f C:/Customerdirectory/dormakaba/B-COMM_5.4.0
```

| No. | Description                                                                                                                                                                                                                                                                                                                                                      | Default |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 1   | <b>HTTP(S) Port</b><br>Port through which the HTTP helper can be accessed by EACM or SWP via HTTP(S). This is the same port as defined in the SWP application or in the Helper-Demo respectively.                                                                                                                                                                | 8089    |
| 2   | <b>Application port</b><br>Port through which B-COMM can access the HTTP helper. This is the same port as defined in the "Application Interface-Port" setting in the B-COMM main application. See the related chapter in the B-COMM documentation for more information. Please note, that the default port number for HTTP is 4005.                              | 4005    |
| 3   | <b>Certificate (server certificate)</b><br>Path to the PKCS12 server certificate, which is used by the HTTP helper for SSL communication. The specified path must not be changed after having saved it since the certificate is loaded during runtime.                                                                                                           |         |
| 4   | <b>Certificate password</b><br>Password for the PKCS12 certificate file.                                                                                                                                                                                                                                                                                         |         |
| 5   | <b>Helper name tag<sup>1)</sup></b><br>Name extension for the HTTP helper. The helper name tag is added from the HTTP helper to EACM/SWP if terminal data or templates are transmitted. The helper name tag defines the B-COMM server the data is sent from. If several B-COMM servers are connected the helper name tag must be unique for every B-COMM server. |         |
| 6   | <b>Enable biometrics</b><br>If the checkbox is ticked, the biometric functions of the HTTP helper are activated.                                                                                                                                                                                                                                                 | false   |
| 7   | <b>Server URL<sup>1)</sup></b><br>URL through which the EACM or SWP can be accessed. Please note, that the URL must begin with <http://>, otherwise the configurator will generate an error.                                                                                                                                                                     |         |
| 8   | <b>Certificate (Client certificate)</b><br>The client certificate which is used by the SAP server for the SSL communication. The specified path must not be changed after having saved it since the certificate is loaded during runtime.                                                                                                                        |         |
| 9   | <b>User<sup>1)</sup></b><br>EACM/SWP communication user.                                                                                                                                                                                                                                                                                                         |         |
| 10  | <b>Password<sup>1)</sup></b><br>Password of the EACM/SWP communication user.                                                                                                                                                                                                                                                                                     |         |

For the parameters marked with <sup>1)</sup>, the entry is mandatory.

### Connection Test

Use the button "Connection test" to check the communication from the configurator to EACM/SWP or HTTPHelper Demo respectively. This test may take up to one minute. If the HTTPHelper Demo is used for connection test, it is necessary to add the port <8001> to the IP address.

**Example:** http://10.10.15.184:8001.

The input fields "User" and "Password" may not left empty and must contain at least some dummy data for the connection test.

For the "Connection test" neither B-COMM nor the B-COMM helper has to run, but only the communication partner such as HTTPHelper demo.

### 5.4.4 Checking the "httphelper.ini" file

After saving the configuration screen, an "httphelper.ini" file is created.

Depending on the used installation mode, the storage location of this file can be:

A) Common installation with no change of the default directory:

Program directory of the configurator. E.g.: C:/Programs/dormakaba/B-COMM

B) Separate: installation with no change of the default directory:

Home directory of the user + /dormakaba/B-COMM. E.g.: <user.name>/dormakaba/B-COMM

C) Customer specific:

Depending on the user defined directory. Calling with -f parameter, see chapter [5.4.3](#).

The "httphelper.ini" file is read by the HTTP helper during program start. The "httphelper.ini" file and the HTTP helper must be saved in the same directory.

Specified passwords are only stored in an encrypted form.

The parameter "Bio" in the [BCOMM] section is set to "true" if the check box "Enable biometrics" in the configuration screen has been marked.

```
[HELPER]
HTTPPort=8089
IOPort=3007
Certificate=C:\server.p12
Password=jTxq3n7PDolUzJdf/I+ViA==1pGmdU4w/vzzPMKNcW4Jlg==
NameTag=BCOMM1

[SERVER]
URL=http://10.10.0.184:8001/kabaawm/bc_connect
User=EACM_BC_USR
Authentication=gH3fzrw/Xe++2Rdl59Cuyg==wmLpKBdD4a4dCGxkZ7aHzCahh7KYDpJmy
Certificate=

[BCOMM]
Bio=true
```

## 5.5 Supported HTTP helper commands

### 5.5.1 Transmission direction B-COMM to HTTP helper and then to EACM/SWP

#### 5.5.1.1 Terminal data (bookings) and newly enrolled templates

##### Task

Transmission of bookings of a terminal or templates newly enrolled at a terminal.

##### Syntax for the receipt of terminal data (bookings):

```
POST /?nameTag=<Helper name tag> &type=data&client=EACM-ZE &stream=BCLAN02
&data=<BPA Daten>
```

HTTP/1.1

Content-Type: text/plain

Authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=

Cache-Control: no-cache

Pragma: no-cache

User-Agent: Java/11.0.1

Host: 127.0.0.1:8888

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

##### Syntax for the receipt of templates:

```
POST /?nameTag=<Helper name tag> &type=template&templateId=0060096
```

```
&pkc1d=<Template data>
```

```
&pkc2d=<Template data>
```

```
&quality1=124
```

```
&quality2=120
```

```
&enrollDate=20181029
```

```
&enrollTime=094749
```

HTTP/1.1

Content-Type: text/plain

Authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=

Cache-Control: no-cache

Pragma: no-cache

User-Agent: Java/11.0.1

Host: 127.0.0.1:8888

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

##### Response:

HTTP Response Code 200 with response message "OK".



## 5.5.2 Transmission direction EACM/SWP to HTTP helper to B-COMM

### 5.5.2.1 get\_bc\_config

#### Task

Requesting the terminal system of the <client>.

#### Syntax (example)

To the helper IP address:port (110.178.56.1:8089).

POST get\_bc\_config HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089

accept-encoding: gzip

sap-language: D

authorization: Basic RUFDTV9CQ19VU1l6ODc2NTQzMjE=

#### Response

XML document with the requested data, see example.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<HelperAnswer Command="get_bc_config">
  <Client ClientName="EACM-AC" ID="1"/>
  <Client ClientName="EACM-ZE" ID="2">
    <Stream ID="1" IsActive=" " ParentID="2" StreamName="BCLAN01">
      <NetworkAdapter AdapterIPAddress="10.10.13.164" ID="1" ParentID="1">
        <Terminal Biometrics=" " DID="0" Description="9600-K6 B-Client HR20" GID="1" ID="1" MACAddress="00:07:cc:ff:74:1a"
          MasterRecordIDLength="7" ParentID="1" Serial="false" TimeZoneID="Europe/Berlin" Type="9600-K6 B-Client HR20"/>
      </NetworkAdapter>
      <NetworkAdapter AdapterIPAddress="10.10.15.179" ID="2" ParentID="1">
        <Terminal Biometrics=" " DID="0" Description="9200-K5 B-Client AC30" GID="3" ID="2" MACAddress=""
          MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="9200-K5 B-Client AC30">
          <Subterminal Biometrics=" " DID="3" Description="9110-K5 (MRD)" GID="3" ID="0" MACAddress=""
            MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="Subterminal 9110-K5 (MRD)"/>
          <Subterminal Biometrics=" " DID="4" Description="9110-K5 (MRD)" GID="3" ID="0" MACAddress=""
            MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="Subterminal 9110-K5 (MRD)"/>
          <Subterminal Biometrics=" " DID="5" Description="9150-K5" GID="3" ID="0" MACAddress=""
            MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="Subterminal 9150-K5"/>
          <Subterminal Biometrics=" " DID="6" Description="9150-K5 (MRD)" GID="3" ID="0" MACAddress=""
            MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="Subterminal 9150-K5 (MRD)"/>
          <Subterminal Biometrics="X" DID="7" Description="B-Net 91 05 FP" GID="3" ID="0" MACAddress=""
            MasterRecordIDLength="7" ParentID="2" Serial="false" TimeZoneID="Europe/Berlin" Type="Subterminal B-Net 91 05 FP"/>
        </Terminal>
      </NetworkAdapter>
    </Stream>
    <Stream ID="2" IsActive=" " ParentID="2" StreamName="BCLAN02">
      <NetworkAdapter AdapterIPAddress="10.10.13.172" ID="3" ParentID="2"><Terminal Biometrics=" " DID="1"
        Description="9700-K6 B-Client HR30" GID="1" ID="3" MACAddress="" MasterRecordIDLength="7" ParentID="3"
        Serial="false" TimeZoneID="Europe/Berlin" Type="9700-K6 B-Client HR30"/>
      </NetworkAdapter>
      <NetworkAdapter AdapterIPAddress="10.10.12.178" ID="4" ParentID="2"><Terminal Biometrics=" " DID="0"
        Description="9700-K6 Wolfgang" GID="2" ID="4" MACAddress="" MasterRecordIDLength="7" ParentID="4"
        Serial="false" TimeZoneID="Europe/Berlin" Type="9700-K6 B-Client HR30"/>
      </NetworkAdapter>
    </Stream>
  </Client>
</result returncode="200"/>
</HelperAnswer>
```

#### Values for the parameters "IsActive":

IsActive=" " ->false = Stream is not started

IsActive="X" ->true = Stream is started.

### 5.5.2.2 send\_data

#### Task

Transmission of one or several data records to the addressed terminal(s). The additional parameter "updateTemplate=X" is used to update the templates on the addressed terminal.

#### Syntax (example)

To the helper IP address:port (110.178.56.1:8089).

POST send\_data HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089

accept-encoding: gzip

sap-language: D

authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=

#### One data record:

client=EACM ZE&stream=BCLAN02&gid=01&did=01&updateTemplates&data01=A@Y00060096038000000000.00.00 0.00 0.00 00000000000 3

#### Several data records:

client=EACM ZE&stream=BCLAN02&gid=01&did=01&updateTemplates&data01=A@Y00060096038000000000.00.00 0.00 0.00 00000000000 3 &data2=.....

#### Response if the transmission was successful

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<HelperAnswer Command="send_data" NameTag="[Helper name tag]">
```

```
<result returncode="200"/>
```

```
</HelperAnswer>
```

In case of an error, the respective error code is entered in <result returncode=" " />.

### 5.5.2.3 start\_stream

**Task**

Starting the <stream> of the <client>.

**Syntax**

To the helper IP address:port (110.178.56.1:8089).

POST start\_stream HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089 accept-encoding: gzip

sap-language: D

authorization: Basic RUFDTV9CQ19VU1l6ODc2NTQzMjE= client=EACM-ZE&stream=BCLAN02

**Response if the transmission was successful**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<HelperAnswer Command="start_stream" NameTag="[Helper name tag]">
```

```
<result returncode="200"/>
```

```
</HelperAnswer>
```

In case of an error, the respective error code is entered in <result returncode=" " />.

### 5.5.2.4 stop\_stream

**Task**

Stopping the <stream> of the <client>.

**Syntax**

To the helper IP address:port (110.178.56.1:8089).

POST stop\_stream HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089 accept-encoding: gzip

sap-language: D authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE= client=EACM-ZE&stream=BCLAN02

**Response if the transmission was successful**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<HelperAnswer Command="stop_stream" NameTag="[Helper name tag]">
```

```
<result returncode="200"/>
```

```
</HelperAnswer>
```

In case of an error, the respective error code is entered in <result returncode=" " />.

### 5.5.2.5 add\_bio\_template

#### Task

Addition of a template <templateID> on the B-COMM server. The badge number must be used as template-ID. Note: a template always consist of 2 template data records (1 for preference finger and 1 for alternative finger).

The template must be provided in the PK\_COMP format.



If the badge number for an employee is changed, the template with the old badge number (template-ID) must be deleted from the B-COMM server.

For this, use the command delete\_bio\_template, see chapter [5.5.2.6](#).

Afterwards, the template with the new badge number (template-ID) must be transmitted again to the B-COMM server.

#### Syntax

To the helper IP address:port (110.178.56.1:8089).

```
POST add_bio_template HTTP/1.1
user-agent: SAP NetWeaver Application Server (1.0;740)
host: kbsn0000.mydorma.com:8089
accept-encoding: gzip
sap-language: D
authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=
templateId=0060096
&pkc1d=<Template data>
&pkc2d=<Template data>
&quality1=124
&quality2=120
&enrollDate=20190205
&enrollTime=082341
```

#### Response if the transmission was successful

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<HelperAnswer Command="add_bio_template" NameTag="[Helper name tag]">
<result returncode="200"/>
</HelperAnswer>
```

In case of an error, the respective error code is entered in <result returncode=" " />.

### 5.5.2.6 delete\_bio\_template

#### Task

Deletion of a template <templateId> from the B-COMM server.

#### Syntax

To the helper IP address:port (110.178.56.1:8089).

POST delete\_bio\_template HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089

accept-encoding: gzip

sap-language: D

authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE= templateId=0060096

#### Response if the transmission was successful

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<HelperAnswer Command="delete_bio_template" NameTag="[Helper name tag]">
```

```
<result returncode="200"/>
```

```
</HelperAnswer>
```

In case of an error, the respective error code is entered in <result returncode=""/>.

### 5.5.2.7 get\_all\_bio\_templates

#### Task

Requesting all templates. Starts the upload of all templates. For each existing template, a separate response is generated. If the number of templates is high, the upload can take several minutes depending on the computing power/network latency.

#### Syntax

To the helper IP address:port (110.178.56.1:8089).

POST get\_all\_bio\_templates HTTP/1.1

user-agent: SAP NetWeaver Application Server (1.0;740)

host: kbsn0000.mydorma.com:8089

accept-encoding:

gzip sap-language: D

authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=

#### Response if the transmission was successful:

See chapter [▶ 5.5.1.1](#).

### 5.5.2.8 send\_xml

#### Task

Forwarding of the specified helper command to the B-COMM helper and then to B-COMM. The available helper commands are described in chapter [4.4](#) and following.

The command "send\_xml" awaits an XML helper command beginning with "<?xml" as a payload.

#### Syntax

To the helper IP address:port (110.178.56.1:8089).

```
POST send_xml HTTP/1.1
user-agent: SAP NetWeaver Application Server (1.0;740)
host: kbsn0000.mydorma.com:8089
accept-encoding: gzip
sap-language: D
authorization: Basic RUFDTV9CQ19VU1I6ODc2NTQzMjE=
<?xml version="1.0"?>
<HelperCommand Command="ReadConfiguration"/>
```

#### Response

Direct response from the B-COMM helper. A return code is not transmitted.

# 6 AMQP Interface

## 6.1 Introduction

### Extract from Wikipedia entry for AMQP

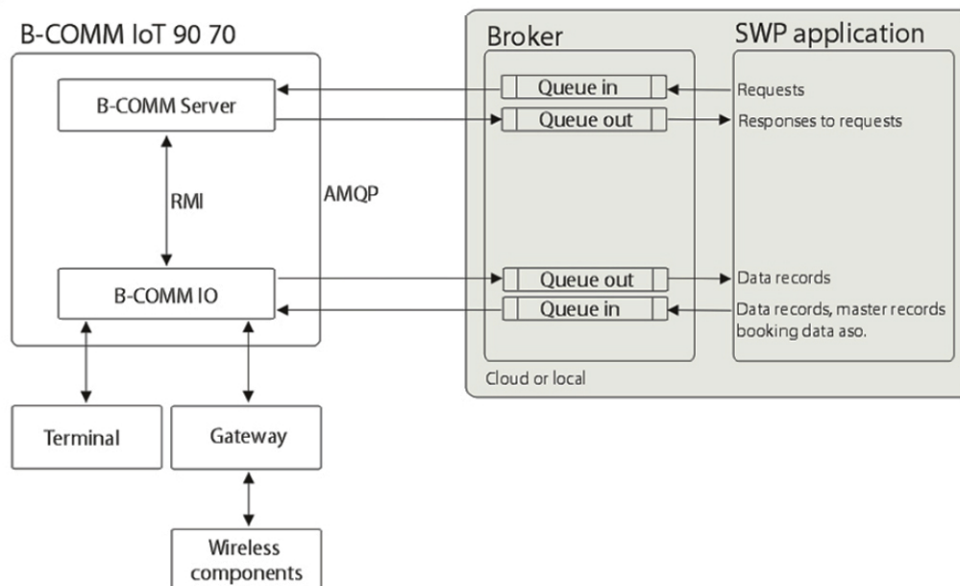
([https://en.wikipedia.org/wiki/Advanced\\_Message\\_Queueing\\_Protocol](https://en.wikipedia.org/wiki/Advanced_Message_Queueing_Protocol))

The Advanced Message Queueing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security.

AMQP is a binary, application layer protocol, designed to efficiently support a wide variety of messaging applications and communication patterns. It provides flow controlled, message-oriented communication with message-delivery guarantees such as at-most-once (where each message is delivered once or never), at-least-once (where each message is certain to be delivered, but may do so multiple times) and exactly-once (where the message will always certainly arrive and do so only once), and authentication and/or encryption based on SASL and/or TLS. It assumes an underlying reliable transport layer protocol such as Transmission Control Protocol (TCP).

The AMQP specification is defined in several layers:

- a type system
- a symmetric, asynchronous protocol for the transfer of messages from one process to another
- a standard, extensible message format
- a set of standardised but extensible 'messaging capabilities'



Generally spoken, the system consists of the following parts:

- A **producer** is a user application that sends messages.
- A **queue** is a buffer that stores messages.
- A **consumer** is a user application that receives messages.

In the B-COMM system, both B-COMM itself and the application are producers and consumers.



## 6.1.1 Broker

All following remarks are related to the RabbitMQ broker, which is used during development and is the recommended AMQP broker for B-COMM IoT 90 70.

### 6.1.1.1 Installation and initial configuration

- 1 Download Erlang from <http://www.erlang.org/downloads> (latest version 13.5.2018: 21.3) and install it.
- 2 Download RabbitMQ from <https://www.rabbitmq.com/download.html> (latest version 13.5.2018: 3.7.14) and install it.

After the installation of Erlang and RabbitMQ open a command window in the sbin subdirectory of the RabbitMQ Server installation.

Example: **C:\Program Files\RabbitMQ Server\rabbitmq\_server-3.7.14\sbin**

Enable Management plugin for later use of the Web UI of RabbitMQ by entering the following command in the command window:

**rabbitmq-plugins enable rabbitmq\_management**

The UI of RabbitMQ can then be called from a browser with URL

**localhost:15672** and a login as an administrator.

User: (usually) **guest**

Password: (usually) **guest**.

### 6.1.1.2 Global

The management UI of RabbitMQ can be called from any browser:

**localhost:15672.**

In the UI, login as administrator

User (usually): **guest**

Password (usually): **guest**.

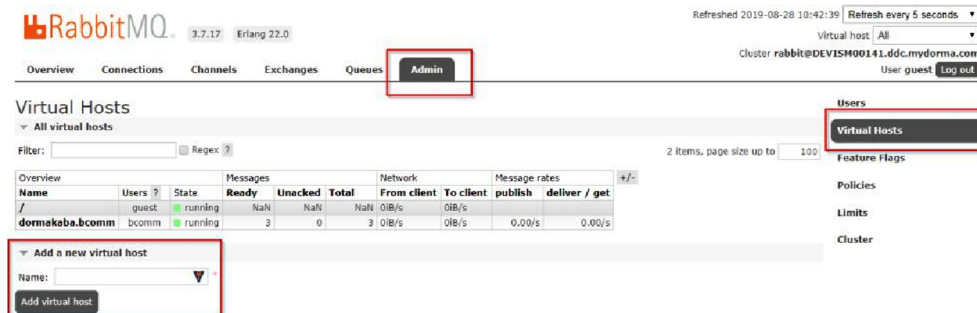
There has to be a **virtual host** and a **user** with his password configured within the broker. The virtual host and the user are used to communicate with the broker, i.e. send data to the queues or receive data from the queues.

Create a virtual host for B-COMM IoT 90 70, e.g. **dormakaba.bcomm** or use the default virtual host **/**.

A virtual host can either be created within the management UI of RabbitMQ or by entering the following command (in the same directory as mentioned above, e.g. C:\Program Files\RabbitMQ Server\rabbitmq\_server-3.7.14\sbin).

**Example: rabbitmqctl add\_vhost dormakaba.bcomm**

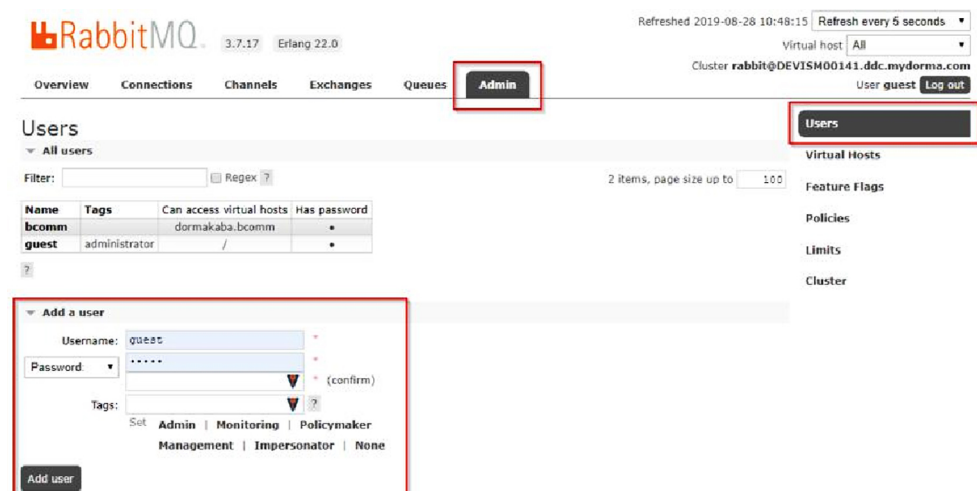
Within the management UI of RabbitMQ a virtual host can be created under the register tab **Admin**, menu item **Virtual Hosts**.



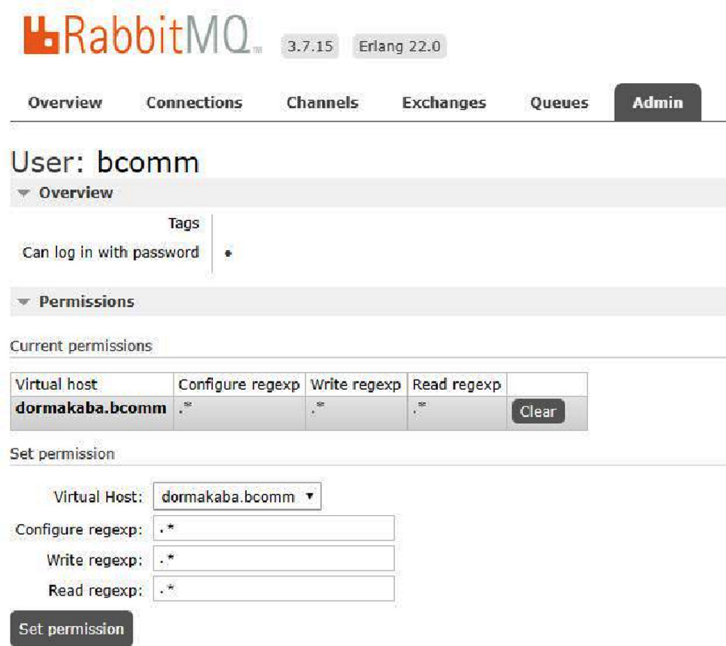
Create the user with password for B-COMM IoT 90 70 and assign access to the virtual host.

**Example: User bcomm with password bcomm with access to virtual host dormakaba.bcomm**  
**rabbitmqctl add\_user bcomm bcomm**

In the RabbitMQ management UI this can be done on register tab Admin, menu item Users:



After creating the user, access rights for the virtual host must be set for the user:



The screenshot shows the RabbitMQ Admin interface for user 'bcomm'. The 'Admin' tab is selected. Under 'Permissions', the 'Current permissions' table shows that for the virtual host 'dormakaba.bcomm', the 'Configure regexp', 'Write regexp', and 'Read regexp' are all set to '.\*'. Below this, the 'Set permission' section allows setting permissions for a specific virtual host (currently 'dormakaba.bcomm'). The 'Configure regexp', 'Write regexp', and 'Read regexp' fields are all set to '.\*'. A 'Set permission' button is at the bottom.

**User: bcomm**

Overview

Tags

Can log in with password

Permissions

Current permissions

| Virtual host    | Configure regexp | Write regexp | Read regexp |
|-----------------|------------------|--------------|-------------|
| dormakaba.bcomm | .*               | .*           | .*          |

Set permission

Virtual Host: dormakaba.bcomm

Configure regexp: .\*

Write regexp: .\*

Read regexp: .\*

Set permission

If you're not going to use TLS encryption for the communication between B-COMM/ Application and the broker, you're done now. Otherwise continue with the next chapter.

### 6.1.1.3 TLS configuration

In order to use TLS encryption with certificates, a few configuration parameters have to be set within the broker configuration file (`rabbitmq.conf`, `rabbitmq.config` or `advanced.config`).

For details about the configuration files see the RabbitMQ configuration files documentation (<https://www.rabbitmq.com/configure.html#configuration-file>).

#### Examples:

```
listeners.ssl.default = 5671

ssl_options.cacertfile = /path/to/ca_certificate_bundle.pem
ssl_options.certfile = /path/to/server_certificate.pem
ssl_options.keyfile = /path/to/server_key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = false
```

where

- **ssl\_options.cacertfile** is the Certificate Authority (CA) bundle file path
- **ssl\_options.certfile** is the server certificate file path
- **ssl\_options.keyfile** is the server private key file path
- **ssl\_options.verify** specifies whether peer verification should be enabled (**verify\_peer**) or not (**verify\_none**)
- **ssl\_options.fail\_if\_no\_peer\_cert** specifies whether a TLS connection should be rejected if the client fails to provide a certificate (**true**) or not (**false**).

or in the classic config file format

```
[
  {rabbit, [
    {ssl_listeners, [5671]},
    {ssl_options, [{cacertfile, "/path/to/ca_certificate_bundle.pem"},
                   {certfile, "/path/to/server_certificate.pem"},
                   {keyfile, "/path/to/server_key.pem"},
                   {verify, verify_peer},
                   {fail_if_no_peer_cert, false}]}
  ]}
]
```

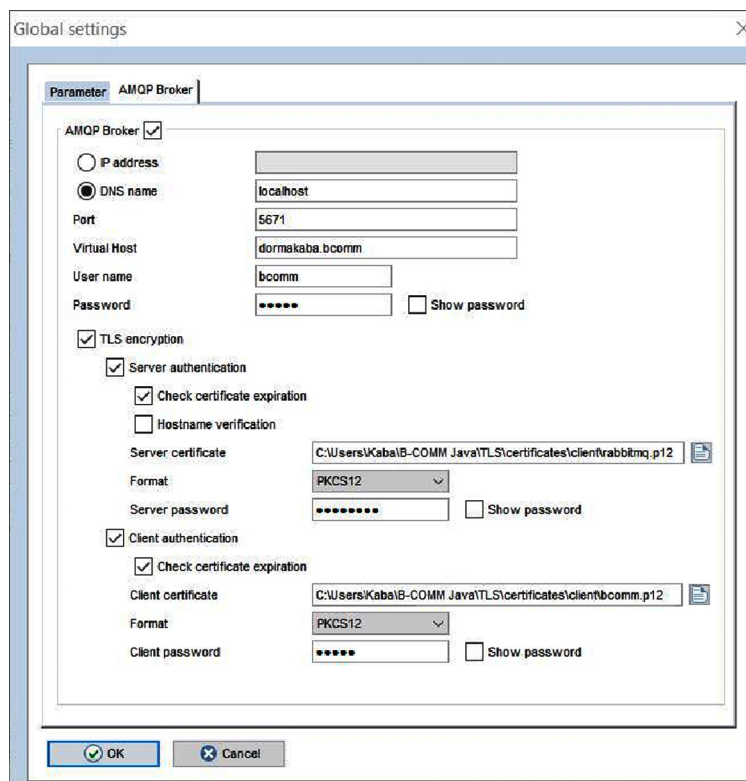
If you don't want a client verification you'll have to specify

- **ssl\_options.verify = verify\_none**
- **ssl\_options.fail\_if\_no\_peer\_cert = false**

and also disable Client verification in the AMQP Broker configuration dialog of B-COMM.

For details see the RabbitMQ TLS documentation, <https://www.rabbitmq.com/ssl.html#enabling-tls>.

Within B-COMM IoT 90 70, there's also an AMQP broker configuration page in the dialog Global settings under the main menu item Configuration, register page AMQP broker:



In this dialog you can specify the following settings:

- use AMQP Broker or not
- the IP address or DNS name of the broker
- the port number (default is 5672 without TLS and 5671 with TLS)
- the Virtual Host to use
- the user name and password
- use TLS encryption or not on server side whe TLS encryption is enabled:
- do server authentication or not
- check whether server certificate has expired or is not yet valid
- do a hostname verification or not
- specify server certificate, its format and its password on client side when TLS encryption is enabled:
- do client authentication or not
- check whether client certificate has expired or is not yet valid
- specify client certificate, its format and its password.

If TLS is enabled and one of the above checks fails when either the B-COMM Server starts or a B-COMM IO is started, no AMQP communication will take place. In such a case, a B-COMM IO will immediately shut down with an error message "Finished because of certificate error".

If a certificate expires when the connection between B-COMM and the Broker has already been established, there are no restrictions or consequences for the current connection.

#### 6.1.1.4 Exchanges

The producer sends messages to an exchange, not directly to a queue. An exchange receives messages from producers and pushes them to queues. The exchange must know exactly what to do with a message it receives, e.g.

- should it be appended to a particular queue
- should it be appended to many queues or
- should it get discarded?

The rules for that are defined automatically by the exchange type. The user must not know anything about these exchanges and its types, the whole creation of exchanges is done automatically by B-COMM.

The exchanges within the broker will have the following names:

**dormakaba.bcomm.server.<serverUUID>** for commands/responses between B-COMM and the application,

where **serverUUID** is the unique UUID of the B-COMM server. This is an exchange of type *Direct*.

and

**dormakaba.bcomm.io.<serverUUID>** for data records between B-COMM and the application,

where **serverUUID** is the unique UUID of the B-COMM server. This exchange is also of type *Direct*.

#### 6.1.1.5 Queues and routing keys

The two queues used by a B-COMM Server to receive commands and send responses to the commands will have the following names:

**dormakaba.bcomm.server.in.<serverUUID>** for the commands sent from the application to B-COMM, where serverUUID is the unique UUID of the B-COMM server.

The routing key used for sending commands therefor is also

*dormakaba.bcomm.server.in.<serverUUID>*.

and

**dormakaba.bcomm.server.out.<serverUUID>** for the responses to commands sent from B-COMM to the application, where serverUUID is the unique UUID of the B-COMM server.

The routing key used for sending responses therefor is also

*dormakaba.bcomm.server.out.<serverUUID>*.

The queues used by a B-COMM IO to send data records to an application and receive data records for a device from an application:

**dormakaba.bcomm.io.out.<serverUUID>** for data records from a device to the application, where serverUUID is the unique UUID of the B-COMM Server.

The routing key used is *dormakaba.bcomm.io.out.<serverUUID>.<client>.<stream>*, where client and stream indicate the location of the device.

and

**dormakaba.bcomm.io.in.<serverUUID>.<client>.<stream>** for data records from the application to a device in stream <stream> of client <client>, where serverUUID is the unique UUID of the B-COMM Server.

The routing key used is also *dormakaba.bcomm.io.in.<serverUUID>.<client>.<stream>*.

## 6.1.2 AMQP messages

All data between B-COMM and the application will be transmitted solely by use of JSON text messages. This has the decisive advantage, that the programming language on partner side can be another one than on side of B-COMM IoT 90 70.

JSON messages consist of a "header", "data", "error" and "response" part.

When sending a command, "header" and "data" have to be filled.

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"command"},"data":  
{"command":"..."}}
```

As a response either "header", "data" and "response" or "header", "data" and "error" will be returned.

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"commandResponse"},"data":  
{"command":"...", "response":{"..."}}
```

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"commandError"},"data":  
{"command":"...", "error":{"code":..., "description":"..."}}
```

When sending a data record from the application to B-COMM "header" and "data" have to be filled.

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"terminalMessage"},"data":  
{"record":"..."}}
```

As a response in case of an error "header", "data" and "error" will be returned.

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"deviceError"},"data":  
{"record":"...", "error":{"code":..., "description":"..."}}
```

When receiving a data record from B-COMM "header" and "data" will be transmitted.

```
{"header":{"sender":{"..."},"receiver":{"..."},"messageType":"..."},"data":{"record":"..."}}
```

## 6.2 AMQP message structure

### 6.2.1 "header"-structure

The "header" part contains control data, which consist of the source, the target and additional information about the message. The "header" structure is the same for all messages.

| JSON key              | Type      | Mandatory | Meaning                                                                                                                                                                                                              |
|-----------------------|-----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ::sender              | Structure | Yes       | Contains the source data (the sender of the message).                                                                                                                                                                |
| ::receiver            | Structure | Yes       | Contains the target data (the receiver of the message).                                                                                                                                                              |
| ::name                | String    | No        | Name of the sender/receiver, e.g.<br>- bcommio<br>- bcommserver<br>- application.                                                                                                                                    |
| ::serverUUID          | String    | Yes       | Unique UUID of the B-COMM Server, a UUID consists of a 16 byte number, which is noted hexadecimal and splitted into five groups.<br><br>In its usual format a UUID looks like: 550e8400-e29b-11d4-a716-446655440000. |
| ::client              | String    | No        | Name of a client.                                                                                                                                                                                                    |
| ::stream              | String    | No        | Name of a stream.                                                                                                                                                                                                    |
| ::device              | Structure | No        | Contains the device specific parameter if sender or receiver is a device.                                                                                                                                            |
| :::name               | String    | No        | - Name of a gateway<br>- Logical name of a component<br>- Description of a terminal                                                                                                                                  |
| :::type               | String    | No        | Type of a device, e.g.<br>- Type of a terminal, e.g. 9600-K5 B-Client HR20<br>- Type of a gateway, always Gateway 9040<br>- Type of a component, e.g. C-Lever Legic Advant.                                          |
| :::networkInformation | String    | No        | IP address/DNS name of a device (terminal, gateway, component).                                                                                                                                                      |
| :::guid               | String    | No        | GUID of a component.                                                                                                                                                                                                 |
| :::groupId            | Integer   | No        | Group address GID of a terminal.                                                                                                                                                                                     |
| :::deviceId           | Integer   | No        | Device address DID of a terminal.                                                                                                                                                                                    |



| JSON key                     | Type    | Mandatory | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|---------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| :::hasAlternativeld          | Boolean | No        | Specifies whether a terminal is addressed by an alternative ID.                                                                                                                                                                                                                                                                                                                                                                       |
| :::alternativeld             | Integer | No        | Alternative ID of a terminal.                                                                                                                                                                                                                                                                                                                                                                                                         |
| :::hasParent                 | Boolean | No        | "True", if a terminal is below a controller.                                                                                                                                                                                                                                                                                                                                                                                          |
| :::parentGroupld             | Integer | No        | Group address GID of the controller if hasParent=true.                                                                                                                                                                                                                                                                                                                                                                                |
| :::parentDeviceld            | Integer | No        | Device address DID of the controller if hasParent=true.                                                                                                                                                                                                                                                                                                                                                                               |
| :::keyplanNumber             | Integer | No        | Number of a keyplan (of a Whitelist component).                                                                                                                                                                                                                                                                                                                                                                                       |
| :::keyplanName               | String  | No        | Name of a keyplan (of a Whitelist component).                                                                                                                                                                                                                                                                                                                                                                                         |
| :::administrationArea Number | Integer | No        | Number of an administration area (of a CardLink component).                                                                                                                                                                                                                                                                                                                                                                           |
| :::administrationArea Name   | String  | No        | Name of an administration area (of a CardLink component).                                                                                                                                                                                                                                                                                                                                                                             |
| :messageType                 | String  | No        | Type of the message<br>- terminalMessage<br>- gatewayMessage<br>- componentMessage<br>- command<br>- commandResponse<br>- commandError<br>- deviceError                                                                                                                                                                                                                                                                               |
| :transactionId               | String  | Yes       | Unique id of the message, the transactionId is in the form of a UUID within B-COMM, e.g. 550e8400-e29b-11d4-a716-446655440000.<br><br>In an online response to a terminal record this transactionId should (not has to) be set to the same value as in the received message.<br><br>In a response of B-COMM to a data record or a command, the transactionId is always set to the same value as in the received data record/ command. |
| :timestamp                   | Date    | Yes       | Creation date/time of the message (in UTC format ISO 8601: yyyy-mm-ddThh:mm:ss.mmm+/-hh:mm).                                                                                                                                                                                                                                                                                                                                          |

```

"header":
{
  "sender":
  {
    "name":".....",
    "serverUUID":".....",
    "client":".....",
    "stream":".....",
    "device":
    {
      "name":".....",
      "type":".....",
      "networkInformation":".....",
      "guid":".....",
      "gid":".....",
      "did":".....",
      "hasParent":".....",
      "parentGid":".....",
      "parentDid":".....",
      "administrationArea":".....",
      "keyplan":"....."
    }
  },
  "receiver":
  {
    "name":".....",
    "serverUUID":".....",
    "client":".....",
    "stream":".....",
    "device":
    {
      "name":".....",
      "type":".....",
      "networkInformation":".....",
      "guid":".....",
      "gid":".....",
      "did":".....",
      "hasParent":".....",
      "parentGid":".....",
      "parentDid":".....",
      "administrationArea":".....",
      "keyPlan":"....."
    }
  },
  "messageType":".....",
  "transactionId":".....",
  "timestamp":"yyyy-mm-ddThh:mm:ss"
}

```

6.2.2 "data"-structure

The "data" part contains the user data of a message. Its structure depends on the message and the sender and will be described in the following sections.

```
"data":  
{  
.....  
}
```

6.2.2.1 Communication between B-COMM IO and application

This communication is the replacement for the present Application Interface (AI). Only data records - respectively requests/events - will be exchanged between devices and an application. The data records/requests/events will be transmitted as they are received from a device respectively as they are expected by a device.

| JSON key | Type   | Mandatory | Meaning                                                                                                                                                                                    |
|----------|--------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| :record  | String | Yes       | The transmitted data record/<br>request/event<br><br>- Benzing data record from/to<br>a terminal<br><br>- Request to a gateway/a<br>component<br><br>- Event from a gateway/a<br>component |

```
"data":  
{  
"record": "....."  
}
```

**Data record/event from a device to an application**

```
{
  "header":
  {
    "sender":
    {
      ...
    },
    "receiver":
    {
      "name": "application"
    },
    "transactionId": ".....",
    "timestamp": ".....",
    "messageType": "terminalMessage | componentMessage | gatewayMessage"
  },
  "data":
  {
    "record": "..."
  }
}
```

### 6.2.2.2 Response/request from application to device

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      ...
    },
    "transactionId": "...",
    "timestamp": "...",
    "messageType": "terminalMessage | componentMessage | gatewayMessage"
  },
  "data":
  {
    "record": ""
  }
}
```

### 6.2.2.3 Communication between B-COMM server and application

This communication is the replacement for the present interface via the B-COMM Helper. It's possible to send commands to a B-COMM server, read/write orders and read/write configuration data.

| JSON key | Type      | Mandatory | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| :sender  | Structure | Yes       | <p>The command to execute (see chapter no.)</p> <ul style="list-style-type: none"> <li>- getConfiguration, 6.3.3</li> <li>- getClients, 6.3.4</li> <li>- getClient, 6.3.5</li> <li>- getStreams, 6.3.6</li> <li>- getStream, 6.3.7</li> <li>- startStream, 6.3.8</li> <li>- stopStream, 6.3.9</li> <li>- getStreamStatus, 6.3.10</li> <li>- getTerminals, 6.3.11</li> <li>- getTerminal, 6.3.12</li> <li>- getKnownGateways, 6.3.13</li> <li>- getGateways, 6.3.14</li> <li>- getGateway, 6.3.15</li> <li>- setGateways, 6.3.16</li> <li>- getAdministrationAreas, 6.3.17</li> </ul> |

| JSON key | Type      | Mandatory | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------|-----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| :sender  | Structure | Yes       | <p>The command to execute, cont.</p> <ul style="list-style-type: none"> <li>- getAdministrationArea, 6.3.18</li> <li>- setAdministrationAreas, 6.3.19</li> <li>- getKeyplans, 6.3.20</li> <li>- getKeyplan, 6.3.21</li> <li>- setKeyplans, 6.3.22</li> <li>- getComponents, 6.3.23</li> <li>- getComponent, 6.3.24</li> <li>- setComponents, 6.3.25</li> <li>- getComponentEventLogFile, 6.3.26</li> <li>- getComponentEventLog, 6.3.27</li> <li>- sendRecord, 6.3.28</li> <li>- reorganizeOrders, 6.3.29</li> <li>- getOrders, 6.3.30</li> <li>- getOrder, 6.3.31</li> <li>- createOrders, 6.3.32</li> <li>- deleteOrders, 6.3.33</li> <li>- getBookings, 6.3.48</li> <li>- setWhitelist, 6.3.36</li> <li>- addWhitelist, 6.3.37</li> <li>- setWhitelistPool, 6.3.44</li> <li>- addWhitelistPool, 6.3.45</li> <li>- getWhitelist, 6.3.35</li> <li>- getWhitelistPool, 6.3.43</li> <li>- sendWhitelist, 6.3.38</li> <li>- deleteWhitelist, 6.3.39</li> <li>- sendParameters, 6.3.34</li> <li>- setDoorReleased, 6.3.40</li> <li>- setDoorLocked, 6.3.41</li> <li>- setDoorNormalOperation, 6.3.42</li> <li>- shutdownServer, 6.3.46</li> <li>- restartServer, 6.3.47</li> <li>- setMobileAccessCredentials, 6.3.49</li> <li>- deleteMobileAccessCredentials, 6.3.51</li> <li>- createProjectDevices, 6.3.52</li> <li>- deleteProjectDevices, 6.3.54</li> </ul> |

| JSON key             | Type      | Mandatory | Meaning                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| :record              | String    | No        | Data record/request to transmit with command sendRecord, 6.3.28.                                                                                                                                                                                                                                                                                                         |
| :orders              | Structure | No        | An array of orders, 6.5.15                                                                                                                                                                                                                                                                                                                                               |
| :whitelist           | Structure | No        | An array of Whitelist entries, 6.5.16                                                                                                                                                                                                                                                                                                                                    |
| :gateways            | Structure | No        | An array of gateways entries, 6.5.6                                                                                                                                                                                                                                                                                                                                      |
| :administrationAreas | Structure | No        | An array of CardLink administration areas, 6.5.8                                                                                                                                                                                                                                                                                                                         |
| :keyplans            | Structure | No        | An array of Whitelist keyplans, 6.5.9                                                                                                                                                                                                                                                                                                                                    |
| :components          | Structure | No        | An array of components, 6.5.7                                                                                                                                                                                                                                                                                                                                            |
| :parameter           | Structure | No        | Potential parameters for a command, the possible respectively necessary parameters will be described in the sections of the various commands.                                                                                                                                                                                                                            |
| ::orderStatus        | String    | No        | One order state or a semicolon (;) separated list of order states.                                                                                                                                                                                                                                                                                                       |
| ::scheduled          | Boolean   | No        | "True" for a scheduled order, otherwise "False".                                                                                                                                                                                                                                                                                                                         |
| ::days               | Integer   | No        | A number of days.                                                                                                                                                                                                                                                                                                                                                        |
| ::subElements        | Boolean   | No        | "True" if also sub elements are requested, otherwise "False". Default: "False".                                                                                                                                                                                                                                                                                          |
| ::startStream        | Boolean   | No        | <p>"True" if a stream should be started if it's not active, otherwise "False". Default: "False".</p> <p>The following commands support this parameter:</p> <p>sendParameter<br/> sendWhitelist<br/> setWhitelist<br/> addWhitelist<br/> deleteWhitelist<br/> sendRecord<br/> getComponentEventLog<br/> setDoorRelease<br/> setDoorLocked<br/> setDoorNormalOperation</p> |

**File example:**

```

"data":
{
  {
    "command": "...",
    "record": "...",
    "orders": {...},
    "parameter": {...},
  }
}

```

**6.2.3 "error"-structure**

The "error" contains the error data of a message.

```

"error":
{
  "code":...
  "description":..."
}

```

| JSON key    | Type    | Mandatory | Meaning                   |
|-------------|---------|-----------|---------------------------|
| code        | Integer | Yes       | Code of the error.        |
| description | String  | Yes       | Description of the error. |

**The following error codes are possible as an answer to a command to a B-COMM Server:**

- -1 an unknown command (command) has been sent
- -2 an unknown message type (messageType) has been sent
- -3 the transmitted UUID of the server (serverUUID) does not match with the UUID of the receiving server
- -4 no client (client) was specified
- -5 a client (client) was specified which is unknown within B-COMM
- -6 a wrong client (client) for the command was specified, e.g. a CardLink client in a command for a Whitelist client
- -7 no stream (stream) was specified
- -8 a stream (stream) was specified which is unknown within B-COMM
- -9 a wrong stream (stream) for the command was specified, e.g. a Wireless stream (BCWLS02) in a command for a Programmer stream (BCSCW01)
- -10 no device (device) was specified
- -11 a device (device) was specified which is unknown within B-COMM
- -12 no group address (groupId) was specified



- -13 no device address (deviceId) was specified
- -14 the specified device (device) could not be found in B-COMM
- -15 the specified device (device) is currently not reachable
- -16 the specified device (device) is currently flagged as "temporarily not available"
- -17 the command could not be sent to the device (device)
- -18 no alternative id (alternativeId) was specified
- -19 no IP address or no DNS name (networkInformation) was specified
- -20 no GUID (guid) was specified
- -21 no logical name (logicalName) was specified
- -22 no keyplan (keyplanNumber or keyplanName) was specified
- -23 no administration area (administrationAreaNumber or administrationAreaName) was specified
- -24 the specified keyplan (keyplanNumber or keyplanName) is unknown within B-COMM
- -25 the specified administration area (administrationAreaNumber or administrationAreaName) is unknown within B-COMM
- -26 the specified stream (stream) is not active
- -27 the order list (orders) is empty or not specified
- -28 the order requested in orders could not be found within the order list of B-COMM
- -29 the parameter list (parameter) is empty or not specified
- -30 an order specified in orders could not be written
- -31 the parameter file could not be found
- -32 no whitelist specified or specified whitelist is empty
- -33 the specified gateway list is empty
- -34 the specified keyplan list is empty
- -35 the specified administration areas list is empty
- -36 the specified component list is empty
- -37 an exception occurred
- -38 there's an error with one of the whitelist entries
- -39 there's an unexecuted order within the order list (when trying to delete the order list)
- -40 an error occurred when trying to delete the order list
- -41 an error occurred during the conversion of a file (when getting the event log file)
- -42 a timeout occurred (when getting the event log from a component via the gateway)
- -43 there's an error with one of the specified dates, either it has wrong format or start date is behind end date

**The following error codes are possible as an answer to a data record sent to B-COMM IO:**

- -1 the transmitted UUID of the server (serverUUID) does not match with the UUID of the receiving server
- -5 a client (client) was specified which is unknown within B-COMM
- -8 a stream (stream) was specified which is unknown within B-COMM
- -11 a device (device) was specified which is unknown within B-COMM
- -15 the specified device (device) is currently not reachable
- -16 the specified device (device) is currently flagged as "temporarily not available"
- -17 a communication error with a device occurred

### 6.2.4 "response"-structure

The "response" structure contains data delivered as a response to a command. Its structure depends on the command and will be described in the following sections about the various commands.

```
{  
.....  
}
```

## 6.3 Commands from application to B-COMM server

### 6.3.1 Command structure

```
{  
  "header":  
  {  
    "sender":  
    {  
      "name": "application"  
    },  
    "receiver":  
    {  
      "name": "bcommserver",  
      "serverUUID": "..."  
    },  
    "messageType": "command",  
    "transactionId": "...",  
    "timestamp": "..."  
  },  
  "data":  
  {  
    "command": "..."  
  }  
}
```

### 6.3.2 Response structure

```
{
  "header": {
    "sender": {
      "name": "bcommserver",
      "serverId": "..."
    },
    "receiver": {
      "name": "application",
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": ""
  },
  "data": {
    "command": "...",
    "response": {...}
  }
}
```

### 6.3.3 Command "getConfiguration"

Get the complete configuration of B-COMM (all clients, streams, network adapters, terminals, administration areas, keyplans, gateways, components and so on).

#### 6.3.3.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getConfiguration"
  }
}
```

### 6.3.3.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver"
      "serverUUID": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "1ec14e29-818c-41ec-89b8-181d8732a1bb",
    "timestamp": "2018-12-18T14:57:14+0200"
  },
  "data":
  {
    "command": "getConfiguration",
    "response":
    {
      "configuration":
      {
        "version": "4.0.0",
        "installation": "common",
        "applicationPath": "C:\\Kaba\\B-COMM Java",
        "dataPath": "C:\\Kaba\\B-COMM Java",
        "writeServerDebug": true,
        "serverDebugDays": 7,
        "serverDebugLevel": "INFO",
        "automaticallyReorganizeOrders": true,
        "reorganizeDays": 7,
        "broker":
        {
          "enabled": true,
          "ipAddress": "10.10.10.10",
          "dnsName": "",
          "port": 5672,
          "virtualHost": "dormakaba.bcomm"
        },
        "clients":
        [
          {
            See description in chapter "Client structure"
          },
          ...
        ]
      }
    }
  }
}

```

### 6.3.4 Command "getClients"

Get a list of all configured clients. An additional boolean parameter subElements can be specified. If subElements is true, then the sub elements of the clients - i.e. streams, terminals, administration areas, keyplans and gateways - will also be returned in the response.

#### 6.3.4.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getClients",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.4.2 Response

In the response an array of all configured clients will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getClients"
  },
  "response":
  {
    "clients":
    [
      {
        See description in chapter "Client structure"
      },
      ...
    ]
  }
}
```

### 6.3.5 Command "getClient"

Get a specified **client**. An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the **client** - i.e. streams, terminals, administration areas, keyplans and gateways - will also be returned in the response.

#### 6.3.5.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getClient"
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```



### 6.3.5.2 Response

In the response an array with one entry for the requested **client** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getClient"
  },
  "response":
  {
    "clients":
    [
      {
        See description in chapter "Client structure"
      }
    ]
  }
}
```

### 6.3.6 Command "getStreams"

Get a list of all configured **streams** for a specified **client**. An additional boolean **parameter** **subElements** can be specified. If **subElements** is true, then the sub elements of the streams - i.e. terminals, administration areas, keyplans and gateways - will also be returned in the response.

#### 6.3.6.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStreams",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.6.2 Response

In the response an array of all configured streams will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStreams"
  },
  "response":
  {
    "streams":
    [
      {
        See description in chapter "Stream structure"
      },
      ...
    ]
  }
}
```

### 6.3.7 Command "getStream"

Get a specified **stream** of a specified **client**. An additional boolean **parameter** **subElements** can be specified. If **subElements** is true, then the sub elements of the **stream** - i.e. terminals, administration areas, keyplans and gateways- will also be returned in the response.

#### 6.3.7.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStream",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.7.2 Response

In the response an array with one entry for the requested stream will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStream"
  },
  "response":
  {
    "streams":
    [
      {
        See description in chapter "Stream structure"
      }
    ]
  }
}
```

### 6.3.8 Command "startStream"

Starts the stream **stream** of the client **client**. For **stream** it's possible to specify the wildcard character \*. In case of that all streams of the specified client will be started. It's not possible to start a programmer stream BCSCW01 of a CardLink or Whitelist client or the terminal discovery stream BCTDS01 of the Terminal Discovery client.

There's no response to this command except when the stream(s) could not be started or one of the specified receiver parameters is wrong.

#### 6.3.8.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan01"
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"startStream"
  }
}
```

### 6.3.9 Command "stopStream"

Stops the stream **stream** of the client **client**. For **stream** it's possible to specify the wildcard character \*. In case of that all streams of the specified client will be stopped. It's not possible to stop a programmer stream BCSCW01 of a CardLink or Whitelist client or the terminal discovery stream BCTDS01 of the Terminal Discovery client.

There's no response to this command except when the stream(s) could not be stopped or one of the specified receiver parameters is wrong.

#### 6.3.9.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "stopStream"
  }
}
```

### 6.3.10 Command "getStreamStatus"

Get the activity state of the specified **stream** of the specified **client**.

#### 6.3.10.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStreamStatus"
  }
}
```



### 6.3.10.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverId": "10015",
      "client": "KBS",
      "stream": "bclan01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getStreamStatus"
  },
  "response":
  {
    "response": "active|inactive"
  }
}
```

### 6.3.11 Command "getTerminals"

Get a list of all configured terminals for a specified **client** and **stream**.

This command is not possible for CardLink, Whitelist/Elolegic and Terminal Discovery clients.

#### 6.3.11.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getTerminals"
  }
}
```

### 6.3.11.2 Response

In the response an array of all configured terminals will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getTerminals"
  }
  "response":
  {
    "terminals":
    [
      {
        See description in chapter "Terminal structure"
      },
      ...
    ]
  }
}
```

### 6.3.12 Command "getTerminal"

Get a specified terminal (**device**) of a specified **client** and stream. The requested terminal (device) can be specified either by its **groupid** (GID) and **deviceId** (DID) or by the **alternativeId** (alternative ID).

This command is not possible for CardLink, Whitelist/Elolegic and Terminal Discovery clients.

#### 6.3.12.1 Command request by groupId and deviceId

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan01",
      "device":
      {
        "groupid":0,
        "deviceId":0,
        "hasAlternativeId":false
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getTerminal"
  }
}
```

### 6.3.12.2 Command request by alternativeld

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "hasAlternativeld": true,
        "alternativeld": 100
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getTerminal"
  }
}
```

### 6.3.12.3 Response

In the response an array with one entry for the requested **terminal** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "groupId": 0,
        "deviceId": 0,
        "hasAlternativeId": false
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getTerminal"
  },
  "response":
  {
    "terminals":
    [
      {
        See description in chapter "Terminal structure"
      }
    ]
  }
}
```

### 6.3.13 Command "getKnownGateways"

Get a list of all gateways that have been recognized by B-COMM via UPnP, i.e. the content of the file gateways.ini.

#### 6.3.13.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getKnownGateways"
  }
}
```

### 6.3.13.2 Response

In the response an array of all known gateways with their MAC and IP addresses will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getKnownGateways"
  },
  "response":
  {
    "gateways":
    [
      {
        See description in chapter "Gateway structure"
      },
      ...
    ]
  }
}
```



### 6.3.14 Command "getGateways"

Get a list of all configured gateways for a specified **client** and **stream**. This command is only possible for CardLink/Whitelist clients and Wireless streams BCWLS02. An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the gateways - i.e. components - will also be returned in the response.

#### 6.3.14.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcwls02"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getGateways",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.14.2 Response

In the response an array of all configured gateways will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcwls02"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getGateways"
  },
  "response":
  {
    "gateways":
    [
      {
        See description in chapter "Gateway structure"
      },
      ...
    ]
  }
}
```

### 6.3.15 Command "getGateway"

Get a specified gateway (**device**) of a specified **client** and **stream**. The requested gateway (**device**) can be specified by the **networkInformation**, which is either its IP address (**ipAddress**) or its DNS name (**dnsName**).

This command is only possible for CardLink/Whitelist clients and Wireless streams.

#### 6.3.15.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"CardLink",
      "stream":"bcwls02",
      "device":
      {
        "networkInformation":"10.10.10.10"
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getGateway"
    "parameter":
    {
      "subElements":true|false
    }
  }
}
```

### 6.3.15.2 Response

In the response an array with one entry for the requested gateway will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10"
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getGateway"
  },
  "response":
  {
    "gateways":
    [
      {
        See description in chapter "Gateway structure"
      }
    ]
  }
}
```

### 6.3.16 Command "setGateways"

Set one or more gateways under a specified **stream** of a specified **client**. This command is only possible for CardLink and Whitelist clients and wireless streams BCWLS02. This command has no response.

#### 6.3.16.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcwls02"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setGateways",
    "gateways":
    [
      {
        see description in chapter "Gateway structure"
      }
    ]
  }
}
```

### 6.3.17 Command "getAdministrationAreas"

Get a list of all configured administration areas for a specified **client** and **stream**. This command is only possible for CardLink clients and Programmer streams BCSCW01. An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the administration areas - i.e. components - will also be returned in the response.

#### 6.3.17.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcscw01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getAdministrationAreas",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.17.2 Response

In the response an array of all configured administration areas will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcscw01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getAdministrationAreas"
  }
  "response":
  {
    "administrationAreas":
    [
      {
        See description in chapter "AdministrationArea structure"
      },
      ...
    ]
  }
}
```

### 6.3.18 Command "getAdministrationArea"

Get a specified administration area (**device**) of a specified **client** and **stream**. The requested administration area (device) can be specified either by its number (**administrationAreaNumber**) or its name (**administrationAreaName**). An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the administration areas - i.e. components - will also be returned in the response.

This command is only possible for CardLink clients and Programmer streams BCSCW01.

#### 6.3.18.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"CardLink",
      "stream":"bcscw01",
      "device":
      {
        "administrationAreaNumber":0
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getAdministrationArea"
    "parameter":
    {
      "subElements":true|false
    }
  }
}
```



### 6.3.18.2 Response

In the response an array with one entry for the requested **administration area** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcscw01",
      "device":
      {
        "administrationAreaNumber": 0
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "... "
  },
  "data":
  {
    "command": "getAdministrationArea"
  },
  "response":
  {
    "administrationAreas":
    [
      {
        See description in chapter "AdministrationArea structure"
      }
    ]
  }
}
```

### 6.3.19 Command "setAdministrationAreas"

Set one or more administration areas under a specified **stream** of a specified **client**. This command is only possible for CardLink clients and programmer streams BCSCW01. This command has no response.

#### 6.3.19.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcwls02"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setAdministrationAreas",
    "administrationAreas":
    [
      {
        see description in chapter "AdministrationArea structure"
      }
    ]
  }
}
```

### 6.3.20 Command "getKeyplans"

Get a list of all configured keyplans for a specified **client** and **stream**. This command is only possible for Whitelist clients and Programmer streams BCSCW01. An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the keyplans - i.e. components - will also be returned in the response.

#### 6.3.20.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getKeyplans",
    "parameter":
    {
      "subElements": true|false
    }
  }
}
```

### 6.3.20.2 Response

In the response an array of all configured keyplans will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getKeyplans"
  }
  "response":
  {
    "keyplans":
    [
      {
        See description in chapter "Keyplan structure"
      },
      ...
    ]
  }
}
```

### 6.3.21 Command "getKeyplan"

Get a specified keyplan (**device**) of a specified **client** and **stream**. The requested keyplan (**device**) can be specified either by its number (**keyplanNumber**) or its name (**keyplanName**). An additional boolean **parameter** subElements can be specified. If subElements is true, then the sub elements of the keyplans - i.e. components - will also be returned in the response.

This command is only possible for Whitelist clients and Programmer streams BCSCW01.

#### 6.3.21.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
      "stream":"bcscw01",
      "device":
      {
        "keyplanNumber":0
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getKeyplan"
    "parameter":
    {
      "subElements":true|false
    }
  }
}
```

### 6.3.21.2 Response

In the response an array with one entry for the requested keyplan will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getKeyplan"
  },
  "response":
  {
    "keyplans":
    [
      {
        See description in chapter "Keyplan structure"
      }
    ]
  }
}
```

### 6.3.22 Command "setKeyplans"

Set one or more keyplans under a specified stream of a specified client. This command is only possible for Whitelist clients and programmer streams BCSCW01. This command has no response.

#### 6.3.22.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setKeyplans",
    "keyplans":
    [
      {
        see description in chapter "Keyplan structure"
      }
    ]
  }
}
```

### 6.3.23 Command "getComponents"

Get a list of all configured components for a specified **client**, **stream** and **administration area**, **keyplan** or **gateway**. This command is only possible for CardLink and Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.23.1 Command to administration area or keyplan

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponents"
  }
}
```



### 6.3.23.2 Command to a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponents"
  }
}
```

### 6.3.23.3 Response from administration area or gateway

In the response an array of all configured components will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponents"
  }
  "response":
  {
    "components":
    [
      {
        See description in chapter "Component structure"
      },
      ...
    ]
  }
}
```

### 6.3.23.4 Response from gateway

In the response an array of all configured components will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "networkInformation": "10.10.10.10"
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponents"
  }
  "response":
  {
    "components":
    [
      {
        See description in chapter "Component structure"
      },
      ...
    ]
  }
}
```

### 6.3.24 Command "getComponent"

Get a specified component (**device**) of a specified **client**, **stream** and **administration area**, **keyplan** or **gateway**. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). This command is only possible for CardLink and Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.24.1 Command to administration area or keyplan

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0,
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponent"
  }
}
```

### 6.3.24.2 Command to a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponent"
  }
}
```

### 6.3.24.3 Response from administration area or keyplan

In the response an array with one entry for the requested **component** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0,
        "guid": "..."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponent"
  },
  "response":
  {
    "components":
    [
      {
        See description in chapter "Component structure"
      }
    ]
  }
}
```

### 6.3.24.4 Response from gateway

In the response an array with one entry for the requested component will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponent"
  },
  "response":
  {
    "components":
    [
      {
        See description in chapter "Component structure"
      }
    ]
  }
}
```

### 6.3.25 Command "setComponents"

Set one or more components under a specified administrationAera or keyplan of a specified stream of a specified client. This command is only possible for CardLink and Whitelist clients and programmer streams BCSCW01. This command has no response.

#### 6.3.25.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setComponents",
    "components":
    [
      {
        see description in chapter "Component structure"
      }
    ]
  }
}
```



### 6.3.26 Command "getComponentEventLogFile"

Get the event log (traceback) of a specified component (**device**) of a specified **client**, **stream** and **administration area**, **keyplan** or **gateway**. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). The event log will be fetched from the appropriate file in B-COMM, which means that the event log has to be loaded from the component in advance, either via the programmer or via the gateway (by an order respectively by a request to the gateway).

This command is only possible for CardLink and Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.26.1 Command to administration area or keyplan

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0,
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLogFile"
  }
}
```

### 6.3.26.2 Command to a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLogFile"
  }
}
```

### 6.3.26.3 Response from administration area or gateway

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "keyplanNumber": 0,
        "guid": "..."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLogFile"
  },
  "response":
  {
    "eventLog":
    [
      {
        See description in chapter "EventLogEntry structure"
      }
    ]
  }
}

```

### 6.3.26.4 Response from gateway

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLogFile"
  },
  "response":
  {
    "eventLog":
    [
      {
        See description in chapter "EventLogEntry structure"
      }
    ]
  }
}

```

### 6.3.27 Command "getComponentEventLog"

Get the event log (traceback) of a specified component (**device**) of a specified **client**, **stream** and **gateway**. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). The event log will be fetched from the component via the gateway, so this command can last a while.

This command is only possible for CardLink/Whitelist clients and Wireless streams BCWLS02.

#### 6.3.27.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLog"
  }
}
```

### 6.3.27.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getComponentEventLog"
  },
  "response":
  {
    "eventLog":
    [
      {
        "See description in chapter
         EventLogEntry structure"
      }
    ]
  }
}

```

### 6.3.28 Command "sendRecord"

Sends the data record **record** to a device **device** in the specified **stream** of the client **client**. This command is only possible for a terminal client and stream or a CardLink/Whitelist client and wireless stream.

If the device is a terminal in a terminal stream, it will be addressed either by its group address (GID) **groupId** and device address (DID) **deviceId** or by its alternative ID **alternativeId**. The record is a Benzing data record exactly as it will be send to the terminal.

If the device is a gateway, it will be addressed either by the IP address or the DNS name **networkInformation** of the gateway. The **record** is a gateway request as described in the KATE API documentation.

If the device is a wireless component, it will be addressed by the IP address/DNS name **networkInformation** of the corresponding gateway and the GUID **guid** of the component. The **record** is also a gateway request as described in the KATE API documentation.

There will be no response to this command except an error occurred during the transmission of the record or an error within the command.

#### 6.3.28.1 Command to a terminal

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "groupId": 0,
        "deviceId": 0,
        "hasAlternativeId": false
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendRecord",
    "record": "..."
  }
}
```

### 6.3.28.2 Command to a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "networkInformation": "10.10.10.10",
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendRecord",
    "record": "..."
  }
}
```



### 6.3.28.3 Command to a wireless component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "networkInformation": "10.10.10.10",
        "guid": "..."
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendRecord",
    "record": "..."
  }
}
```

### 6.3.29 Command "reorganizeOrders"

Reorganize the order list and file of a specified stream in the specified client. Some parameters must/can be specified:

| JSON key    | Type    | Mandatory | Meaning                                                                                                 |
|-------------|---------|-----------|---------------------------------------------------------------------------------------------------------|
| orderStatus | String  | Yes       | A semicolon ( ; ) separated list of the states of the orders to delete, e.g. FinishedOK;FinishedMessage |
| scheduled   | Boolean | No        | "True" if scheduled orders should also be reorganized, otherwise "False". Default: "False".             |
| days        | Integer | No        | Orders older than <b>days</b> days will be reorganized. Default: 7 days.                                |

There is no response to this command except an error occurred during reorganization of the order list.

### 6.3.29.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan01"
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"reorganizeOrders",
    "parameter":
    {
      "orderStatus":"...",
      "scheduled":false,
      "days":7
    }
  }
}
```

### 6.3.30 Command "getOrders"

Get all orders of the specified **stream** of the specified **client**. This command is not possible for the Terminal Discovery client or stream.

#### 6.3.30.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getOrders"
  }
}
```

### 6.3.30.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "CardLink",
      "stream": "bcscw01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getOrders"
  },
  "response":
  {
    "orders":
    [
      an array of the orders,
      see chapter "Order structure"
    ]
  }
}
```

### 6.3.31 Command "getOrder"

Get one or more orders of the stream **stream** of client **client**. The order(s) will be specified by submitting an order array with at least the section name(s) of the requested order(s). The response to this command contains an array with those requested order(s), that have been found within the internal order list of B-COMM or an empty array if the requested order(s) have not been found.

#### 6.3.31.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getOrder",
    "orders":
    [
      an array with at least one order,
      see chapter "Order structure" e.g.
      {
        "section": "ORDER_..."
      }
    ]
  }
}
```

### 6.3.31.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "10015",
      "client": "KBS",
      "stream": "bclan01"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getOrder"
  },
  "orders":
  [
    an array with the requested order(s),
    see chapter "Order structure" e.g.
    {
      "section": "ORDER_...",
      "command": "...",
      "subCommand": "...",
      ...
    }
  ]
}

```

### 6.3.32 Command "createOrders"

Create one or more orders in the order list of the specified **stream** and **client**. The **orders** to write are specified in the orders structure, where the section field must be omitted. This command is not possible for the Terminal Discovery client or stream.

#### 6.3.32.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "createOrders",
    "orders":
    [
      an array with the orders (without section name),
      see chapter "Order structure"
    ]
  }
}
```



### 6.3.32.2 Response

In the response, an array with the orders that could be created in the order list, will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan01"
    },
    "receiver":
    {
      "name":"application"
    },
    "messageType":"commandResponse",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"createOrders"
  }
  "response":
  {
    "orders":
    [
      an array with the written orders,
      see chapter "Order structure"
    ]
  }
}
```

### 6.3.33 Command "deleteOrders"

Delete **all** orders (inclusive the scheduled orders) of the specified **client** and **stream**. This command will fail if there are any unexecuted orders within the order list.

There's no response to this command except an error occurred.

#### 6.3.33.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteOrders"
  }
}
```

### 6.3.34 Command "sendParameters"

The following parameters can be send to a device:

- The assigned parameter file to a terminal, in this case the terminal must be specified by either its groupId and deviceId or its alternativeld
- The default parameter file to a standalone component (programmer PD1364 and PD1460), in this case the component must be specified by its keyplanName or keyplan-Number as well as its name or guid
- The configured parameters to a gateway, in this case the gateway must be specified by its networkInformation
- The configured parameters to a wireless component, in this case the component must be specified by the networkInformation of the corresponding gateway and its name or guid.

In each case, an appropriate order will be created in B-COMM.

There's no response to this command except when the order could not be created or one of the receiver parameters is wrong or missing.

#### 6.3.34.1 Command to a terminal

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan01",
      "device":
      {
        "groupId":0,
        "deviceId":0,
        "hasAlternativeld":false
      }
    },
    "messageType":"command",
    "transactionId":"",
    "timestamp":"..."
  },
  "data":
  {
    "command":"sendParameter"
  }
}
```

### 6.3.34.2 Command to a standalone component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bcscw01",
      "device":
      {
        "name": "CLEVER-STANDALONE",
        "keyplanNumber": 0
      }
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "...",
  },
  "data":
  {
    "command": "sendParameter"
  }
}
```

### 6.3.34.3 Command to a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bcwls02",
      "device":
      {
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendParameter"
  }
}
```

#### 6.3.34.4 Command to a wireless component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER-WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendParameter"
  }
}
```

### 6.3.35 Command "getWhitelist"

Get the whitelist of a specified component (**device**) of a specified **client**, **stream** and **keyplan** or **gateway**. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**).

This command is only possible for Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.35.1 Command to a standalone component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "name": "CLEVER-STANDALONE",
        "keyplanNumber": 0
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getWhitelist"
  }
}
```

### 6.3.35.2 Command to a wireless component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER_WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getWhitelist"
  }
}
```



### 6.3.35.3 Response

In the response an array with the whitelist entries of the requested **component** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "...",
      "stream": "...",
      "device":
      {
        ...
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getWhitelist"
  },
  "response":
  {
    "whitelist":
    [
      {
        See description in chapter
        "WhitelistEntry structure"
      }
    ]
  }
}
```

### 6.3.36 Command "setWhitelist"

Set the **whitelist** of one or more specified wireless or standalone components (**device**) of a specified **client**, **stream** and **keyplan** or **gateway**. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). It's also possible to specify the wildcard character \* for either guid or name, in this case the specified **whitelist** is set for all components of the keyplan or gateway.

**The existing whitelists in the components will be overwritten.**

This command is only possible for Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.36.1 Command to a standalone component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "name": "CLEVER-STANDALONE",
        "keyplanNumber": 0
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setWhitelist",
    "whitelist":
    [
      an array with the whitelist entries, see
      description in chapter "WhitelistEntry structure"
    ]
  }
}
```

### 6.3.36.2 Command to a wireless component

```

{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER_WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setWhitelist",
    "whitelist":
    [
      an array with the whitelist entries, see
      description in chapter "WhitelistEntry structure"
    ]
  }
}

```

### 6.3.37 Command "addWhitelist"

Add entries to the **whitelists** of one or more specified wireless or standalone components (**device**) of a specified **client**, **stream** and **keyplan** or gateway. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). It's also possible to specify the wildcard character \* for either **guid** or **name**, in this case the specified **whitelist** is set for all components of the keyplan or gateway. The new entries will be added to the existing whitelists in the components without checking for duplicates.

This command is only possible for Whitelist clients and Programmer or Wireless streams BCSCW01 and BCWLS02.

#### 6.3.37.1 Command to a standalone component

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcscw01",
      "device":
      {
        "name": "CLEVER-STANDALONE",
        "keyplanNumber": 0
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "addWhitelist",
    "whitelist":
    [
      an array with the whitelist entries, see
      description in chapter "WhitelistEntry structure"
    ]
  }
}
```

### 6.3.37.2 Command to a wireless component

```

{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER_WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "addWhitelist",
    "whitelist":
    [
      an array with the whitelist entries, see
      description in chapter "WhitelistEntry structure"
    ]
  }
}

```

### 6.3.38 Command "sendWhitelist"

Send the whitelist to a wireless component by creating an order for the appropriate component. The component must be specified by the **networkInformation** of the corresponding gateway and its logical name (**name**) or GUID (**guid**). It's also possible to specify the wildcard character \* for either **guid** or **name**, in this case the specified **whitelists** of all components of the gateway will be sent.

This command is only possible for Whitelist clients and Wireless streams BCWLS02.

There's no response to this command except when the order could not be created or one of the receiver parameters is wrong or missing.

#### 6.3.38.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER-WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "sendWhitelist"
  }
}
```

### 6.3.39 Command "deleteWhitelist"

Delete either the complete whitelist or specified whitelist entries of a wireless Whitelist component or the complete whitelist of a standalone component within B-COMM. For a wireless component also an order for the appropriate component will be created. A wireless component must be specified by the **networkInformation** of the corresponding gateway and its logical name (**name**) or GUID (**guid**), a standalone component by the number (**keyplanNumber**) or name (**keyplanName**) of the corresponding keyplan and its logical name (**name**) or GUID (**guid**). It's also possible to specify the wildcard character \* for either **guid** or **name**, in this case the **whitelists** of all components of the keyplan or gateway will be deleted.

This command is only possible for Whitelist clients and wireless or programmer streams.

There's no response to this command except when the order could not be created or one of the receiver parameters is wrong or missing.

#### 6.3.39.1 Command delete the complete whitelist

For both wireless and standalone components.

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
      "stream":"bcwls02",
      "device":
      {
        "name":"CLEVER-WIRELESS",
        "networkInformation":"10.10.10.10"
      }
    },
    "messageType":"command",
    "transactionId":"",
    "timestamp":"..."
  },
  "data":
  {
    "command":"deleteWhitelist"
  }
}
```

### 6.3.39.2 Command delete single whitelist entry

For wireless components only.

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
      "stream": "bcwls02",
      "device":
      {
        "name": "CLEVER-WIRELESS",
        "networkInformation": "10.10.10.10"
      }
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "... "
  },
  "data":
  {
    "command": "deleteWhitelist"
    "whitelist":
    [
      an array with the whitelist entries, see
      description in chapter "WhitelistEntry structure"
    ]
  }
}
```



### 6.3.40 Command "setDoorReleased"

Set one or more doors of a specified **client**, **stream** and **gateway** to release state. The requested gateway (**networkInformation**) must be specified either by its IP address or its DNS name. It's also possible to specify the wildcard character \* for the **networkInformation**, then all gateways are affected by the command. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). It's also possible to specify the wildcard character \* for either **guid** or name, in this case all components connected to the specified gateway(s) are affected by the command.

This command is only possible for Whitelist or CardLink clients and Wireless streams BCWLS02.

There's no response to this command except when the stream(s) could not be started or one of the specified receiver parameters is wrong.

This command doesn't create orders but the requests are sent directly to the gateway(s), just like with the command **sendRecord**.

#### 6.3.40.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
      "stream":"bcwls02",
      "device":
      {
        "name":"CLEVER_WIRELESS",
        "networkInformation":"10.10.10.10"
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"setDoorReleased"
  }
}
```

### 6.3.41 Command "setDoorLocked"

Set one or more doors of a specified **client**, **stream** and **gateway** to locked state. The requested gateway (**networkInformation**) must be specified either by its IP address or its DNS name. It's also possible to specify the wildcard character \* for the **networkInformation**, then all gateways are affected by the command. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). It's also possible to specify the wildcard character \* for either guid or **name**, in this case all components connected to the specified gateway(s) are affected by the command.

This command is only possible for Whitelist or CardLink clients and Wireless streams BCWLS02.

There's no response to this command except when the stream(s) could not be started or one of the specified receiver parameters is wrong.

This command doesn't create orders but the requests are sent directly to the gateway(s), just like with the command **sendRecord**.

#### 6.3.41.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
      "stream":"bcwls02",
      "device":
      {
        "name":"CLEVER_WIRELESS",
        "networkInformation":"10.10.10.10"
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"setDoorLocked"
  }
}
```

### 6.3.42 Command "setDoorNormalOperation"

Set one or more doors of a specified **client**, **stream** and **gateway** to normal operation mode. The requested gateway (**networkInformation**) must be specified either by its IP address or its DNS name. It's also possible to specify the wildcard character \* for the **networkInformation**, then all gateways are affected by the command. The requested component (**device**) must be specified either by its GUID (**guid**) or its logical name (**name**). It's also possible to specify the wildcard character \* for either **guid** or **name**, in this case all components connected to the specified gateway(s) are affected by the command.

This command is only possible for Whitelist or CardLink clients and Wireless streams BCWLS02.

There's no response to this command except when the stream(s) could not be started or one of the specified receiver parameters is wrong.

This command doesn't create orders but the requests are sent directly to the gateway(s), just like with the command **sendRecord**.

#### 6.3.42.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
      "stream":"bcwls02",
      "device":
      {
        "name":"CLEVER_WIRELESS",
        "networkInformation":"10.10.10.10"
      }
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"setDoorNormalOperation"
  }
}
```

### 6.3.43 Command "getWhitelistPool"

Get the whitelist pool of a specified **client**. This command is only possible for Whitelist clients.

#### 6.3.43.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"Whitelist",
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getWhitelistPool"
  }
}
```

### 6.3.43.2 Response

In the response an array with the whitelist entries of the requested **client** will be returned.

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "Whitelist",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getWhitelistPool"
  },
  "response":
  {
    "whitelist":
    [
      {
        See description in chapter "WhitelistEntry structure"
      }
    ]
  }
}
```

### 6.3.44 Command "setWhitelistPool"

Set the whitelist pool of a specified client.

**The existing whitelist pool of the client will be overwritten.**

This command is only possible for Whitelist clients.

There's no response to this command except when one of the specified receiver parameters is wrong.

#### 6.3.44.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setWhitelistPool",
    "whitelist":
    [
      an array with the whitelist entries,
      see description in chapter "WhitelistEntry structure"
    ]
  }
}
```

### 6.3.45 Command "addWhitelistPool"

Add new entries to the whitelist pool of a specified **client**.

**The new entries will be added to the existing whitelist pool of the client without checking for duplicates.**

This command is only possible for whitelist clients.

There's no response to this command except if one of the specified receiver parameters is wrong.

#### 6.3.45.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan01"
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "addWhitelistPool",
    "whitelist":
    [
      an array with the whitelist entries,
      see description in chapter "WhitelistEntry structure"
    ]
  }
}
```

### 6.3.46 Command "shutdownServer"

Shuts the B-COMM Server down.

There's no response to this command.

#### 6.3.46.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"..."
    },
    "messageType":"command",
    "transactionId":"...",
    "timestamp":"..."
  },
  "data":
  {
    "command":"shutdownServer"
  }
}
```



### 6.3.47 Command "restartServer"

Restarts the B-COMM Server, i.e. all RMI objects will be re-registered and the complete configuration will be refreshed.

There's no response to this command.

#### 6.3.47.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "...",
    "timestamp": "..."
  },
  "data":
  {
    "command": "restartServer"
  }
}
```

### 6.3.48 Command "getBookings"

Get the bookings of the specified **client** and **stream** for the period specified by the parameters **startDate** and **endDate** (both either in format **YYYY-MM-DD** or **YYMMDD**).

For this command to work properly, the client must be configured to write multiple booking files (client configuration dialog, advanced settings, multiple booking files), see the relating chapter in the B-COMM manual.

#### 6.3.48.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name":"application"
    },
    "receiver":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"KBS",
      "stream":"bclan02"
    },
    "messageType":"command",
    "transactionId":"",
    "timestamp":"..."
  },
  "data":
  {
    "command":"getBookings",
    "parameter":
    {
      "startDate":"YYYY-MM-DD",
      "endDate":"YYYY-MM-DD"
    }
  }
}
```

### 6.3.48.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "KBS",
      "stream": "bclan02"
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "getBookings"
  },
  "response":
  {
    "bookings":
    [
      an array of the bookings as strings
    ]
  }
}
```

### 6.3.49 Command "setMobileAccessCredentials"

Send static credentials to LEGIC Connect, which then will be distributed to the appropriate mobile phones.

The response contains those credentials, that have been successfully transmitted to LEGIC Connect.

#### 6.3.49.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setMobileAccessCredentials",
    "staticCredentials":
    [
      an array with the static credentials,
      see description in chapter "StaticCredential structure"
    ]
  }
}
```

### 6.3.49.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "setMobileAccessCredentials"
  },
  "response":
  {
    "staticCredentials":
    [
      an array of the static credentials which have been successfully
      transmitted to LEGIC Connect,
      see description in chapter "StaticCredential structure"
    ]
  }
}

```

### 6.3.50 Command "writeMobileAccessCredentials"

This command is the same as setMobileAccessCredentials, i.e. send static credentials to LEGIC Connect, which then will be deployed to the appropriate mobile phones. The response contains those credentials, that have been successfully transmitted to LEGIC Connect. This command is available up from B-COMM version 5.0.0.

The specification of a client is optional. If it is omitted or set to "", the credentials will be deployed for the default project of BCOMM, else the credentials will be deployed for the project assigned to the specified client in B-COMM.

#### 6.3.50.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "writeMobileAccessCredentials",
    "staticCredentials":
    [
      an array with the static credentials,
      see chapter StaticCredential structure
    ]
  }
}
```

### 6.3.50.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "writeMobileAccessCredentials"
  },
  "response":
  {
    "staticCredentials":
    [
      an array of the static credentials which have been
      successfully transmitted to LEGIC Connect,
      see chapter StaticCredential structure
    ]
  }
}

```

### 6.3.51 Command "deleteMobileAccessCredentials"

Send static credentials to LEGIC Connect, which then will be deleted from the appropriate mobile phones.

The response contains those credentials, that have been successfully transmitted to LEGIC Connect.

#### 6.3.51.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteMobileAccessCredentials",
    "staticCredentials":
    [
      an array with the static credentials,
      see description in chapter "StaticCredential structure"
    ]
  }
}
```



### 6.3.51.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteMobileAccessCredentials"
  },
  "response":
  {
    "staticCredentials":
    [
      an array of the static credentials which have been
      successfully transmitted to LEGIC Connect,
      see description in chapter "StaticCredential structure"
    ]
  }
}
```

### 6.3.52 Command "createProjectDevices"

Create project devices (mobile phones) in LEGIC Connect to which VCPs or Static Credentials can be deployed.

#### 6.3.52.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "createProjectDevices",
    "projectDevices":
    [
      an array with the devices to create in LEGIC Connect,
      see description in chapter "ProjectDevice Structure"
    ]
  }
}
```

### 6.3.52.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "...",
  },
  "data":
  {
    "command": "createProjectDevices"
  },
  "response":
  {
    "projectDevices":
    [
      an array with the devices which have been successfully created in LEGIC Connect,
      see description in chapter "ProjectDevice Structure"
    ]
  }
}
```

### 6.3.53 Command "listProjectDevices"

List project devices (mobile phones) created in LEGIC Connect. This command is available up from B-COMM version 4.1.0.

#### 6.3.53.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listProjectDevices"
  }
}
```

### 6.3.53.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listProjectDevices"
  },
  "response":
  {
    "projectDevices":
    [
      an array with the devices which have been read from LEGIC Connect,
      see description in chapter "ProjectDevice Structure"
    ]
  }
}
```

### 6.3.54 Command "deleteProjectDevices"

Remove project devices (mobile phones) from LEGIC Connect.

#### 6.3.54.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteProjectDevices",
    "projectDevices":
    [
      an array with the devices to remove from LEGIC Connect,
      see description in chapter "ProjectDevice Structure"
    ]
  }
}
```

### 6.3.54.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteProjectDevices"
  },
  "response":
  {
    "projectDevices":
    [
      an array with the devices which have been
      successfully removed from LEGIC Connect,
      see description in chapter "ProjectDevice Structure"
    ]
  }
}

```

### 6.3.55 Command "listProvisionedFiles"

List all provisioned files (credentials and VCPs) available for registered apps for all or just the specified mobile phones. This command is available up from B-COMM version 5.0.0.

The specification of a client is optional. If it is omitted or set to "\*", the provisioned files of the default project of B-COMM will be listed, else the provisioned files of the project assigned to the specified client in B-COMM will be listed.

#### 6.3.55.1 Command

##### 6.3.55.1.1 List all provisioned files for all smartphones

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listProvisionedFiles"
  }
}
```



### 6.3.55.1.2 List all provisioned files for specified smartphones

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listProvisionedFiles",
    "provisionedFiles":
    [
      {
        "phoneNo": "..."
      },
      ...
    ]
  }
}
```

### 6.3.55.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listProvisionedFiles"
  },
  "response":
  {
    "registeredMobileApp":
    [
      {
        "publicRegistrationId": "...",
        "mobileAppId": "...",
        "provisionedFiles":
        [
          {
            "fileId": [...],
            "isVcp": "...",
            "modificationDate": "...",
            "fileState": "...",
            "rfiInterfaces": [...]
          },
          ...
        ]
      },
      ...
    ]
  }
}

```

### 6.3.56 Command "listVcps"

List VCPs (Versatile Configuration Package) created in LEGIC Connect. This command is available up from B-COMM version 5.0.0.

The specification of a client is optional. If it is omitted or set to "\*", the VCPs of the default project of B-COMM will be listed, else the VCPs of the project assigned to the specified client in B-COMM will be listed.

#### 6.3.56.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listVcps"
  }
}
```

### 6.3.56.2 Response

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "listVcps"
  },
  "response":
  {
    "vcps":
    [
      {
        "name": "...",
        "readerSeries4000": "...",
        "readerSeries6000": "...
      },
      ...
    ]
  }
}

```

### 6.3.57 Command "deleteVcps"

Remove all VCPs (Versatile Configuration Package) from specified mobile phones. This command is available up from BCOMM version 5.0.0.

The specification of a client is optional. If it is omitted or set to "\*", the VCPs of the default project of B-COMM will be removed, else the VCPs of the project assigned to the specified client in B-COMM will be removed.

#### 6.3.57.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "deleteVcps",
    "vcps":
    [
      {
        "phoneNo": "..."
      },
      ...
    ]
  }
}
```

### 6.3.57.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name":"bcommserver",
      "serverUUID":"...",
      "client":"..."
    },
    "receiver":
    {
      "name":"application"
    },
    "messageType":"commandResponse",
    "transactionId":"",
    "timestamp":"..."
  },
  "data":
  {
    "command":"deleteVcps"
  },
  "response":
  {
    "vcps":
    [
      {
        "phoneNo":"...",
        "requestId":"..."
      },
      ...
    ]
  }
}
```

### 6.3.58 Command "writeVcps"

Deploy all VCPs (Versatile Configuration Package) to specified mobile phones. This command is available up from B-COMM version 5.0.0.

The specification of a client is optional. If it is omitted or set to "\*", the VCPs of the default project of B-COMM will be deployed, else the VCPs of the project assigned to the specified client in B-COMM will be deployed.

#### 6.3.58.1 Command

```
{
  "header":
  {
    "sender":
    {
      "name": "application"
    },
    "receiver":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "messageType": "command",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "writeVcps",
    "vcps":
    [
      {
        "phoneNo": "..."
      },
      ...
    ]
  }
}
```

### 6.3.58.2 Response

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommserver",
      "serverUUID": "...",
      "client": "..."
    },
    "receiver":
    {
      "name": "application"
    },
    "messageType": "commandResponse",
    "transactionId": "",
    "timestamp": "..."
  },
  "data":
  {
    "command": "writeVcps"
  },
  "response":
  {
    "vcps":
    [
      {
        "phoneNo": "...",
        "requestId": "..."
      },
      ...
    ]
  }
}
```



## 6.4 Examples

### 6.4.1 Data record/event from device to application

#### 6.4.1.1 From a terminal

```
{
  "header":
  {
    "sender":
    {
      "name":"bcommio",
      "serverUUID":".....",
      "client":"KBS",
      "stream":"bclan01",
      "device":
      {
        "type":"9600-K5 B-Client HR20",
        "name":"Haupteingang",
        "groupId":0,
        "deviceId":0,
        "hasAlternativId":false
      }
    },
    "receiver":
    {
      "name":"application"
    },
    "transactionId":".....",
    "timestamp":".....",
    "messageType":"terminalMessage"
  },
  "data":
  {
    "record": "... "
  }
}
```

### 6.4.1.2 From a gateway

```
{
  "header":
  {
    "sender":
    {
      "name": "bcommio",
      "serverUUID": ".....",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "name": "GW1",
        "type": "Gateway 9040",
        "networkInformation": "10.10.10.10"
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "transactionId": ".....",
    "timestamp": ".....",
    "messageType": "gatewayMessage"
  },
  "data":
  {
    "record": "..."
  }
}
```

### 6.4.1.3 From a component

```

{
  "header":
  {
    "sender":
    {
      "name": "bcommio",
      "serverUUID": ".....",
      "client": "KBS",
      "stream": "bclan01",
      "device":
      {
        "name": "C-LEVER-1",
        "type": "C-Lever Legic Advant Wireless E321 V4",
        "networkInformation": "10.10.10.10",
        "guid": "...."
      }
    },
    "receiver":
    {
      "name": "application"
    },
    "transactionId": ".....",
    "timestamp": ".....",
    "messageType": "componentMessage"
  },
  "data":
  {
    "record": "..."
  }
}

```

## 6.5 Description of B-COMM structures

### 6.5.1 Client structure

| JSON key                            | Type      | Mandatory | Meaning                                                                                                                                                                    |
|-------------------------------------|-----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>                         | String    | Yes       | Name of the client                                                                                                                                                         |
| <b>type</b>                         | String    | Yes       | Type of the client <ul style="list-style-type: none"> <li>• TERMINAL</li> <li>• CARDLINK</li> <li>• WHITELIST</li> <li>• ELOLEGIC</li> <li>• TERMINAL_DISCOVERY</li> </ul> |
| <b>paths</b>                        | Structure | No        | The paths for the various files within B-COMM; not in a Terminal Discovery client.                                                                                         |
| :pathForOrderFiles                  | String    | Yes       | Path for order files.                                                                                                                                                      |
| :pathForParameterFiles              | String    | Yes       | Path for parameter files.                                                                                                                                                  |
| :pathForDebugFiles                  | String    | Yes       | Path for debug files.                                                                                                                                                      |
| :pathForBookingFiles                | String    | Yes       | Path for booking files; not in an ELOLEGIC client.                                                                                                                         |
| :pathForMirrorFiles                 | String    | Yes       | Path for mirror files; not in an ELOLEGIC client.                                                                                                                          |
| :pathForMasterRecordFiles           | String    | Yes       | Path for master record files; not in an ELOLEGIC client.                                                                                                                   |
| :pathForAvisoFiles                  | String    | No        | Path for AVISO files; only in a TERMINAL client.                                                                                                                           |
| :pathForValidationUpdateRecordFiles | String    | No        | Path for LEGIC validation and update record files; only in a TERMINAL client.                                                                                              |
| :pathForTemplateFiles               | String    | No        | Path for biometric templates; only in a TERMINAL client.                                                                                                                   |
| :pathForTransparentDataFiles        | String    | No        | Path for transparent data files; only in a TERMINAL client.                                                                                                                |
| <b>alternativeIdEnabled</b>         | Boolean   | No        | "True" if alternative id is used for terminal addressing, otherwise "False"; only in a TERMINAL client.                                                                    |
| <b>sshFunctionsEnabled</b>          | Boolean   | No        | "True" if SSH functions are enabled, otherwise "False"; only in a TERMINAL client.                                                                                         |
| <b>bookingFileAnsiEnabled</b>       | Boolean   | Yes       | "True" if the booking files are ANSI encoded, otherwise "False"; not in an ELOLEGIC or TERMINAL_DISCOVERY client.                                                          |
| <b>adjustProgrammerClockEnabled</b> | Boolean   | No        | "True" if programmer clock will be adjusted automatically; only for ELOLEGIC, WHITELIST and CARDLINK client.                                                               |

| JSON key                     | Type      | Mandatory | Meaning                                                                                                                                             |
|------------------------------|-----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>rmiPort</b>               | Integer   | No        | RMI port for the programmer stream BCSCW01, only in a CARDLINK, WHITELIST or ELOLEGIC client.                                                       |
| <b>applicationInterfaces</b> | Structure | No        | List of application interfaces, if multiple application interfaces are selected; only in a TERMINAL client; 6.5.5 - ApplicationInterface structure. |
| <b>ariosConfiguration</b>    | String    | No        | The Arios SCC if there is one for the client.                                                                                                       |
| <b>siteKeys</b>              | Structure | No        | An array of Arios sitekeys if there are sitekeys for the client.                                                                                    |
| :index                       | Integer   | Yes       | Index of the sitekey (0 to 7).                                                                                                                      |
| :siteKey                     | String    | Yes       | The sitekey itself..                                                                                                                                |
| :fingerPrint                 | String    | Yes       | Fingerprint of the sitekey.                                                                                                                         |
| <b>streams</b>               | Structure | No        | An array of streams, 6.5.2 - Stream structure.                                                                                                      |

## 6.5.2 Stream structure

| JSON key                                    | Type      | Mandatory | Meaning                                                                                                                                                                                                |
|---------------------------------------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>                                 | String    | Yes       | Name of the stream.                                                                                                                                                                                    |
| <b>client</b>                               | String    | Yes       | Name of the client the stream belongs to.                                                                                                                                                              |
| <b>type</b>                                 | String    | Yes       | Type of the stream: <ul style="list-style-type: none"> <li>• ACCESSMANAGER</li> <li>• BIOMETRICS</li> <li>• TERMINAL</li> <li>• SMARTCOM</li> <li>• WIRELESS</li> <li>• TERMINAL_DISCOVERY.</li> </ul> |
| <b>subType</b>                              | String    | No        | Subtype of the stream; only if <b>type</b> is TERMINAL: <ul style="list-style-type: none"> <li>• NETWORK</li> <li>• SERIAL.</li> </ul>                                                                 |
| <b>description</b>                          | String    | No        | Description of the stream.                                                                                                                                                                             |
| <b>active</b>                               | Boolean   | No        | "True" if stream is active, otherwise "False".                                                                                                                                                         |
| <b>autoStartEnabled</b>                     | Boolean   | No        | "True" if stream is automatically started by the B-COMM server, otherwise "False".                                                                                                                     |
| <b>applicationInterfaceEnabled</b>          | Boolean   | Yes       | "True" if stream has at least one application interface, otherwise "False".                                                                                                                            |
| <b>multipleApplicationInterfacesEnabled</b> | Boolean   | No        | "True" if stream has multiple application interfaces enabled, otherwise "False".                                                                                                                       |
| <b>applicationInterfaceAMQPEnabled</b>      | Boolean   | No        | "True" if the AMQP interface is enabled, otherwise "False".                                                                                                                                            |
| <b>applicationInterfaces</b>                | Structure | No        | List of application interfaces, if multiple application interfaces are selected; only in a TERMINAL stream; 6.5.5 - ApplicationInterface structure.                                                    |
| <b>avisolInterfaceIPAddress</b>             | String    | No        | IP address to use for the AVISO interface.                                                                                                                                                             |
| <b>avisolInterfaceDNSName</b>               | String    | No        | DNS name to use for the AVISO interface.                                                                                                                                                               |
| <b>avisolInterfacePort</b>                  | Integer   | No        | Port to use for the AVISO interface.                                                                                                                                                                   |
| <b>ipAddress</b>                            | String    | No        | Host IP address.                                                                                                                                                                                       |
| <b>dnsName</b>                              | String    | No        | Host DNS name.                                                                                                                                                                                         |
| <b>port</b>                                 | Integer   | No        | UDP port for communication.                                                                                                                                                                            |
| <b>bookingFileEnabled</b>                   | Boolean   | No        | "True" if the writing of a booking file is enabled, otherwise "False".                                                                                                                                 |
| <b>blackList</b>                            | String    | No        | A semicolon ( ; ) separated list of record types that are not bookings.                                                                                                                                |
| <b>debugFileEnabled</b>                     | Boolean   | No        | "True" if the writing of debug files is enabled, otherwise "False".                                                                                                                                    |

| JSON key                                        | Type      | Mandatory | Meaning                                                                                                                  |
|-------------------------------------------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------|
| <b>mirrorFileEnabled</b>                        | Boolean   | No        | "True" if the writing of mirror files is enabled, otherwise "False".                                                     |
| <b>encryptionEnabled</b>                        | Boolean   | No        | "True" if the communication between the stream and its terminals is encrypted or not; only for TERMINAL NETWORK streams. |
| <b>encryptionExchangeInterval</b>               | Integer   | No        | Interval in days in which the encryption key will be changed.                                                            |
| <b>timeSynchronizationEnabled</b>               | Boolean   | No        | "True" if the time synchronization of devices is enabled, otherwise "False".                                             |
| <b>daylightSavingTimeSynchronizationEnabled</b> | Boolean   | No        | "True" if the time synchronization of devices is enabled, otherwise "False".                                             |
| <b>hasModem</b>                                 | Boolean   | No        | "True" if stream has a modem, otherwise "False"; only for TERMINAL SERIAL streams.                                       |
| <b>modemType</b>                                | String    | No        | Type of the modem if hasModem=true.                                                                                      |
| <b>terminals</b>                                | Structure | No        | Array of connected terminals; not in SMARTCOM, WIRELESS and TERMINAL_DISCOVERY streams; 6.5.4 - Terminal structure.      |
| <b>administrationAreas</b>                      | Structure | No        | Array of administration areas; only in SMARTCOM streams; 6.5.8 - AdministrationArea structure.                           |
| <b>keyplans</b>                                 | Structure | No        | Array of keyplans; only in SMARTCOM streams; 6.5.9 - Keyplan structure.                                                  |
| <b>gateways</b>                                 | Structure | No        | Array of gateway; only in WIRELESS streams; 6.5.6 - Gateway structure.                                                   |

### 6.5.3 NetworkAdapter structure

| JSON key                         | Type    | Mandatory | Meaning                                                                   |
|----------------------------------|---------|-----------|---------------------------------------------------------------------------|
| <b>groupId</b>                   | Integer | Yes       | Group address (GID) of the connected terminal(s).                         |
| <b>deviceIdFrom</b>              | Integer | Yes       | Lower limit of the device address(es) (DID) of the connected terminal(s). |
| <b>deviceIdTo</b>                | Integer | Yes       | Upper limit of the device address(es) (DID) of the connected terminal(s). |
| <b>ipAddress</b>                 | String  | No        | IP address of the network adapter.                                        |
| <b>dnsName</b>                   | String  | No        | DNS name of the network adapter.                                          |
| <b>port</b>                      | Integer | Yes       | Port number for UDP communication.                                        |
| <b>type</b>                      | String  | Yes       | Type of the network adapter.                                              |
| <b>subterminalSupportEnabled</b> | Boolean | Yes       | "True" if subterminals are supported, otherwise "False".                  |
| <b>subterminalRange</b>          | Integer | No        | Number of supported subterminals.                                         |

### 6.5.4 Terminal structure

| JSON key                    | Type    | Mandatory | Meaning                                                                                                                        |
|-----------------------------|---------|-----------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>groupid</b>              | Integer | No        | Group address (GID) of the terminal (0 to 29).                                                                                 |
| <b>deviceld</b>             | Integer | No        | Device address (DID) of the terminal (0 to 60).                                                                                |
| <b>alternativeld</b>        | Integer | No        | Alternative ID of the terminal (-1 if not used or 0 to 9999).                                                                  |
| <b>temporarilyDisabled</b>  | Boolean | No        | "True" if the terminal is temporarily disabled within B-COMM.                                                                  |
| <b>type</b>                 | String  | No        | Type of the terminal, e.g. "9600-K5 B-Client HR20"                                                                             |
| <b>operatingSystem</b>      | String  | No        | Operating system of the terminal: <ul style="list-style-type: none"> <li>Android</li> <li>Jbed.</li> </ul>                     |
| <b>description</b>          | String  | No        | Optional description of the terminal.                                                                                          |
| <b>timeZone</b>             | String  | Yes       | Time zone of the terminal, e.g. "Europe/Berlin".                                                                               |
| <b>characterSet</b>         | String  | Yes       | Character set of the terminal, e.g. "ISO-8859-1: Latin-1, Western European".                                                   |
| <b>programNumber</b>        | String  | No        | Program number of the terminal.                                                                                                |
| <b>operatingMode</b>        | String  | Yes       | Operating mode of the terminal: <ul style="list-style-type: none"> <li>Online</li> <li>Offline</li> <li>Autonomous.</li> </ul> |
| <b>masterRecordIdLength</b> | Integer | Yes       | Length of the ID in a master record.                                                                                           |
| <b>pinLength</b>            | Integer | No        | Length of the PIN.                                                                                                             |
| <b>parameterFile</b>        | String  | No        | Complete file path of a parameter file if a parameter file is assigned to this terminal.                                       |
| <b>reader1</b>              | String  | No        | Type of reader 1.                                                                                                              |
| <b>reader2</b>              | String  | No        | Type of reader 2.                                                                                                              |
| <b>reader3</b>              | String  | No        | Type of reader 3.                                                                                                              |
| <b>arios</b>                | Boolean | Yes       | "True" if terminal is an ARIOS terminal (MIFARE), otherwise "False".                                                           |
| <b>biometricsEnabled</b>    | Boolean | Yes       | "True" if terminal has a biometric reader and should appear in biometric stream BCFCT01, otherwise "False".                    |
| <b>biometricsCBMEnabled</b> | Boolean | Yes       | "True" if biometric reader is a CBM module, otherwise "False".                                                                 |
| <b>ftpEnabled</b>           | Boolean | Yes       | "True" if terminal is capable of FTP transfers, otherwise "False".                                                             |
| <b>sftpEnabled</b>          | Boolean | Yes       | "True" if terminal is capable of SFTP transfers, otherwise "False".                                                            |
| <b>subterminal</b>          | Boolean | Yes       | "True" if terminal is a subterminal, i.e. is located below a controller, otherwise "False".                                    |



| JSON key                 | Type      | Mandatory | Meaning                                                                                                 |
|--------------------------|-----------|-----------|---------------------------------------------------------------------------------------------------------|
| <b>hasSubterminals</b>   | Boolean   | Yes       | "True" if terminal has subterminals connected, i.e. terminal itself is a controller, otherwise "False". |
| <b>subterminals</b>      | Structure | No        | Array of subterminals.                                                                                  |
| <b>hasParent</b>         | Boolean   | Yes       | Alternative ID of the terminal (-1 if not used or 0 to 9999).                                           |
| <b>parentGroupId</b>     | Integer   | No        | Group address (GID) of the parent terminal.                                                             |
| <b>parentDeviceId</b>    | Integer   | No        | Device address (DID) of the parent terminal.                                                            |
| <b>hasNetworkAdapter</b> | Boolean   | Yes       | "True" if terminal has a network adapter, otherwise "False".                                            |
| <b>networkAdapter</b>    | Structure | No        | Network adapter this terminal belongs to, 6.5.3 - NetworkAdapter structure.                             |
| <b>hasModem</b>          | Boolean   | No        | "True" if terminal has a modem, otherwise "False".                                                      |
| <b>modemCallNumber</b>   | String    | No        | The call number of the modem if hasModem=true                                                           |

### 6.5.5 ApplicationInterface structure

| JSON key                     | Type    | Mandatory | Meaning                                                                                                                      |
|------------------------------|---------|-----------|------------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>                  | String  | No        | Name of the application interface.                                                                                           |
| <b>ipAddress</b>             | String  | Yes       | IP address of the application interface.                                                                                     |
| <b>dnsName</b>               | String  | Yes       | DNS name of the application interface.                                                                                       |
| <b>port</b>                  | Integer | Yes       | Port number to use.                                                                                                          |
| <b>virtualHost</b>           | String  | No        | Virtual host if application interface uses AMQP.                                                                             |
| <b>timeout</b>               | Integer | No        | Timeout in seconds.                                                                                                          |
| <b>connectionInterval</b>    | Integer | No        | Interval in seconds, in which a re-connection will be tried.                                                                 |
| <b>connectionRepetitions</b> | Integer | No        | Number of re-connection tries.                                                                                               |
| <b>connectionDelay</b>       | Integer | No        | Connection delay in seconds.                                                                                                 |
| <b>recordTypes</b>           | String  | No        | A semicolon ( ; ) separated list of record types/ record type modifications if multiple application interfaces are selected. |

### 6.5.6 Gateway structure

| JSON key                         | Type      | Mandatory | Meaning                                                                                                                   |
|----------------------------------|-----------|-----------|---------------------------------------------------------------------------------------------------------------------------|
| <b>name</b>                      | String    | No        | Name of the gateway.                                                                                                      |
| <b>type</b>                      | String    | No        | Type of the gateway, currently fix "Gateway 90 40".                                                                       |
| <b>index</b>                     | Integer   | Yes       | Index of the gateway within B-COMM.                                                                                       |
| <b>timeZone</b>                  | String    | Yes       | Time zone of the gateway, e.g. Europe/Berlin.                                                                             |
| <b>requestTimeout</b>            | Integer   | No        | Request timeout in seconds.                                                                                               |
| <b>commissioningOpenTime</b>     | Integer   | No        | Time span in seconds for which the ZigBee network remains open for commissioning before it's automatically closed.        |
| <b>linkQualityEventsEnabled</b>  | Boolean   | No        | "True" if LinkQuality events will be sent from the gateway to B-COMM, otherwise "False".                                  |
| <b>firmwareVersion</b>           | String    | No        | Firmware version of the gateway.                                                                                          |
| <b>osVersion</b>                 | String    | No        | OS version of the gateway.                                                                                                |
| <b>rootFileSystemVersion</b>     | String    | No        | Root file system version of the gateway.                                                                                  |
| <b>zigBeeVersion</b>             | String    | No        | ZigBee version of the gateway.                                                                                            |
| <b>zigBeeChannel</b>             | Integer   | No        | ZigBee channel used by the gateway for communication with the components.                                                 |
| <b>hardwareVersion</b>           | String    | No        | Hardware version of the gateway                                                                                           |
| <b>componentFirmwareVersions</b> | String    | No        | A semicolon ( ; ) separated list of component firmware versions for components which have been uploaded into the gateway. |
| <b>componentCount</b>            | Integer   | No        | Number of components joined to this gateway.                                                                              |
| <b>routerCount</b>               | Integer   | No        | Number of routers joined to this gateway.                                                                                 |
| <b>available</b>                 | Boolean   | Yes       | "True" if the gateway is available via the socketIO communication interface, otherwise "False".                           |
| <b>network</b>                   | Structure | No        | Some network parameters.                                                                                                  |
| :ipAddress                       | String    | No        | IP address of the gateway.                                                                                                |
| :dnsName                         | String    | No        | DNS name of the gateway.                                                                                                  |
| :macAddress                      | String    | Yes       | MAC address of the gateway.                                                                                               |
| :subnet                          | Integer   | No        | Subnet of the gateway (CIDR index).                                                                                       |
| :defaultGateway                  | String    | No        | IP address of the standard gateway.                                                                                       |
| :primaryDns                      | String    | No        | IP address of the primary DNS server of the gateway.                                                                      |
| :secondaryDns                    | String    | No        | IP address of the secondary DNS server of the gateway.                                                                    |
| :dhcpEnabled                     | Boolean   | No        | "True" if DHCP is enabled, otherwise "False".                                                                             |

| JSON key            | Type      | Mandatory | Meaning                                                                                                                                                                                  |
|---------------------|-----------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>kateServer</b>   | Structure | No        | Some settings of the KATE server.                                                                                                                                                        |
| :active             | Boolean   | No        | "True" if KATE server is active, otherwise "False".                                                                                                                                      |
| :port               | Integer   | No        | Port number used by the KATE server.                                                                                                                                                     |
| <b>webServer</b>    | Structure | No        | Some settings of the Web server.                                                                                                                                                         |
| :active             | Boolean   | No        | "True" if Web server is active, otherwise "False".                                                                                                                                       |
| :port               | Integer   | No        | Port number used by the Web server.                                                                                                                                                      |
| <b>networkState</b> | String    | No        | State of the ZigBee network: <ul style="list-style-type: none"> <li>• openWithHomeAutomationSecurity</li> <li>• openWithInstallationCode</li> <li>• closed</li> <li>• unknown</li> </ul> |
| <b>components</b>   | Structure | No        | Array with the components joined to this gateway, 6.5.7 - Component structure description.                                                                                               |

### 6.5.7 Component structure

| JSON key                        | Type    | Mandatory | Meaning                                                                                                                                                                        |
|---------------------------------|---------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>keyplanNumber</b>            | Integer | No        | Number of the keyplan the component belongs to, only for Whitelist components.                                                                                                 |
| <b>keyplanName</b>              | String  | No        | Name of the keyplan the component belongs to, only for Whitelist components.                                                                                                   |
| <b>administrationAreaNumber</b> | Integer | No        | Number of the administration area the component belongs to, only for CardLink components.                                                                                      |
| <b>administrationAreaName</b>   | String  | No        | Name of the administration area the component belongs to, only for CardLink components.                                                                                        |
| <b>mode</b>                     | String  | No        | Operating mode of the component: <ul style="list-style-type: none"> <li>• CardLink</li> <li>• CardLink Wireless</li> <li>• Whitelist</li> <li>• Whitelist Wireless.</li> </ul> |
| <b>type</b>                     | String  | Yes       | Component type, e.g. C-Lever Legic Advant evolo V4.                                                                                                                            |
| <b>deviceTypeid</b>             | Integer | Yes       | Device type ID, see file actuators.ini of B-COMM.                                                                                                                              |

| JSON key                     | Type    | Mandatory | Meaning                                                                                                                                                                                                                  |
|------------------------------|---------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>logicalName</b>           | String  | Yes       | Logical name of the component.                                                                                                                                                                                           |
| <b>mediaTechnology</b>       | String  | Yes       | Media technology of the component: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• LegicPrime</li> <li>• Mifare</li> </ul>                                                                               |
| <b>version</b>               |         | No        | Version of the component, either 3 or 4.                                                                                                                                                                                 |
| <b>guid</b>                  | String  | Yes       | Unique GUID of the component.                                                                                                                                                                                            |
| <b>configurationState</b>    | String  | No        | Configuration state of the component: <ul style="list-style-type: none"> <li>• changed</li> <li>• downloadedToGateway</li> <li>• downloadedToProgrammer</li> <li>• normalOperation.</li> </ul>                           |
| <b>whitelistState</b>        | String  | No        | Whitelist state of the component, only for wireless components: <ul style="list-style-type: none"> <li>• changed</li> <li>• downloadedToGateway</li> <li>• downloadedToProgrammer</li> <li>• normalOperation.</li> </ul> |
| <b>timeZone</b>              | String  | Yes       | Time zone of the component, e.g. Europe/Berlin.                                                                                                                                                                          |
| <b>buzzerEnabled</b>         | Boolean | Yes       | "True" if the buzzer is enabled, otherwise "False".                                                                                                                                                                      |
| <b>lightRingEnabled</b>      | Boolean | Yes       | "True" if the light ring is enabled, otherwise "False".                                                                                                                                                                  |
| <b>tracebackEnabled</b>      | Boolean | Yes       | "True" if the component internal traceback is enabled, otherwise "False".                                                                                                                                                |
| <b>mediaTracebackEnabled</b> | Boolean | Yes       | "True" if the media traceback is enabled, otherwise "False". Not for elolectic V3 components.                                                                                                                            |
| <b>tracebackFullPercent</b>  | Integer | Yes       | Specifies for which percentage a traceback full indicator will be written to the ActuatorStatus/ CardLink segment.<br>0 = no event.                                                                                      |
| <b>boltOpenTime</b>          | Integer | Yes       | Bolt open duration in seconds.                                                                                                                                                                                           |
| <b>boltRecreationTime</b>    | Integer | No        | Bolt recreation time in minutes, only possible for C- Lever components.<br>0 = function off.                                                                                                                             |
| <b>reporting</b>             | String  | No        | Specifies the type of reporting, only for wireless components: <ul style="list-style-type: none"> <li>• All</li> <li>• AllowedAccess</li> <li>• MediaRelatedEvents</li> <li>• None.</li> </ul>                           |

| JSON key                      | Type      | Mandatory | Meaning                                                                                                                                                                                                      |
|-------------------------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>identificationDuration</b> | Integer   | No        | Duration for the identification request in seconds, only for wireless components.                                                                                                                            |
| <b>segmentSearchString</b>    | String    | No        | Segment search string for LEGIC prime components.                                                                                                                                                            |
| <b>segmentOffset</b>          | Integer   | No        | Segment offset for LEGIC prime components.                                                                                                                                                                   |
| <b>supportedTechnologies</b>  | Structure | No        | Array with the supported technologies.                                                                                                                                                                       |
| :name                         | String    | Yes       | Name of the technology: <ul style="list-style-type: none"> <li>• ISO_14443_A</li> <li>• LEGIC_PRIME</li> <li>• ISO_15693</li> </ul>                                                                          |
| :priority                     | Integer   | Yes       | Priority of the technology (0 to 2).                                                                                                                                                                         |
| <b>stamps</b>                 | Structure | No        | Default stamps for the components, only for LEGIC advant components, 6.5.10 - Stamp structure.                                                                                                               |
| <b>desfireCardLinkFileIds</b> | Structure | No        | Array with the MIFARE Desfire CardLink file IDs.                                                                                                                                                             |
| :siteKeyIndex                 | Integer   | Yes       | Index of the sitekey (0 to 7).                                                                                                                                                                               |
| :desfireCardLinkFileId        | Integer   | Yes       | Corresponding file ID of the CardLink segment.                                                                                                                                                               |
| <b>validationMode</b>         | String    | Yes       | Validation modes for the components, only for CardLink components: <ul style="list-style-type: none"> <li>• ACTUATOR</li> <li>• MEDIA</li> <li>• ACTUATOR_AND_MEDIA</li> <li>• ACTUATOR_OR_MEDIA.</li> </ul> |
| <b>cidMode</b>                | String    | Yes       | Default CID mode for the components: <ul style="list-style-type: none"> <li>• CID</li> <li>• UID_PLAIN</li> <li>• UID_SECURE</li> <li>• UID_AND_CID.</li> </ul>                                              |
| <b>masters</b>                | Structure | No        | Array of Master A and Master B, 6.5.11 - Master structure.                                                                                                                                                   |
| :uid                          | String    | Yes       | UID of the master.                                                                                                                                                                                           |
| :name                         | String    | No        | Optional name for the master.                                                                                                                                                                                |
| :type                         | String    | Yes       | Type of the master, i.e. MasterA or MasterB.                                                                                                                                                                 |
| :mediaTechnology              | String    | Yes       | Media technology of the master: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• LegicPrime</li> <li>• Mifare.</li> </ul>                                                                     |

| JSON key              | Type      | Mandatory | Meaning                                                                                                                     |
|-----------------------|-----------|-----------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>timePro</b>        | Structure | No        | TimePro timeprofile assigned to this component, 6.5.14 - TimeProfile structure.                                             |
| <b>holidays</b>       | Structure | No        | Array with holidays.                                                                                                        |
| :startDate            | String    | Yes       | Start date of the holiday, format YYYY-MM-DD.                                                                               |
| :endDate              | String    | Yes       | End date of the holiday, format YYYY-MM-DD.                                                                                 |
| <b>specialDays</b>    | Structure | No        | Array with special days, only for V3 components.                                                                            |
| :date                 | String    | Yes       | Date of the special day, format YYYY-MM-DD.                                                                                 |
| <b>specialDays1</b>   | Structure | No        | Array with special days 1, only for V4 components.                                                                          |
| :date                 | String    | Yes       | Date of the special day 1, format YYYY-MM-DD.                                                                               |
| <b>specialDays2</b>   | Structure | No        | Array with special days 2, only for V4 components.                                                                          |
| :date                 | String    | Yes       | Date of the special day 2, format YYYY-MM-DD.                                                                               |
| <b>oif</b>            | Structure | No        | A structure with the OiF (Object in Field) parameters, only for LEGIC advant V4 components.                                 |
| :mode                 | String    | Yes       | Mode for OiF: <ul style="list-style-type: none"> <li>fix</li> <li>variable.</li> </ul>                                      |
| :interval             | Integer   | Yes       | Interval in milliseconds if mode is fix or the starting interval if mode is variable.                                       |
| :restingPeriod        | Integer   | No        | Resting period in milliseconds if mode is variable.                                                                         |
| <b>doorNumber</b>     | Integer   | No        | Door number, only for CardLink components.                                                                                  |
| <b>doorGroups</b>     | Structure | No        | Array with door groups, 6.5.12 - DoorGroup structure.                                                                       |
| <b>blacklist</b>      | Structure | No        | Blacklist entries, only for CardLink components, 6.5.18 - BlacklistEntry structure.                                         |
| <b>cabinetLock</b>    | Structure | No        | Some CabinetLock specific parameters.                                                                                       |
| :doorNumber           | Integer   | Yes       | Cabinet lock number.                                                                                                        |
| :lockMode             | String    | Yes       | Lock mode of the cabinet lock: <ul style="list-style-type: none"> <li>WithoutMedia</li> <li>WithAuthorizedMedia.</li> </ul> |
| :freeSelectionEnabled | Boolean   | Yes       | "True" if FreeSelection mode is enabled, otherwise "False".                                                                 |
| :maintenanceMediaMode | String    | Yes       | Mode of the maintenance media: <ul style="list-style-type: none"> <li>OpenAndClose</li> <li>Open.</li> </ul>                |
| :maintenanceMedia     | Structure | No        | Array with maintenance media, 6.5.13 - MaintenanceMediaEntry structure.                                                     |

| JSON key                    | Type      | Mandatory | Meaning                                                                                                                                                                                                                     |
|-----------------------------|-----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>firmwareVersion</b>      | String    | No        | Firmware version of the component.                                                                                                                                                                                          |
| <b>zigBeeNetworkAddress</b> | String    | No        | ZigBee network address, only for Wireless components.                                                                                                                                                                       |
| <b>eui64Address</b>         | String    | No        | EUI64 address, only for Wireless components.                                                                                                                                                                                |
| <b>zigBeeVersion</b>        | String    | No        | ZigBee version, only for Wireless components.                                                                                                                                                                               |
| <b>hardwareVersion</b>      | String    | No        | Hardware version, only for Wireless components.                                                                                                                                                                             |
| <b>serialNumber</b>         | Long      | No        | Serial number of the component.                                                                                                                                                                                             |
| <b>batteryPercentage</b>    | String    | No        | Last known battery percentage level of the component: <ul style="list-style-type: none"> <li>percentage for battery types != CR2, e.g. "good (70%)" or "low (8%)"</li> <li>"low" or "good" for battery type CR2.</li> </ul> |
| <b>batteryQuantity</b>      | Integer   | No        | Number of batteries ( $\geq 0$ ).                                                                                                                                                                                           |
| <b>batteryType</b>          | String    | No        | Quantity and type of the battery, e.g. "AA" or "CR2" or "no battery".                                                                                                                                                       |
| <b>linkQuality</b>          | Integer   | No        | Last known link quality between component and gateway, only for Wireless components.                                                                                                                                        |
| <b>gateway</b>              | Structure | No        | Gateway to which this component is connected to, only for Wireless components, 6.5.6 - Gateway structure.                                                                                                                   |
| <b>available</b>            | Boolean   | No        | "True" if component is available for the gateway, otherwise "False". For Wireless components only.                                                                                                                          |
| <b>joinInfo</b>             | Structure | No        | Information about the join state of a wireless component.                                                                                                                                                                   |
| :gatewayIndex               | Integer   | Yes       | Index of the gateway the component should be joined to.                                                                                                                                                                     |
| :gatewayName                | String    | Yes       | Name of the gateway the component should be joined to.                                                                                                                                                                      |
| :gatewayIPAddress           | String    | Yes       | IP address of the gateway the component is joined.                                                                                                                                                                          |
| :gatewayJoinName            | String    | Yes       | Name of the gateway the component is joined to.                                                                                                                                                                             |
| :gatewayJoinIPAddress       | String    | Yes       | IP address of the gateway the component is joined to.                                                                                                                                                                       |
| :joinState                  | String    | Yes       | Join status: <ul style="list-style-type: none"> <li>disjoined</li> <li>initialized</li> <li>associated</li> <li>joined</li> <li>joinedToWrongGateway.</li> </ul>                                                            |

| JSON key                   | Type      | Mandatory | Meaning                                                                                                                                                                                                             |
|----------------------------|-----------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>doorSettings</b>        | Structure | No        | Some settings of the door, only for wireless components.                                                                                                                                                            |
| :doorCommandsEnabled       | Boolean   | Yes       | "True" if door commands should be enabled, otherwise "False".                                                                                                                                                       |
| :lockModeReportingEnabled  | Boolean   | Yes       | "True" if lock mode reporting is enabled, otherwise "False".                                                                                                                                                        |
| :frameContactSetting       | String    | No        | Frame contact setting if the door is equipped with a frame contact: <ul style="list-style-type: none"> <li>DISABLED</li> <li>CONTACT_IDLE_OPEN</li> <li>CONTACT_IDLE_CLOSE</li> </ul>                               |
| :handleContactSetting      | String    | No        | Handle contact setting if the door is equipped with a handle contact: <ul style="list-style-type: none"> <li>DISABLED</li> <li>CONTACT_IDLE_OPEN</li> <li>CONTACT_IDLE_CLOSE</li> </ul>                             |
| :boltContactSetting        | String    | No        | Bolt contact setting if the door is equipped with a bolt contact: <ul style="list-style-type: none"> <li>DISABLED</li> <li>CONTACT_IDLE_OPEN</li> <li>CONTACT_IDLE_CLOSE</li> </ul>                                 |
| :cylinderContactSetting    | String    | No        | Cylinder contact setting if the door is equipped with a cylinder contact: <ul style="list-style-type: none"> <li>DISABLED</li> <li>CONTACT_IDLE_OPEN</li> <li>CONTACT_IDLE_CLOSE</li> </ul>                         |
| :doorForcedMessagesEnabled | Boolean   | No        | "True" if door forced messages are enabled, otherwise "False". Can only be true if all contact settings are not DISABLED.                                                                                           |
| :frameMessagesEnabled      | Boolean   | No        | "True" if frame messages are enabled, otherwise "False". Can only be true if frameContactSetting is not DISABLED.                                                                                                   |
| :handleMessagesEnabled     | Boolean   | No        | "True" if handle messages are enabled, otherwise "False". Can only be true if handleContactSetting is not DISABLED.                                                                                                 |
| :boltMessagesEnabled       | Boolean   | No        | "True" if bolt messages are enabled, otherwise "False". Can only be true if boltContactSetting is not DISABLED.                                                                                                     |
| :cylinderMessagesEnabled   | Boolean   | No        | "True" if cylinder messages are enabled, otherwise "False". Can only be true if cylinderContactSetting is not DISABLED.                                                                                             |
| <b>whitelist</b>           | Structure | No        | The Whitelist will not be included in this structure, because it can consist of up to 4000 entries, and therefore the responses will be too large, but there's the command getWhitelist to get it from a component. |



## 6.5.8 AdministrationArea structure

| JSON key                      | Type      | Mandatory | Meaning                                                                                                                                                                        |
|-------------------------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>number</b>                 | Integer   | Yes       | Number of the administration area.                                                                                                                                             |
| <b>name</b>                   | String    | No        | Name of the administration area.                                                                                                                                               |
| <b>timeZone</b>               | String    | No        | Default time zone for the components, e. g. Europe/Berlin.                                                                                                                     |
| <b>mediaTechnology</b>        | String    | Yes       | Media technology of the components: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• LegicPrime</li> <li>• Mifare.</li> </ul>                                   |
| <b>segmentSearchString</b>    | String    | No        | Segment search string for LEGIC prime components.                                                                                                                              |
| <b>segmentOffset</b>          | Integer   | No        | Segment offset for LEGIC prime components.                                                                                                                                     |
| <b>desfireCardLinkFileIds</b> | Structure | No        | Array with the MIFARE Desfire CardLink file IDs.                                                                                                                               |
| :siteKeyIndex                 | Integer   | Yes       | Index of the sitekey (0 to 7).                                                                                                                                                 |
| :desfireCardLinkFileId        | Integer   | Yes       | Corresponding file ID of the CardLink segment.                                                                                                                                 |
| <b>supportedTechnologies</b>  | Structure | No        | Array with the supported technologies.                                                                                                                                         |
| :name                         | String    | Yes       | Name of the technology: <ul style="list-style-type: none"> <li>• ISO_14443_A</li> <li>• LEGIC_PRIME</li> <li>• ISO_15693.</li> </ul>                                           |
| :priority                     | Integer   | Yes       | Priority of the technology (0 to 2).                                                                                                                                           |
| <b>stamps</b>                 | Structure | No        | Default stamps for the components, 6.5.10 - Stamp structure.                                                                                                                   |
| <b>validationDurations</b>    | Structure | No        | Validation durations for the components.                                                                                                                                       |
| :index                        | Integer   | Yes       | Index of the validation duration (2 to 7).                                                                                                                                     |
| :duration                     | String    | Yes       | The duration itself. If index is 2, it's an hour (e.g. "18"), if index is 3 to 7 it's a duration (P<x>DT<y>H, where <x> are days and <y> are hours.                            |
| <b>validationMode</b>         | String    | Yes       | Validation modes for the components: <ul style="list-style-type: none"> <li>• ACTUATOR</li> <li>• MEDIA</li> <li>• ACTUATOR_AND_MEDIA</li> <li>• ACTUATOR_OR_MEDIA.</li> </ul> |
| <b>cidMode</b>                | String    | Yes       | Default CID mode for the components: <ul style="list-style-type: none"> <li>• CID</li> <li>• UID_PLAIN</li> <li>• UID_SECURE</li> <li>• UID_AND_CID.</li> </ul>                |

| JSON key            | Type      | Mandatory | Meaning                                                                                                                                  |
|---------------------|-----------|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>masters</b>      | Structure | No        | Array of Master A and Master B, 6.5.11 - Master structure.                                                                               |
| :uid                | String    | Yes       | UID of the master.                                                                                                                       |
| :name               | String    | No        | Optional name for the master.                                                                                                            |
| :type               | String    | Yes       | Type of the master, i.e. MasterA or MasterB.                                                                                             |
| :mediaTechnology    | String    | Yes       | Media technology of the master: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• LegicPrime</li> <li>• Mifare.</li> </ul> |
| <b>timeProfiles</b> | Structure | No        | Array with time profiles, 6.5.14 - TimeProfile structure.                                                                                |
| <b>timePros</b>     | Structure | No        | Array with TimePro time profiles, 6.5.14 - TimeProfile structure.                                                                        |
| <b>doorGroups</b>   | Structure | No        | Array with door groups, 6.5.12 - DoorGroup structure.                                                                                    |
| <b>holidays</b>     | Structure | No        | Array with holidays.                                                                                                                     |
| :startDate          | String    | Yes       | Start date of the holiday, format YYYY-MM-DD.                                                                                            |
| :endDate            | String    | Yes       | End date of the holiday, format YYYY-MM-DD.                                                                                              |
| <b>specialDays</b>  | Structure | No        | Array with special days, only for V3 components.                                                                                         |
| :date               | String    | Yes       | Date of the special day, format YYYY-MM-DD.                                                                                              |
| <b>specialDays1</b> | Structure | No        | Array with special days 1, only for V4 components.                                                                                       |
| :date               | String    | Yes       | Date of the special day 1, format YYYY-MM-DD.                                                                                            |
| <b>specialDays2</b> | Structure | No        | Array with special days 2, only for V4 components.                                                                                       |
| :date               | String    | Yes       | Date of the special day 2, format YYYY-MM-DD.                                                                                            |
| <b>oif</b>          | Structure | No        | A structure with the OiF (Object in Field) parameters, only for LEGIC advant V4 components.                                              |
| :mode               | String    | Yes       | Mode for OiF: <ul style="list-style-type: none"> <li>• fix</li> <li>• variable.</li> </ul>                                               |
| :interval           | Integer   | Yes       | Interval in milliseconds if mode is fix or the starting interval if mode is variable.                                                    |
| :restingPeriod      | Integer   | No        | Resting period in milliseconds if mode is variable.                                                                                      |
| <b>components</b>   | Structure | No        | Array with the components that belong to this administration area, 6.5.7 - Component structure.                                          |

## 6.5.9 Keyplan structure

| JSON key                     | Type      | Mandatory | Meaning                                                                                                                                                 |
|------------------------------|-----------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>number</b>                | Integer   | Yes       | Number of the keyplan.                                                                                                                                  |
| <b>name</b>                  | String    | No        | Name of the keyplan.                                                                                                                                    |
| <b>timeZone</b>              | String    | No        | Default time zone for the components, e. g. Europe/Berlin.                                                                                              |
| <b>supportedTechnologies</b> | Structure | No        | Array with the supported technologies.                                                                                                                  |
| :name                        | String    | Yes       | Name of the technology: <ul style="list-style-type: none"> <li>ISO_14443_A</li> <li>LEGIC_PRIME</li> <li>ISO_15693.</li> </ul>                          |
| :priority                    | Integer   | Yes       | Priority of the technology (0 to 2).                                                                                                                    |
| <b>stamps</b>                | Structure | No        | Default stamps for the components, 6.5.10 - Stamp structure.                                                                                            |
| <b>cidMode</b>               | String    | Yes       | Default CID mode for the components: <ul style="list-style-type: none"> <li>CID</li> <li>UID_PLAIN</li> <li>UID_SECURE</li> <li>UID_AND_CID.</li> </ul> |
| <b>masters</b>               | Structure | No        | Array of Master A and Master B, 6.5.11 - Master structure.                                                                                              |
| :uid                         | String    | Yes       | UID of the master.                                                                                                                                      |
| :name                        | String    | No        | Optional name for the master.                                                                                                                           |
| :type                        | String    | Yes       | Type of the master, i.e. MasterA or MasterB.                                                                                                            |
| :mediaTechnology             | String    | Yes       | Media technology of the master: <ul style="list-style-type: none"> <li>LegicAdvant</li> <li>LegicPrime</li> <li>Mifare.</li> </ul>                      |
| <b>timeProfiles</b>          | Structure | No        | Array with time profiles, 6.5.14 - TimeProfile structure.                                                                                               |
| <b>timePros</b>              | Structure | No        | Array with TimePro time profiles, 6.5.14 - TimeProfile structure.                                                                                       |
| <b>holidays</b>              | Structure | No        | Array with holidays.                                                                                                                                    |
| :startDate                   | String    | Yes       | Start date of the holiday, format YYYY-MM-DD.                                                                                                           |
| :endDate                     | String    | Yes       | End date of the holiday, format YYYY-MM-DD.                                                                                                             |

| JSON key            | Type      | Mandatory | Meaning                                                                                     |
|---------------------|-----------|-----------|---------------------------------------------------------------------------------------------|
| <b>specialDays</b>  | Structure | No        | Array with special days, only for V3 components.                                            |
| :date               | String    | Yes       | Date of the special day, format YYYY-MM-DD.                                                 |
| <b>specialDays1</b> | Structure | No        | Array with special days 1, only for V4 components.                                          |
| :date               | String    | Yes       | Date of the special day 1, format YYYY-MM-DD.                                               |
| <b>specialDays2</b> | Structure | No        | Array with special days 2, only for V4 components.                                          |
| :date               | String    | Yes       | Date of the special day 2, format YYYY-MM-DD.                                               |
| <b>oif</b>          | Structure | No        | A structure with the OiF (Object in Field) parameters, only for LEGIC advant V4 components. |
| :mode               | String    | Yes       | Mode for OiF: <ul style="list-style-type: none"> <li>fix</li> <li>variable.</li> </ul>      |
| :interval           | Integer   | Yes       | Interval in milliseconds if mode is fix or the starting interval if mode is variable.       |
| :restingPeriod      | Integer   | No        | Resting period in milliseconds if mode is variable.                                         |
| <b>components</b>   | Structure | No        | Array with the components that belong to this keyplan, 6.5.7 - Component structure.         |

### 6.5.10 Stamp structure

| JSON key                      | Type    | Mandatory | Meaning                                                                                                                                                                                                                                            |
|-------------------------------|---------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>priority</b>               | Integer | Yes       | Stamp priority (-1, 0 to 7).                                                                                                                                                                                                                       |
| <b>stamp</b>                  | String  | Yes       | The stamp itself.                                                                                                                                                                                                                                  |
| <b>cardLinkFileOffset</b>     | Integer | No        | The CardLink file offset, default 0.                                                                                                                                                                                                               |
| <b>cardLinkEnabled</b>        | Boolean | Yes       | "True" if CardLink is enabled, otherwise "False".                                                                                                                                                                                                  |
| <b>cardLinkFileId</b>         | Integer | No        | The CardLink file ID, default 32 (0x20).                                                                                                                                                                                                           |
| <b>identificationEnabled</b>  | Boolean | Yes       | "True" if Identification is enabled, otherwise "False".                                                                                                                                                                                            |
| <b>identificationType</b>     | String  | No        | The identification type: <ul style="list-style-type: none"> <li>• Identification</li> <li>• KabaGroupHeader</li> <li>• KabaAdvantId</li> <li>• LegicAccessSegment</li> <li>• UniversalIdentificationSegment</li> <li>• TaifunIdSegment.</li> </ul> |
| <b>identificationFileId</b>   | Integer | No        | The Identification file ID, default 0 (0x00).                                                                                                                                                                                                      |
| <b>actuatorStatusEnabled</b>  | Boolean | Yes       | "True" if ActuatorStatus is enabled, otherwise "False".                                                                                                                                                                                            |
| <b>actuatorStatusFileId</b>   | Integer | No        | The ActuatorStatus file ID, default 1 (0x01).                                                                                                                                                                                                      |
| <b>tracebackEnabled</b>       | Boolean | Yes       | "True" if Traceback is enabled, otherwise "False".                                                                                                                                                                                                 |
| <b>tracebackFileId</b>        | Integer | No        | The Traceback file ID, default 2 (0x02).                                                                                                                                                                                                           |
| <b>freeSelectionEnabled</b>   | Boolean | Yes       | "True" if FreeSelection is enabled, otherwise "False".                                                                                                                                                                                             |
| <b>freeSelectionFileId</b>    | Integer | No        | The FreeSelection file ID, default 48 (0x30).                                                                                                                                                                                                      |
| <b>useStampAsSearchString</b> | Boolean | Yes       | "True" if a stamp is used as a search string (omit the file ID), otherwise "False". Setting this value to true is only possible if just one segment is selected.                                                                                   |

### 6.5.11 Master structure

| JSON key               | Type   | Mandatory | Meaning                                                                                                                                  |
|------------------------|--------|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>uid</b>             | String | Yes       | UID of the master.                                                                                                                       |
| <b>name</b>            | String | No        | Optional name for the master.                                                                                                            |
| <b>type</b>            | String | Yes       | Type of the master, i.e. MasterA or MasterB.                                                                                             |
| <b>mediaTechnology</b> | String | Yes       | Media technology of the master: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• LegicPrime</li> <li>• Mifare.</li> </ul> |

### 6.5.12 DoorGroup structure

| JSON key                | Type    | Mandatory | Meaning                                                             |
|-------------------------|---------|-----------|---------------------------------------------------------------------|
| <b>number</b>           | Integer | Yes       | Number of the door group.                                           |
| <b>name</b>             | String  | Yes       | Name of the door group.                                             |
| <b>timeProfileIndex</b> | Integer | Yes       | Index of the assigned time profile, 6.5.14 - TimeProfile structure. |
| <b>timeProfileName</b>  | String  | Yes       | Name of the assigned time profile, 6.5.14 - TimeProfile structure.  |

### 6.5.13 MaintenanceMediaEntry structure

| JSON key                      | Type    | Mandatory | Meaning                                                                                                                                                                                                        |
|-------------------------------|---------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>technology</b>             | String  | Yes       | Technology of this entry: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• Mifare.</li> </ul>                                                                                                   |
| <b>applicationId</b>          | Integer | Yes       | The stamp/sitekey index of the entry: <ul style="list-style-type: none"> <li>• 0...7 =&gt; index of the stamp or sitekey, cardId is a CID</li> <li>• 8 =&gt; cardId is a UID.</li> </ul>                       |
| <b>cardId</b>                 | String  | Yes       | CID or UID of the entry                                                                                                                                                                                        |
| <b>identificationStandard</b> | String  | Yes       | Coding of the entry: <ul style="list-style-type: none"> <li>• Identification (for LEGIC advant and MI-FARE)</li> <li>• KabaGroupHeader (for LEGIC prime)</li> <li>• KabaAdvantId (for LEGIC advant)</li> </ul> |
| <b>description</b>            | String  | No        | Optional description of the entry, e.g. user name.                                                                                                                                                             |

### 6.5.14 TimeProfile structure

| JSON key                  | Type      | Mandatory | Meaning                                                                                                                                                                                                                                      |
|---------------------------|-----------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>index</b>              | Integer   | Yes       | Index of the time profile.                                                                                                                                                                                                                   |
| <b>name</b>               | String    | Yes       | Name of the time profile.                                                                                                                                                                                                                    |
| <b>timeWindows</b>        | Structure | Yes       | Array of time windows.                                                                                                                                                                                                                       |
| :active                   | Boolean   | Yes       | "True" if this time window is active, otherwise "False".                                                                                                                                                                                     |
| :startTime                | String    | Yes       | Start time of the window, format HHMM, e.g. 0800.                                                                                                                                                                                            |
| :endTime                  | String    | Yes       | End time of the window, format HHMM, e.g. 1800.                                                                                                                                                                                              |
| :monday                   | Boolean   | No        | "True" if time window is valid on Mondays.                                                                                                                                                                                                   |
| :tuesday                  | Boolean   | No        | "True" if time window is valid on Tuesdays.                                                                                                                                                                                                  |
| :wednesday                | Boolean   | No        | "True" if time window is valid on Wednesdays.                                                                                                                                                                                                |
| :thursday                 | Boolean   | No        | "True" if time window is valid on Thursdays.                                                                                                                                                                                                 |
| :friday                   | Boolean   | No        | "True" if time window is valid on Fridays.                                                                                                                                                                                                   |
| :saturday                 | Boolean   | No        | "True" if time window is valid on Saturdays.                                                                                                                                                                                                 |
| :sunday                   | Boolean   | No        | "True" if time window is valid on Sundays.                                                                                                                                                                                                   |
| :validOnNormalDays        | Boolean   | Yes       | "True" if time window is valid on normal days.                                                                                                                                                                                               |
| :validOnHolidays          | Boolean   | Yes       | "True" if time window is valid on holidays.                                                                                                                                                                                                  |
| :validOnSpecialDays       | Boolean   | No        | "True" if time window is valid on special days (only for V3 components).                                                                                                                                                                     |
| :validOnSpecialDays1      | Boolean   | No        | "True" if time window is valid on special days 1 (only for V4 components).                                                                                                                                                                   |
| :validOnSpecialDays2      | Boolean   | No        | "True" if time window is valid on special days 2 (only for V4 components).                                                                                                                                                                   |
| <b>officeModeFunction</b> | String    | No        | <p>If the time profile is a TimePro, then this is the function of the office mode:</p> <ul style="list-style-type: none"> <li>• Standard</li> <li>• Office</li> <li>• OfficeIndividual</li> <li>• OfficeMedia</li> <li>• DayNight</li> </ul> |

## 6.5.15 Order structure

For a detailed description of the various fields please refer to the B-COMM manual.

| JSON key                  | Type    | Mandatory | Meaning                                                              |
|---------------------------|---------|-----------|----------------------------------------------------------------------|
| <b>section</b>            | String  | No        | Section name of an order.                                            |
| <b>command</b>            | String  | Yes       | Command of an order.                                                 |
| <b>subCommand</b>         | String  | No        | Subcommand of an order.                                              |
| <b>status</b>             | String  | No        | Execution state of an order.                                         |
| <b>alternativeld</b>      | Integer | No        | Alternative ID of a terminal.                                        |
| <b>groupid</b>            | Integer | No        | Group address (GID) of a terminal.                                   |
| <b>deviceid</b>           | Integer | No        | Device address (DID) of a terminal.                                  |
| <b>extension</b>          | String  | No        | Extension field of an order.                                         |
| <b>file</b>               | String  | No        | File field of an order.                                              |
| <b>data</b>               | String  | No        | Data field of an order.                                              |
| <b>priority</b>           | Integer | No        | Priority of an order (1 or 2).                                       |
| <b>schedulerActivated</b> | Boolean | No        | "True" if the order is scheduled, otherwise "False".                 |
| <b>schedulerStartTime</b> | String  | No        | Start time of the order in format "HHMM", e.g. "1805".               |
| <b>schedulerInterval</b>  | Integer | No        | Interval of the order in minutes.                                    |
| <b>ftpEnabled</b>         | Boolean | No        | "True" if the order should be executed by (S)FTP, otherwise "False". |



### 6.5.16 WhitelistEntry structure

| JSON key                      | Type    | Mandatory | Meaning                                                                                                                                                                                                         |
|-------------------------------|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>technology</b>             | String  | Yes       | The technology of this entry: <ul style="list-style-type: none"> <li>• LegicAdvant</li> <li>• Mifare.</li> </ul>                                                                                                |
| <b>applicationId</b>          | Integer | Yes       | The stamp/sitekey index of the entry: <ul style="list-style-type: none"> <li>• 0...7 =&gt; index of the stamp or sitekey, cardId is a CID</li> <li>• 8 =&gt; cardId is a UID.</li> </ul>                        |
| <b>cardId</b>                 | String  | Yes       | CID or UID of the entry.                                                                                                                                                                                        |
| <b>identificationStandard</b> | String  | Yes       | Coding of the entry: <ul style="list-style-type: none"> <li>• Identification (for LEGIC advant and MI-FARE)</li> <li>• KabaGroupHeader (for LEGIC prime)</li> <li>• KabaAdvantId (for LEGIC advant).</li> </ul> |
| <b>description</b>            | String  | No        | Optional description of the entry, e.g. user name.                                                                                                                                                              |
| <b>officeIndividual</b>       | Boolean | No        | "True" if OfficeIndividual is active, otherwise "False".                                                                                                                                                        |
| <b>safeUID</b>                | Boolean | No        | "True" if entry has a secure UID, otherwise "False".                                                                                                                                                            |
| <b>timeProfileIndex</b>       | Integer | Yes       | Index of the corresponding time profile.                                                                                                                                                                        |

### 6.5.17 EventLogEntry structure

| JSON key               | Type    | Mandatory | Meaning                                                                                                                                                                                                                                                                                   |
|------------------------|---------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>eventCode</b>       | Integer | Yes       | Code of the event.                                                                                                                                                                                                                                                                        |
| <b>online</b>          | Boolean | No        | "True" if component has connection to gateway, otherwise "False". Only for wireless components.                                                                                                                                                                                           |
| <b>hallSensorState</b> | Boolean | No        | "True" if hall sensor is in state "down", otherwise "False". Only for digital cylinders.                                                                                                                                                                                                  |
| <b>batteryLow</b>      | Boolean | Yes       | "True" if battery low was detected, otherwise "False".                                                                                                                                                                                                                                    |
| <b>clockError</b>      | Boolean | Yes       | "True" if a clock error was detected, otherwise "False".                                                                                                                                                                                                                                  |
| <b>doorState</b>       | Boolean | Yes       | "True" if door was open (closing), "False" if door was closed (opening). Primarily for Cabinet Lock.                                                                                                                                                                                      |
| <b>mediaType</b>       | String  | Yes       | Type of the media: <ul style="list-style-type: none"> <li>UNKNOWN</li> <li>MASTER_A</li> <li>MASTER_B</li> <li>USER_MEDIA</li> <li>SMODULE</li> <li>TIMER</li> <li>USER_MEDIA_WITH_CID</li> <li>MASTER_T</li> <li>USER_MEDIA_ELOSTAR_WITHOUT_KABA_CODE</li> <li>WIRELESS_AIDA.</li> </ul> |
| <b>cardId</b>          | String  | Yes       | UID or CID of the media.                                                                                                                                                                                                                                                                  |
| <b>timestamp</b>       | String  | Yes       | The timestamp of the event.                                                                                                                                                                                                                                                               |

| JSON key          | Type   | Mandatory | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|--------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>technology</b> | String | Yes       | <p>The content of this field depends on the component technology.</p> <p>For Evolo components with MIFARE:</p> <ul style="list-style-type: none"> <li>• Unknown/Irrelevant</li> <li>• MIFARE Classic</li> <li>• MIFARE DESFire</li> <li>• MIFARE Ultralight</li> <li>• Other (ISO) media.</li> </ul> <p>For Evolo components with LEGIC advant:</p> <ul style="list-style-type: none"> <li>• Unknown/Irrelevant</li> <li>• LegicPrime</li> <li>• LegicAdvant</li> <li>• Other (ISO) media.</li> </ul> <p>For REMP components with MIFARE and LEGIC advant:</p> <ul style="list-style-type: none"> <li>• Unknown/Irrelevant</li> <li>• MIFARE Classic</li> <li>• MIFARE DESFire</li> <li>• MIFARE Ultralight</li> <li>• FeliCa</li> <li>• LegicPrime</li> <li>• LegicAdvant</li> <li>• MIFARE Plus</li> <li>• Mobile</li> <li>• Other (ISO) media.</li> </ul> <p>For Evolo components with activated CID, the technology field will contain either the sitekey index (MIFARE) or the stamp index (LEGIC advant): 0 to 7.</p> |

### 6.5.18 BlacklistEntry structure

| JSON key              | Type    | Mandatory | Meaning                                                                                                                                                                             |
|-----------------------|---------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>applicationId</b>  | Integer | Yes       | The stamp/sitekey index of the entry. <ul style="list-style-type: none"> <li>0...7 =&gt; index of the stamp or sitekey, cardId is a CID</li> <li>8 =&gt; cardId is a UID</li> </ul> |
| <b>cardId</b>         | String  | Yes       | CID or UID of the entry.                                                                                                                                                            |
| <b>expirationDate</b> | String  | Yes       | Expiration date of the entry in format YYYY-MM-DDTHH:MM:SS.                                                                                                                         |
| <b>description</b>    | String  | No        | Optional description of the entry, e.g. user name.                                                                                                                                  |

### 6.5.19 StaticCredential structure

| JSON key                  | Type    | Mandatory | Meaning                                                                                                                                                                  |
|---------------------------|---------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>fileDefinitionName</b> | String  | No        | Name of the file definition as defined in LEGIC Control Center, which serves as a template for the static credential file. If omitted, the name staticCred will be used. |
| <b>phoneNo</b>            | String  | Yes       | Phone number of the mobile phone.                                                                                                                                        |
| <b>cardId</b>             | String  | Yes/No    | The UID/CID of the mobile phone.<br><br>Mandatory for command setMobileAccessCredentials<br><br>Optional for command deleteMobileAccessCredentials .                     |
| <b>isHex</b>              | Boolean | No        | "True" if cardId is a hex value, otherwise "False". Default: false.                                                                                                      |
| <b>bleAccess</b>          | Boolean | No        | "True" if mobile phone can book by BLE, otherwise "False". Default: true.                                                                                                |
| <b>nfcAccess</b>          | Boolean | No        | "True" if mobile phone can book by NFC, otherwise "False". Default: true.                                                                                                |

### 6.5.20 ProjectDevice structure

| JSON key       | Type   | Mandatory | Meaning                                                                                                                           |
|----------------|--------|-----------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>phoneNo</b> | String | Yes       | Phone number of the mobile phone, format phone#+xxxxxxxxxxxx.                                                                     |
| <b>name</b>    | String | Yes/No    | Name of the mobile phone.<br><br>Mandatory for command createProjectDevices.<br><br>Unnecessary for command deleteProjectDevices. |

# Index

## A

|                                               |          |                                                    |     |
|-----------------------------------------------|----------|----------------------------------------------------|-----|
| <b>A</b>                                      |          |                                                    |     |
| add_bio_template                              | 141      | "listProvisionedFiles"                             | 264 |
| additional_information, gateway or components | 37       | "listVcps"                                         | 267 |
| AMQP interface                                |          | "reorganizeOrders"                                 | 218 |
| broker                                        | 145      | "restartServer"                                    | 249 |
| broker, installing the..                      | 145      | "sendParameters"                                   | 227 |
| command structure                             | 162      | "sendRecord"                                       | 215 |
| commands from applic. to B-COMM server        | 162      | "sendWhitelist"                                    | 238 |
| data structure                                | 155      | "setAdministrationAreas"                           | 194 |
| error structure                               | 160      | "setComponents"                                    | 208 |
| header structure                              | 152      | "setDoorLocked"                                    | 242 |
| Introduction                                  | 144      | "setDoorNormalOperation"                           | 243 |
| message structure                             | 152      | "setDoorReleased"                                  | 241 |
| response structure                            | 162, 163 | "setGateways"                                      | 189 |
| AMQP interface: commands                      |          | "setKeyplans"                                      | 199 |
| "addWhitelist"                                | 236      | "setMobileAccessCredentials"                       | 252 |
| "addWhitelistPool"                            | 247      | "setWhitelist"                                     | 234 |
| "createOrders"                                | 224      | "setWhitelistPool"                                 | 246 |
| "createProjectDevices"                        | 258      | "shutdownServer"                                   | 248 |
| "deleteMobileAccessCredentials"               | 256      | "startStream"                                      | 174 |
| "deleteOrders"                                | 226      | "stopStream"                                       | 175 |
| "deleteProjectDevices"                        | 262      | "writeMobileAccessCredentials"                     | 254 |
| "deleteVcps"                                  | 269      | "writeVcps"                                        | 271 |
| "deleteWhitelist"                             | 239      | Application interface                              |     |
| "getAdministrationArea"                       | 192      | Gateway or component event, additional_information | 37  |
| "getAdministrationAreas"                      | 190      | Gateway or component event, event_code             | 30  |
| "getBookings"                                 | 250      | Gateway or component event, event_source           | 22  |
| "getClient"                                   | 168      | Gateway or component event, log_level              | 23  |
| "getClients"                                  | 166      | Gateway or component event, media_information      | 35  |
| "getComponent"                                | 204      | Gateway or component event, message_id             | 24  |
| "getComponentEventLog"                        | 213      | Gateway or component event, sensor                 | 39  |
| "getComponentEventLogFile"                    | 209      | Application interface for HTTP helper              | 129 |
| "getComponents"                               | 200      |                                                    |     |
| "getConfiguration"                            | 164      | <b>B</b>                                           |     |
| "getGateway"                                  | 187      | B-COMM connector in HTTP helper                    | 126 |
| "getGateways"                                 | 185      | B-COMM Helper                                      | 40  |
| "getKeyplan"                                  | 197      | Commands                                           | 40  |
| "getKeyplans"                                 | 195      | B-COMM Helper, calling the..                       | 40  |
| "getKnownGateways"                            | 183      | B-COMM Helper, introduction                        | 40  |
| "getOrder"                                    | 222      | B-COMM structures                                  |     |
| "getOrders"                                   | 220      | AdministrationArea structure                       | 289 |
| "getStream"                                   | 172      | ApplicationInterface structure                     | 281 |
| "getStreams"                                  | 170      | BlacklistEntry structure                           | 300 |
| "getStreamStatus"                             | 176      | Client structure                                   | 276 |
| "getTerminal"                                 | 180      | Component structure                                | 283 |
| "getTerminals"                                | 178      | DoorGroup structure                                | 294 |
| "getWhitelist"                                | 231      | EventLogEntry structure                            | 298 |
| "getWhitelistPool"                            | 244      | Gateway structure                                  | 282 |
| "listProjectDevices"                          | 260      | Keyplan structure                                  | 291 |

|                                                |     |                                        |          |
|------------------------------------------------|-----|----------------------------------------|----------|
| MaintenanceMediaEntry structure                | 294 | DeleteBookingFile                      | 44       |
| Master structure                               | 294 | DeleteMobileAccessProjectDevices       | 114      |
| NetworkAdapter structure                       | 279 | DeleteMobileAccessVcps                 | 120      |
| Order structure                                | 296 | DeleteOrderFile                        | 44       |
| Stamp structure                                | 293 | DeleteWhitelistActuatorWhitelist       | 96       |
| StaticCredential structure                     | 300 | DeleteWhitelistMobileAccessCredentials | 113      |
| Stream structure                               | 278 | DoorControl                            | 100, 109 |
| Terminal structure                             | 280 | FileDownload                           | 52       |
| TimeProfile structure                          | 295 | FileUpload                             | 45       |
| WhitelistEntry structure                       | 297 | FileUploadContent                      | 45       |
| bcommhttphelperconfigurator.bat file           | 133 | ListMobileAccessProjectDevices         | 115      |
| Broker                                         | 145 | ListMobileAccessProvisionedFiles       | 118      |
| Broker, data exchange with..                   | 150 | ListMobileAccessVcps                   | 117      |
| Broker, global information on RabbitMQ         | 146 | ReadActuatorCardLinkConfigStatus       | 69       |
| Broker, queues and routing keys                | 150 | ReadActuatorCardLinkEventLog           | 66       |
| Broker, TLS configuration                      | 148 | ReadActuatorConfigStatus               | 62       |
|                                                |     | ReadActuatorEventLog                   | 64       |
|                                                |     | ReadActuatorJoinStates                 | 99, 108  |
|                                                |     | ReadActuatorWhitelistConfigStatus      | 67       |
|                                                |     | ReadActuatorWhitelistEventLog          | 65       |
|                                                |     | ReadBookings                           | 77       |
|                                                |     | ReadCardLinkConfiguration              | 78       |
|                                                |     | ReadClient                             | 56       |
|                                                |     | ReadClients                            | 57       |
|                                                |     | ReadConfiguration                      | 54       |
|                                                |     | ReadGateways                           | 99, 107  |
|                                                |     | ReadGlobalConfigValue                  | 55       |
|                                                |     | ReadKnownGateways                      | 99, 106  |
|                                                |     | ReadLegicConnectEvents                 | 116      |
|                                                |     | ReadNetworkAdapter                     | 55       |
|                                                |     | ReadNetworkAdapters                    | 56       |
|                                                |     | ReadOrders                             | 71       |
|                                                |     | ReadOrdersExecuted                     | 71       |
|                                                |     | ReadOrderStatus                        | 72       |
|                                                |     | ReadProgrammerStatus                   | 110      |
|                                                |     | ReadRecentOrderStatus                  | 72       |
|                                                |     | ReadStream                             | 58       |
|                                                |     | ReadStreamConfigValue                  | 59       |
|                                                |     | ReadStreams                            | 58       |
|                                                |     | ReadStreamStatus                       | 59       |
|                                                |     | ReadTerminal                           | 60       |
|                                                |     | ReadTerminals                          | 60       |
|                                                |     | ReadTerminalSections                   | 61       |
|                                                |     | RefreshClients                         | 57       |
|                                                |     | RefreshTemplateList                    | 109      |
|                                                |     | RefreshTemplateList (Client)           | 109      |
|                                                |     | RenameCurrentBookingFile               | 53       |
|                                                |     | ReorganizeOrderFile                    | 53       |
|                                                |     | SendDataRecord                         | 52       |
|                                                |     | ShutdownHelper                         | 42       |
|                                                |     | ShutdownServer                         | 42       |
|                                                |     | SSHKeyExport                           | 121      |
|                                                |     | StartStream                            | 42       |
|                                                |     | StopStream                             | 43       |
| <b>C</b>                                       |     |                                        |          |
| Certificates, provision for HTTP helper        | 132 |                                        |          |
| Communication B-COMM IO/Applic.                |     |                                        |          |
| Acknowledgements                               | 17  |                                        |          |
| B-COMM ApplicationDemo                         | 16  |                                        |          |
| B-COMM Class documentation                     | 16  |                                        |          |
| Commands                                       | 17  |                                        |          |
| Communication examples                         | 18  |                                        |          |
| Data                                           | 17  |                                        |          |
| Data classification                            | 17  |                                        |          |
| Introduction                                   | 16  |                                        |          |
| Configuration manager                          | 125 |                                        |          |
| Configuration of HTTP helper                   | 129 |                                        |          |
| Configuring the HTTP helper                    | 133 |                                        |          |
| Connection test, HTTP helper                   | 134 |                                        |          |
| Creating application interface for HTTP helper | 129 |                                        |          |
| <b>D</b>                                       |     |                                        |          |
| Data record examples:                          |     |                                        |          |
| from a component                               | 275 |                                        |          |
| from a gateway                                 | 274 |                                        |          |
| from a terminal                                | 273 |                                        |          |
| <b>E</b>                                       |     |                                        |          |
| event_code, gateway or components              | 30  |                                        |          |
| event_source, gateway or components            | 22  |                                        |          |
| Examples                                       | 273 |                                        |          |
| <b>G</b>                                       |     |                                        |          |
| get_all_bio_templates                          | 142 |                                        |          |
| get_bc_config                                  | 137 |                                        |          |
| <b>H</b>                                       |     |                                        |          |
| Helper command                                 |     |                                        |          |
| AppendWhitelistActuatorWhitelist               | 95  |                                        |          |
| CreateMobileAccessProjectDevices               | 114 |                                        |          |

|                                               |                              |                                           |     |
|-----------------------------------------------|------------------------------|-------------------------------------------|-----|
| SwitchApplicationInterface                    | 43                           | Components of the helper                  | 125 |
| UpdateCardLinkActuatorConfiguration           | 79                           | Configuration manager                     | 125 |
| UpdateCardLinkAdministrationAreaConfiguration | 82                           | Configuring the HTTP helper               | 133 |
| UpdateCardLinkGatewayConfiguration            | 105                          | Configuring the..                         | 129 |
| UpdateWhitelistActuatorConfiguration          | 89                           | Connection test                           | 134 |
| UpdateWhitelistGatewayConfiguration           | 98                           | Creating an application interface         | 129 |
| UpdateWhitelistKeyPlanConfiguration           | 92                           | Graphical overview                        | 123 |
| WriteCardLinkActuatorConfiguration            | 78                           | Graphical overview on the communication   | 124 |
| WriteCardLinkAdministrationAreaConfiguration  | 81                           | HTTP client                               | 126 |
| WriteCardLinkConfiguration                    | 84, 102                      | HTTP server                               | 126 |
| WriteCardLinkGatewayConfiguration             | 104                          | I/O connector                             | 126 |
| WriteElologicActuatorConfiguration            | 111                          | Implementing the h. in the system         | 123 |
| WriteElologicKeyPlanConfiguration             | 111                          | Introduction                              | 122 |
| WriteMobileAccessVcps                         | 121                          | Provision of the certificates             | 132 |
| WriteOrder                                    | 73                           | Response codes                            | 128 |
| WriteOrderDirect                              | 75                           | Template manager                          | 127 |
| WriteOrderOverwrite                           | 74                           | HTTP helper commands                      |     |
| WriteOrderOverwriteDirect                     | 76                           | add_bio_template                          | 141 |
| WriteWhitelistActuatorConfiguration           | 88                           | delete_bio_template                       | 142 |
| WriteWhitelistActuatorWhitelist               | 94                           | get_all_bio_templates                     | 142 |
| WriteWhitelistConfiguration                   | 86                           | get_bc_config                             | 137 |
| WriteWhitelistGatewayConfiguration            | 97                           | send_data                                 | 138 |
| WriteWhitelistKeyPlanConfiguration            | 91                           | send_xml                                  | 143 |
| WriteWhitelistMobileAccessCredentials         | 112                          | start_stream                              | 139 |
| WriteWhitelistPool                            | 85                           | stop_stream                               | 140 |
| Helper commands                               |                              | HTTP server                               | 126 |
| Sorting                                       | 41                           | httphelper.ini file                       | 125 |
| with XML                                      | 40                           | httphelper.ini file, checking the..       | 135 |
| Helper commands, sorted                       |                              | <b>I</b>                                  |     |
| - actuator related                            | 62, 67, 69                   | I/O connector                             | 126 |
| - biometric related                           | 109                          | <b>L</b>                                  |     |
| - booking related                             | 77                           | log_level, gateway or components          | 23  |
| - CardLink related                            | 78                           | <b>M</b>                                  |     |
| - CardLink wireless related                   | 102                          | media_information, gateway or components  | 35  |
| - client related                              | 56                           | message_id, gateway or components         | 24  |
| - configuration related                       | 54                           | <b>O</b>                                  |     |
| - elologic related                            | 111                          | Orders for WriteOrder                     | 9   |
| - file related                                | 44                           | <b>P</b>                                  |     |
| - for starting and terminating of functions   | 42                           | Provision of certificates for HTTP helper | 132 |
| - mobile access related                       | 112, 116, 117, 118, 120, 121 | <b>R</b>                                  |     |
| - order related                               | 71                           | Response codes in HTTP helper             | 128 |
| - programmer related                          | 110                          | <b>S</b>                                  |     |
| - SSH key related                             | 121                          | send_data                                 | 138 |
| - stream related                              | 58                           | send_xml                                  | 143 |
| - terminal related                            | 60                           | sensor, gateway or components             | 39  |
| - whitelist related                           | 85                           |                                           |     |
| Helper, calling the..                         | 40                           |                                           |     |
| HTTP client                                   | 126                          |                                           |     |
| HTTP helper                                   |                              |                                           |     |
| B-COMM connector                              | 126                          |                                           |     |
| bcommhttphelperconfigurator.bat file          | 133                          |                                           |     |
| Biometric orders                              | 127                          |                                           |     |
| Checking the httphelper.ini file              | 135                          |                                           |     |

|              |     |
|--------------|-----|
| start_stream | 139 |
| stop_stream  | 140 |

## T

|                  |     |
|------------------|-----|
| Template manager | 127 |
|------------------|-----|

## V

|         |   |
|---------|---|
| Version | 6 |
|---------|---|

## W

|                                           |    |
|-------------------------------------------|----|
| WriteOrder                                |    |
| Arguments                                 | 8  |
| Examples                                  | 12 |
| Further order parameters                  | 11 |
| Introduction                              | 8  |
| Options                                   | 8  |
| Orders for biometrics                     | 10 |
| Orders for CardLink Wireless or Whitelist | 10 |
| Orders for stream BCSCWxy                 | 10 |
| Orders for streams BCLANxy or BCSErxy     | 9  |
| orders for..                              | 9  |



## Notes

[illegible]



[illegible]

04047683 - 05/2023  
Copyright © dormakaba 2023



[www.dormakaba.com](http://www.dormakaba.com)