dormakaba

B-Client HR40

Reference Manual

040500544 - 08/2024

EN

Table of content

1 About this manual

Validity

This manual describes the dormakaba application B-Client HR40 as of

- Manufacturing date: June 2024
- Firmware version: 739-00-X-K00
- Android 12

Addressees

The explanations are addressed to software developers and programmers of applications, they cannot replace product training. Knowledge about the structure of configuration files is imperative (.ini files).

Contents and purpose

The descriptions in this manual are limited to the presentation of individual parameter and data records as well as possible entries in different .ini files. When using proprietary parameter editor, the user must be able to activate the required functions by means of parameter records.

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 screenshots or examples like names, IP addresses and similar data are dummy data.

Additional documentation

Communication software

- B-COMM manual

Orientation in the manual

This manual contains the following orientation guide to find specific topics more easily:

- The table of contents at the beginning of the manual gives an overview of all topics.
- The header always contains the respective main chapter.
- Cross references. They refer to chapters where additional information about a certain topic can be found. Example [▸ 5.7].

  PDF relating note

  When using cross references in PDF documents, you can easily turn back to the source page by using the key combination ALT-Left Arrow (depending on the PDF reader).
- Glossary. A glossary can be found in the beginning of certain chapters. It summarizes the most important terms and gives a short description on them.
- Index. An index in alphabetical order is at the end of the manual.

Modification log

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

- First edition of this manual in June 2024

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 Product description

2.1 Overview

The B-Client HR40 device software is based on the Android operating system (currently Android 12) and is optimized for the NexT terminal series with the dormakaba terminals 9600-K7, 9700-K7 and 9800-K7 devices as well as the terminal ONE-K7.
In conjunction with these time and data recording terminals, the software offers the following range of functions, depending on the function variant purchased.

9600-K7
ONE-K7

9700-K7

9800-K7

2.2 Function range (summary)

- Time recording for more than 10,000 persons/50,000 data records, ONE-K7 up to 1,000 persons
- Multilingual user interface with easy language selection
- Separate URL start address for every function key definable
- Supports biometric CBM reader module and local enrollment
- Operating of a LEGIC advant reader
- Supports CardLink functionality
- Secured data transmission
- Inputs and outputs can be freely assigned by means of a I/O Mapping section
- Supports Mifare readers with ARIOS concept
- Freely definable user interface
- Supporting of dynamic lists via G dialog
- Calling a of browser or additional apps via partner interface
- Displaying of time sheets via function keys
- Booking by use of cellular phones (Mobile Access)

2.3 Function variants

The B-Client HR40 consists of several functional units, which are available as individual function variants. Different functions are offered depending on the device type or function variant purchased.

Terminals

- dormakaba Terminal 96 00-K7
- dormakaba Terminal 97 00-K7
- dormakaba Terminal 98 00-K7

Function variant K7-20	Basic device with B-Client HR40 and • >10,000 master records • Time recording functions • MobileAccess
Function variant K7-40	Basic device with B-Client HR40 and • >10,000 master records • Time recording functions • MobileAccess • Operating of CardLink components • Supports AoC (Access on Card)
Function variant K7-60	Basic device with B-Client HR40 and • >10,000 master records • Time recording functions • MobileAccess • Operating of CardLink components • Supports AoC (Access on Card) • Door control (access control functions and relay)

Terminal

- dormakaba Terminal ONE-K7

No function variant	Basic device with B-Client HR40 and • 1,000 master records • Time recording functions • Optional: Door control (access control functions and relay)

2.4 Licensing

The required functions can be activated at any time using the various function variants of the software.

A valid license file (sop.ini) must be transmitted via FTP into the directory ../data/data/ com.kaba.apps.hr/files/init.

After the transmission has been completed, depending on the licensed option the terminal must be restarted as follows:

Cold start

- CardLink

A cold start must always be performed for licensing the CardLink function.

All master records or templates available in the terminal are deleted during this process.

Warm start

- Door control
- MobileAccess

The possible function variants and their properties are listed below.

Please observe the following notes!

Icon	Function variant	Description
DC	Door control	This function variant includes the following door control features: • Activating the relay • Monitoring the door contacts • Double access blocking • Use of door open profiles • Using the F00 function key • Code input.
CL	CardLink	The function variant allows data writing (update and validation records) on badges (LEGIC and MIFARE). It also supports AoC (Access on Card).
BIO	Biometrics	The function variant includes booking at the terminal using a smartphone.
MA	MobileAccess	The (hardware) function variant allows the use of biometric readers.

Note 1)

For the integration of a partner app, a partner app demo is available, which already contains the most important program parts for communication. This shortens development times and increases operational reliability in connection with the B-Client used.

Note 2)

dormakaba terminals are not comparable with commercially available mobile phones or tablets, which are exchanged in short cycles and always equipped with the latest, multi-functional Android operating system for universal applications. The dormakaba terminal hardware used and the Android operating system adapted to it are optimized for a specific application with a focus on maximum operational security, high data protection, low energy consumption and economical memory usage.
Therefore it cannot be guaranteed that any native Android app will run in this system environment.
This also applies to Internet browsers, which are continuously changed and updated by the manufacturers.

2.5 Memory partitioning without/with CardLink function variant

The number of the master records as well as the update/validation records for the CardLink function variant depends on the licensing, see the tables below.

B-Client HR40 without CardLink

Standard configuration	Master records
10,000+	50,000

B-Client HR40 with CardLink

Function variant	Master records	Update records	Validation records
10,000+ with CardLink	30,000	60,000	30,000

2.6 Security hints for 3rd party apps

2.6.1 General hint

NOTICE

Functional impairment due to native apps

The installation and/or use of your own apps or apps from third-party manufacturers can impair or even block the function of the device.

The installation and/or use of proprietary apps or apps from third-party manufacturers is **exclusively**in the responsibility of the user.

2.6.2 Subsequent authorizations at runtime

Subsequently installed apps, in particular browsers (example: Chrome), offer the possibility to subsequently load and install additional apps or program parts without further authorization and install them without asking the user. Under certain circumstances, this can be a security risk.

To minimize this security risk, dormakaba strongly recommends that the effective setting in the Android operating system in the parameter REQUEST_INSTALL_PACKAGES is defined accordingly.

Procedure

1 Start the SSH server in the device using the TASSH START command record, see also chapter [▶ 3.8].

2 Start a suitable SSH client on the host, e.g. PuTTY, install it beforehand if necessary.

3 Establish an SSH connection to the device with the appropriate key.

4 Use the command `appops set com.android.chrome REQUEST_INSTALL_PACKAGES ignore` to revoke the authorization for the application with the corresponding installation of further packages.
In the example, this is the package "`com.android.chrome`".
This specification must be adjusted accordingly for other browsers/packages.

5 Terminate the connection. A reboot is not necessary.

Finally, the SSH server in the device can be terminated again with the command record "TASSH STOP" if it is no longer used for other applications (e.g. B-COMM).

2.7 Directory structure

All system settings of the device are stored in .ini files. The following overview shows the directories containing the .ini files intended to be modified by the customer. A detailed description of these .ini files and the parameters used there can be found in chapter [▶ 14].

All other files in the directory structure are not intended for customization and may not be changed!

NOTICE

Total breakdown of the device due to misconfiguration of the .ini files. The device is no longer bootable, usable or shows incorrect or instable behavior.

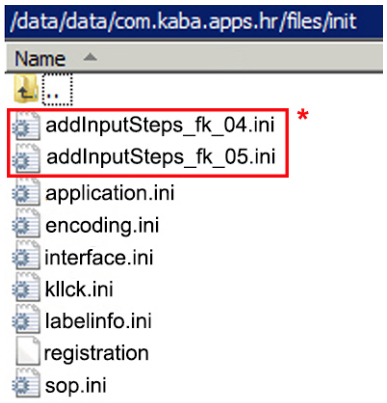
- Experience on how to use .ini files is mandatory.

- Customizing is only permitted in the .ini files intended for this purpose.

- dormakaba recommends creating backup files of the .ini files before making any customizations.

- Define only parameter values that are logical and reasonable.

2.7.1 Directory ../com.kaba.apps.hr

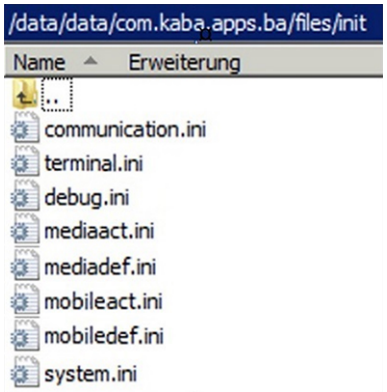


* = files to be created by the customer.

These files contain the function key parameters when using the "additional input steps" function, see chapter [▶ 3.10.31].

In the example parameters are defined for the function keys fk04 and fk05.

2.7.2 Directory ../com.kaba.apps.ba



2.7.3 Directory ../sdcard

The /sdcard directory is located in the root directory of the terminal. This directory has no read or write restrictions. Users can easily use this directory to load data such as images, movies, .pdf files etc. onto the terminal or from the terminal to the host. The directory has several subdirectories, see graphic.

Files that are to be opened from this directory or it´s subdirectories are to be called by use ofa URL. In this URL, the directory name "sdcard" is not to be mentioned, see example here and in chapter [▶ 15] , note 6).

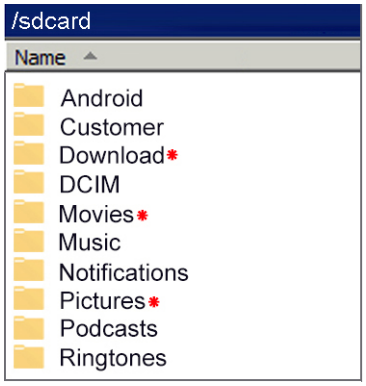
Example for calling a graphic file: URL=file:///Pictures/<Filename>.png

The directory has several subdirectories, see picture.

Only the subdirectories

- Download
- Movies and
- Pictures

are intended for use by the user.



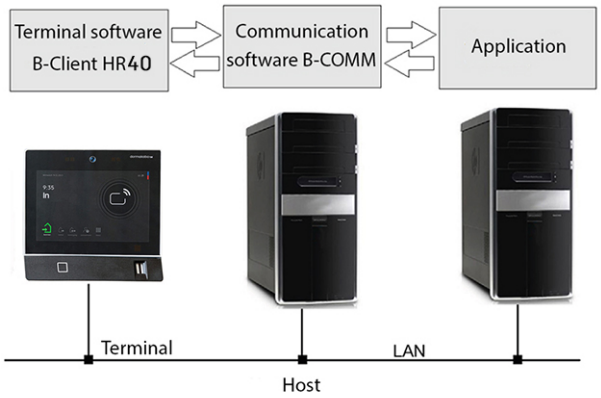
3 Communication

The following chapters contain the following topics:

- basic explanations about the communication between terminal and host
- description of the terms up- and download
- terminal addressing and operating modes
- parameter setting via download records
- parameter setting via SFTP
- remote control via command records.

3.1 Communication concept

The communication between a host computer and the terminals is normally carried out via a network.



Data exchange between the terminal software and the application program takes place by means of the communication software B-COMM. Application program and communication software can be installed on the same computer or on different computers.

The communication software transmits the data records recorded by the terminals to files or transmits them to the application via a defined interface. Data such as user information, time and date or personal master records can be transmitted from the host to a terminal. The parameter setting of the terminals can be managed centrally on the host and transmitted to the terminals.

The terminal can be operated in the online, offline and autonomous operating modes. The data exchange of the terminals with the host is done in the same manner. Likewise, the communication orders are awarded in the corresponding file.

3.2 Upload/Download

The data transfer descriptions in this manual often use the terms "upload" and "download". They are defined as shown below.

- Download: Transfer direction Host --> Terminal
- Upload: Transfer direction Terminal --> Host

3.3 Operating mode

Data records from and to the terminal contain a digit that describes the momentary operating mode of the device. In data records transmitted from the host to the terminal, this digit may be filled with a blank (20hex).

The operating mode is represented by ASCII characters according to the following table.

Operating mode	ASCII	Hex
Online	Space	20
Offline	0	30
Autonomous	@	40

Info	Topic
5.2.3	Switching the operation mode

3.3.1 Online

The "online" operating mode is advisable if the data to be recorded must be permanently available in the host.

After a booking has been made, the terminal carries out the checks and writes the test result as error identification into the input record. The data record is transmitted to the host computer. After that the terminal expects a logical booking response from the host. With this response the terminal is informed by the host if the booking is authorized or not.

If the terminal does not receive a logical booking response from the host, it will change to the offline operating mode and decide itself if the booking is authorized or not. As soon as the host computer is accessible again, the terminal will change back to the online mode. All data records stored in the meantime in the offline mode will be transmitted to the host.

3.3.2 Offline

The "offline" mode is recommended if the recorded data does not have to be permanently available.

After a booking has been made, the terminal carries out the checks and decides immediately by way of an internal booking response if the booking is authorized or not. If the host is available, the data records of authorized and unauthorized bookings (depending on the set parameters) are transmitted. Otherwise, the data records will be stored in the terminal. As soon as the host computer is available, all data records stored since the last transmission are transmitted to the host.

3.3.3 Autonomous

After a booking has been made, the terminal carries out the checks and decides immediately by way of an internal booking response if the booking is authorized or not. Data records of authorized or unauthorized bookings are stored in the terminal, depending on the set parameters.

If the host is available, any existing error and alarm records are transmitted to the host.

The stored data records are transmitted to the host in a special data record after being requested. The data records transmitted to the host are then deleted in the terminal, depending on the setting of the "Buffer Backup" parameter.

Info	Topic
3.9.1	Buffer Backup

3.4 Table of group address (GID) and device address (DID)

The addressing of a device is done by the group ID (GID) and device ID (DID). The first two digits of the data record always contain the terminal identification.

Logical Group Address = GID = Group-ID, allowed values 00 through 29.

Logical Device Address = DID = Device-ID, allowed values 00 through 59.

The group identification (GID) and the device identification (DID) are represented by ASCII characters according to the following table.

Addr.	ASCII	Hex
00	@	40
01	A	41
02	B	42
03	C	43
04	D	44
05	E	45
06	F	46
07	G	47
08	H	48
09	I	49
10	J	4A
11	K	4B
12	L	4C
13	M	4D
14	N	4E
15	O	4F
16	P	50
17	Q	51
18	R	52
19	S	53
20	T	54
21	U	55
22	V	56
23	W	57
24	X	58
25	Y	59
26	Z	5A
27	[	5B
28	\	5C
29	]	5D

Addr.	ASCII	Hex
30	^	5E
31	-	5F
32	`	60
33	a	61
34	b	62
35	c	63
36	d	64
37	e	65
38	f	66
39	g	67
40	h	68
41	i	69
42	j	6A
43	k	6B
44	l	6C
45	m	6D
46	n	6E
47	o	6F
48	p	70
49	q	71
50	r	72
51	s	73
52	t	74
53	u	75
54	v	76
55	w	77
56	x	78
57	y	79
58	z	7A
59	{	7B

3.5 Parameter setting via download and settings in the .ini files

The various functions of the terminal mainly depend on the parameters you set. Parameter setting allows the terminal to be adapted to most user requirements. When setting a terminals parameter, the existing parameters are modified and the new values are stored in the terminals memory.

Parameter setting via download is done by transmitting data records from a computer to the terminal. To do so, the so-called download records are transmitted to the terminal. The following chapters explain the type of records you can download. After you cold start a terminal in service mode, all parameters are reset to their default values (please refer to the terminal documentation regarding service mode). The parameters in the .ini files can be transmitted to the host via SFTP and be modified then.

Info	Topic
14	Compilation of the .ini-files

3.5.1 Hexadecimal interpretation of characters

Some download records offer the possibility to select parameters by entry of hexcoded values. These records are explained precisely with separate tables, in which the possible values are defined. The bits marked with "Reserved" have no function.

Example: The parameters of bit 2 and 3 are to be activated (set to 1)

8	4	2	1	Hex valency
3	2	1	0	Bit
1	1	0	0	Binary value

Binary writing of this value: 1 1 0 0

Decimal conversion: $1x8 + 1x4 + 0x2 + 0x1 = 12_{dec}$

Hex chart

Hex value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Value that is inscribed in the data record: C (as an ASCII character).

This bit coding means: bit 2 and bit 3 are set, i.e. the functions defined here are activated.

3.6 Basic data record structure

The ">3E01" download record shown below explains the data record structure. This data record switches the terminal from standard time to daylight saving time and vice versa. With this data record, you can indicate whether data records generated by the terminal will include seconds in their time format or not. The "Example" column in the table below shows typical values used in this data record.

All entries in the download records are ASCII characters.

A more detailed explanation of this data record can be found in Download record Daylight saving time switchover.

Abbreviation	Meaning	Example
GID	Group address	@
DID	Device address	A
ON/OFF/AUTO	Operating mode	0
RT	Record type	>
RTM	Record type modification	3
BI	Block identification	E
BI		0
BI		1
M	Month of switchover	0
M		3
D	Day of switchover	2
D		9
RES	Reserved	0
0	Seconds in data record	0
h	Switchover time, hours	-
h		-
m	Switchover time, minutes	-
m		-
+	Direction of switchover	+
h	Offset of switchover, hours	0
h		0
m	Offset of switchover, minutes	0
m		0

3.6.1 Explanation of abbreviations

The following abbreviations are used in the descriptions of the data records.

GID (Group address) and DID (Device address)
The group- and device identification represent the address of the device. Possible group and device addresses are shown in chapter [▶ 3.4].

ON/OFF/AUTO
Determines the current operating status of the terminal. Chapter [▶ 3.3] shows the predefined characters of the operating status. In data records transmitted from the computer to the terminal, this digit may be filled with a blank.

RT, RTM
Record type and record type modification. They specify the data records. Chapter [▶ 13.1.1] and chapter [▶ 13.1.2] show the possible record type/record type modifications in download- and upload records.

BI
Block identification. It specifies the type of up- or download records. Chapter [▶ 13.6] shows the possible block identifications.

CI
Clock identification. The clock identification indicates the current status of the time in the terminal. The table in chapter [▶ 13.4] shows the defined characters.

EI
Error identification. If an error occurs during the verification of the recording data, this will be entered into the data record as an error identification. If several errors are detected, the error with the highest priority is entered. Chapter [▶ 13.5] shows the possible error identifications.

YY/MM/DD
Date indication: Year/Month/Day.

hh/mm/ss
Time indication: hours/minutes and possibly seconds. After a cold start of the terminal, the time indication consists of hours and minutes. To include seconds, this option must be set using the download record ">3E01", see chapter [▶ 3.9.8.1].

ll/cc/vv
Tag (6 characters) for user language: language/country/variant. After a cold start of the terminal, the tag is not entered in the data record. The entry of the user language identification must set using a download record, see chapter [▶ 3.9.8.2].

RES
Reserved: the download record must be filled with the specified characters.

3.7 File transfer

General

For easy and efficient data transfer, an SFTP server is integrated in the terminal. This feature is automatically loaded when switching on the terminal. This is particularly efficient when large quantities of data must be distributed to a large number of terminals via SFTP.

The terminal contains several .ini files whose setting can also be adjusted by means of access via SFTP.

B-Client is able to interpret ASCII or Unicode Big endian characters. In data- or parameter records, only ASCII characters are transmitted.

The "text.ini" file and parameter files must be encoded in Unicode Big Endian with the appropriate BOM (byte order mark).

The file transfer is a parameter setting option that avoids direct communication with the terminal. The parameters are stored in a file in the described format and then transferred to the terminal. If the application runs on the terminal, it will read the parameters from the file.

Files are only transferred from the host to the terminal.

All files needed for data transfer are stored in the /data/data/com.kaba.apps.hr/files/ transfer directory. The transferring method uses only two files: a data file ("record.zip" or "record.dat") and a control file ("input.cmd"). If the data file is present in both formats, only the zipped version is used. The unzipped file is not considered in this case.

The data file contains the parameter records and the master records. The data records begin with the GID/DID and are separated by a carriage return character. The files generated by B-COMM software can be used if you delete the first line which contains the program number.

3.7.1 Procedure from the host computer's point of view

First, the host has to check the existence of the control file. This is the case if either the last data transfer is not finalized or a problem occurred while reading the data file. If no control file exists, first the data file must be transferred and after that the control file. If done, the data transfer is finished for the host´s point of view.

To review the success of the data transfer, the deleted control file "input.cmd" indicates, that all records were read. In this case, all data records were processed successfully.

3.7.2 Procedure from the terminal's point of view

If the application runs, the terminal checks periodically (every 10 seconds) the existence of the data file. If it exists, the data file is read and all data records are processed one after another. The processing takes place in the same way as if the data records were received via the normal communication channel.

After processing the last record, the data file will be deleted.

In the last step, the control file is deleted to provide a success message. The terminal is now ready for the next transfer.

3.8 Remote control using command records

The following command record enables the terminal to carry out some special functions.

- TARESET
- TARESET COLD
- TAREBOOT
- TAWEBSERVER START
- TAWEBSERVER STOP
- TASSH START
- TASSH STOP

The command to be executed must be written directly into the command record.

GID DID ON/OFF	Group address Device address Operating mode
T A R E S E T	Command, see following table.

Command	Description
TARESET	After download, the terminal application will be shut down and afterwards restarted. The defined parameters remain unchanged.
TARESET COLD	After download, the terminal carries out a cold start.
TAREBOOT	After download, the operating system will be shut down and afterwards restarted. After the restart, the terminal application will be started.
TAWEBSERVER START	Starts the web server service. If the start is currently impossible (server yet started or in initial phase) this will be reported to the host by an error record (O2: Command cannot be executed at the moment)
TAWEBSERVER STOP	Stops the web server service. If the stop is currently impossible (server yet stopped or in initial phase) this will be reported to the host by an error record (O2: Command cannot be executed at the moment)
TASSH START	Starts the SSH server.
TASSH STOP	Stops the SSH server

3.9 General terminal definitions

3.9.1 Buffer Backup

Data records correctly transmitted from the terminal to the computer are deleted in the buffer memory of the terminal. If "Buffer Backup" is set, transmitted recording data records are not deleted and remain in the terminals buffer memory. The data records in the buffer memory can only be requested again or deleted with special data records.

Standard factory setting
After transmission, the data records will be deleted in the terminal. No buffer backup.

Info	Topic
5.2.11	Control buffer memory
3.9.7	Necessary download record

3.9.2 Fix record length for badge data

A recording data record may contain badge as well as keyboard data. This data may have different length.

The parameter defines that the badge data which is written into the recording data record has the same length each time. That means, the parameter states how many digits are entered into the recording data record by cutting off the last digits or filling them with blanks (20hex), if necessary. If two badge inputs are set by parameter setting, cutting or filling the badge data is made after the data of both badges are written one after the other.

1. Example:
If 13 digits are read from the badge and the parameter fix record length is set to 20, the digits 14 to 20 are filled with blanks (20hex).

Fix record length for badge data with 20 digits																		
1	2	3	4	5	6	7	8	9	0	1	2	3						
13 digits badge data													7 digits attached spaces					

2. Example:
If 13 digits are read from the badge and the parameter fix record length is set to 10, the digits 11 to 13 are cut off.

Fix record length for badge data with 10 digits										3 digits cut off		
1	2	3	4	5	6	7	8	9	0	11	12	13
13 digits badge data												

Standard factory setting
No fix record length for badge data.

Info	Topic
3.9.7	Necessary download record

3.9.3 Fix record length for keyboard data and lists

A recording data record may contain badge and keyboard data or list data respectively. This data may have different length.

The parameter defines that the keyboard data / list data resp. which is written into the recording data record has the same length each time. That means, the parameter states how many digits are entered into the recording data record by cutting off the last digits or filling them with blanks (20hex), if necessary. If two or three keyboard / list inputs are defined, the cutting or filling is made after all the data is written one after the other.

Standard factory setting
No fix record length for keyboard data or list data.

Info	Topic
3.9.7	Necessary download record

3.9.4 Logical group identification

The logical group identification is a part of the terminals address. Allowed values: 00 - 29.

Standard factory setting
The logical group address is 00.

Info	Topic
3.4	Table of group identifications
3.9.7	Necessary download record

3.9.5 Logical device identification

The logical device identification is a part of the terminals address. Allowed values: 00 - 59.

Standard factory setting
The logical device identification is 00.

Info	Topic
3.4	Table of device identifications
3.9.7	Necessary download record

3.9.6 Response time out

The parameter supervises the arriving of a logical booking response after the terminal has transmitted recording data to the computer. The terminal awaits a valid response within the response timeout. During this time the message "Bitte warten" ("Please wait") is displayed.

If the terminal does not receive any response it switches over to the offline operating mode. The response timeout is as four times long as the registered value.

Standard factory setting
The registered value is 3.0 seconds, the response timeout 12.0 seconds.

Info	Topic
3.9.7	Necessary download record

3.9.7 Download record Buffer backup and logical addresses

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the standard factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
S 0 1	Block identification
0	Buffer backup 0 = No; 1 = Yes
0	Reserved 0
— —	Fix record length for badge data — — = None; 00-99
— —	Fix record length for keyboard data/list data — — = None; 00-99
0 0	Logical group address 00-29
0 0	Logical device address 00-59
0 3 0	Response time-out 000-99.9 seconds (030x0.1 seconds = 3.0 seconds)
0 .. 0	9 digits reserved: 0

3.9.8 Daylight saving time switchover

This parameter defines how and at what time a switchover shall take place between two time periods (daylight saving time). The parameter consists of three blocks:

- day of switchover
- time of switchover
- offset of switchover with its direction (forward or backward).

You can choose between

- automatic switchover with fixed time rules or
- adjustable switchover with adjustable time rules.

If a switchover time is **not** defined, the automatic switchover is activated automatically.

Adjustable switchover
The terminal switches to new period at the defined day and time with the defined offset and direction.

Automatic switchover
If a date of the first half of the year is entered, the terminal automatically changes from standard time to daylight saving time at 2:00 am on that morning (2:00 am to 3:00 am). If a date of the second half of the year is entered, the terminal automatically changes from daylight saving time to standard time at 3:00 am on that morning (3:00 am to 2:00 am).

After the time switchover has taken place, the entered values are deleted in the terminal. For another switchover, the parameters must be entered once again.

If the parameter daylight saving time switchover is used, the switchover in the terminals operating system must absolutely be set to off (factory setting)!

Because it is not sure that the time server and the terminal change at the same time, do not transmit any date/time record in this switchover period! Note: If the time server is located in another time zone (maybe even without daylight saving time switchover) this must be taken into account when transmitting a date/time record!

Standard factory setting
No switchover is set.

Info	Topic
5.4	Date/time setting
3.9.8.3	Necessary download record

3.9.8.1 Seconds in the data record

If this parameter is set, the data records generated by the terminal and transmitted to the computer which contain hours and minutes as time indication, will also contain the seconds. The data records becomes two digits longer and the clock identification in the data record changes.

Standard factory setting
No seconds in the data record.

Info	Topic
13.4	Table of clock identifications
3.9.8.3	Necessary download record

3.9.8.2 User language in data record

If this parameter is set, the data records generated by the terminal and transmitted to the computer, will also contain a tag for user language after the time indication. The data records therefore become six digits longer and the clock identification in the data record changes.

Standard factory setting
No tag for user language in the data records.

Info	Topic
13.4	Table of clock identifications
3.9.8.3	Necessary download record

3.9.8.3 Download record Daylight saving time switchover

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
E 0 1	Block identification
— — — —	Switchover date From the top: Month, month, day, day. — — — — = no switchover.
0	Reserved 0
0	Seconds and user language; see following description.
— — — —	Switchover time From the top: Hour, hour, minute, minute. — — — — = automatic switchover.
+	Switchover direction (+ or -)
0 0 0 0	Switchover offset From the top: Hour, hour, minute, minute.

Seconds and user language

The terminal interprets this character entered in the download record as a hex code which contains several parameters.

8	4	2	1	Hex value

3	2	1	0	Bit
				Seconds in data record 0 = No 1 = Yes
				User language in data record 0 = No 1 = Yes
				Reserved 0
				Reserved 0

Info	Topic
3.5.1	Hexadecimal coding

3.9.9 Display duration "Authorized"

The parameter defines the display duration of messages for an **authorized** booking.

Standard factory setting
Display duration 5.0 seconds.

Info	Topic
3.9.14	Necessary download record

3.9.10 Display duration "Not authorized"

The parameter defines the display duration of messages for a **not authorized** booking.

Standard factory setting
Display duration 5.0 seconds.

Info	Topic
3.9.14	Necessary download record

3.9.11 Relay operation time

The parameter defines the operation time of the relays.

The relay can be activated either after pressing a function button or via special data records. Depending on the desired use case, the relay can be assigned via the sections [IOMapping] and [OutputFunctions] in the "application.ini" file.

Standard factory setting
The operation time of the relays is 3.0 seconds.

Info	Topic
3.10.14	Relay operation via function key
5.2.2	Relay operation via download records
3.9.14	Necessary download record
14.3.1	File "application.ini"

3.9.12 Operation timeout

The parameter defines how long the terminal waits for a badge or keyboard input after a function key has been touched. If no input is made within this time, the terminal returns to its base state.

Standard factory setting
Operation timeout 12.0 seconds.

Info	Topic
3.9.14	Necessary download record

3.9.13 Display duration "Display info"

The parameter defines the display duration for authorized bookings with display info.

Standard factory setting
The value for the display duration is 000. This means, the "Display info duration" is equal like the display duration "Authorized".

Info	Topic
3.9.9	Display duration "Authorized"
3.9.14	Necessary download record

3.9.14 Download record Durations

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID	Group address
DID	Device address
ON/OFF	Operating mode
>	Record type
3	Record type modification
X	Block identification
0	
1	
0	Display duration "Authorized"
5	000-99.9 seconds
0	(050x0.1 seconds = 5.0 seconds)
0	Reserved 0
0	Display duration "Not authorized"
5	000-99.9 seconds
0	(050x0.1 seconds = 5.0 seconds)
0	Reserved 0
0	Operation time for relay
3	000-99.9 seconds
0	(030x0.1 seconds = 3.0 seconds)
0	Reserved 0
1	Operation time-out
2	000-99.9 seconds
0	(120x0.1 seconds = 12.0 seconds)
0	Reserved 0
0	Display duration "Display-info"
0	000-99.9 seconds
0	(000 = like "Display duration authorized")
0	Reserved 0

3.9.15 Password for parameter setting

This parameter is no longer evaluated.

3.9.16 Customer number

The customer number is a check criterion for access control.

If a customer number has been defined via the badge definitions and a customer no. is stored in the terminal, this customer number is checked with every badge input. The parameter is used to store a customer number in the terminal. Digits of the customer number which shall not be checked must be filled up with the hyphen character "—". If all 8 digits are filled up with the hyphen character, the customer number check is completely switched off.

Example:

Customer number stored in the terminal	1	2	3	—	—	6
Customer number read from the badge	1	2	3	4	5	6

The first three digits and the last digit are checked.

If the customer number read from the badge is not identical with the stored customer number or with any stored additional customer number, the error identification 2 "Customer no. error" is entered into the recording data record. In the operating modes offline/autonomous the booking is rejected with the message "Kunden-Nr. Fehler" (Customer no. error).

Standard factory setting
No customer number is stored in the terminal.

Info	Topic
3.9.23	Additional customer number
3.10.12	Badge input
3.9.17	Necessary download record

3.9.17 Download record Password setting and customer number

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
X 0 2	Block identification
— — — —	These 4 digits are not evaluated.
— — — —	4 digits reserved —
— — — — — — — —	8 digits customer number 00000000-99999999 and A to F — = digit is not checked .
— .. —	8 digits reserved —

3.9.18 Additional password for parameter setting

The password definition for local parameter setting is mandatory. This is done at the 1st program start via the user interface of the terminal. The password can also be set or changed later via this parameter record.

The password must meet the following conditions or allow the following characters:

- Length: min 10 characters, max. 16 characters
- At least 1 lowercase letter a-z
- At least 1 uppercase letter A-Z
- At least 1 cipher 0-9
- At least one of the following special characters: **! " ( ) [ ] \ @ # $ % / ^ § & + =**
- No whitespace chacters, i.e. tabs or spaces
- **NOT permitted** for technical reasons: the characters * and **-**

The definitions are checked internally. If the conditions are not met, the password is rejected and an O1 record is generated.

Local parameter setting will be locked after entering the wrong password 3 times. Unlocking is only possible using an I2 data record; see chapter [▶ 5.2.1].

It is recommended to perform a parameter upload after setting the password so that the currently valid password is stored in the host system.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
X 1 2	Block identification
— ... —	Min. 10 to max. 16 digits additional password for parameter setting. Possible values: see description above.

Info	Topic
7.2	Error records

3.9.19 Code control byte

DC

This parameter is only important if the employee`s master record

- does not refer to a time profile and
- a keypad input is set as a parameter and
- the data type **code** is set as a parameter.

The code control byte defines the interpretation of the entered data and the checks that are to be made.

No check (Code control byte = 0)
The entered keyboard data will not be checked. If a keyboard input is set as a parameter and the data type code is defined, the keyboard input will be cancelled.

PIN check (Code control byte = 1)
The entered keyboard data will be compared with the PIN (personal identification number) stored in the master record. If the PIN differs, the error identification : "PIN wrong or missing" is entered into the recording data record. In the operating modes offline/autonomous the booking is rejected with the message "Code falsch" (Code false).

Standard factory setting
The entered keyboard data will be compared with the PIN (personal identification number) stored in the master record.

Info	Topic
3.10.15	Keypad input
3.17.4	PIN
3.10.17	Data type "Code"
13.5	Table of error identifiers

3.9.20 Download record Code control byte

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
X 0 4	Block identification
1	Code control byte (0 or 1)
0	Reserved 0

3.9.21 Character set identification

For compatibility reasons concerning the Bedas series, the character set identification can be set like before (transmission of 7-bit characters). If the feature multilingualism will be used, the character set 000 USA ASCII must be set (transmission of 8-bit characters).

The terminal can use different character sets. The character set identification defines which one is used.

The table of character sets states the characters that can be displayed.

Standard factory setting
The character set 000 (USA ASCII) is set.

Info	Topic
13.9	Table of character set identifiers

3.9.22 Download record Character set identification

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
X 0 7	Block identification
0	Reserved 0
n n n	3 digits character set identification

3.9.23 Additional customer number

The additional customer number is a check criterion.

The additional customer number will only be checked, if

- a customer number is defined with the badge definitions and
- a customer number is stored in the terminal.

Up to 15 additional customer numbers can be stored in the terminal.

The parameter is used to store up to 15 additional customer numbers in the terminal. Digits of the additional customer number which shall not be checked must be filled up with the hyphen character "—". If all 8 digits are filled up with the hyphen character, the check of the additional customer number it completely switched off.

Example:

Customer number stored in the terminal	1	2	3	—	—	6
Customer number read from the badge	1	2	3	4	5	6

The first three digits and the last digit are checked.

If the customer number read from the badge is not identical with any stored additional customer number or with the stored customer number, the error identification 2 "Customer no. error" is entered into the recording data record. In the operating modes offline/ autonomous, the booking is rejected with the message "Kunden-Nr. Fehler" (Customer number error).

Standard factory setting
No additional customer numbers are stored in the terminal.

Info	Topic
3.9.16	Customer number
3.9.24	Necessary download record

3.9.24 Download record Additional customer number

The parameters described in the previous chapters are modified in the terminal with the following download record. The example shows the factory settings.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
X 0 8	Block identification
— — — — — — — —	8 digits additional customer number 00000000-99999999 and A to F — = digit is not checked
— .. —	Up to 14 more additional customer numbers.

3.10 Function key functions

The parameters in this parameter group are used to define the booking procedure when using function keys.

3.10.1 Function key memorizing

Normal case: Before you can book at a terminal you must operate a certain function key. After the booking, the terminal returns to its previous state. To use the same function key again, it must be operated again. The "Function key memorizing" parameter allows you to book again without operating the function key again, because the function key remains active until another function key is operated.

Standard factory setting

Function key	Function key memorizing
F0 (base state)	Yes
F1	No
F2	No
F3	No
F4	No
F5	No
F31	No
F32	No
F33	No
F34	No
F35	No
F36	No
F37	No
F38	No
F39	No
F40	No

Info	Topic
3.10.34	Necessary download record

3.10.2 Terminal awaits logical booking response in offline/autonomous mode

This parameter defines if the terminal awaits a logical booking response from the computer also in operating mode offline and autonomous. If no logical booking response arrives in the terminal within the defined response time out (default: = 4x response time out =12 seconds) the internal booking response is displayed.

Standard factory setting
For all function keys the terminal does not await a logical booking response from the computer in operating mode offline and autonomous.

Info	Topic
3.10.34	Necessary download record

3.10.3 Check on time authorization

DC

This parameter is important if the employee´s master record refers to a time profile. The parameter defines, whether the parameter "Time authorization" in the time profile will be regarded.

Standard factory setting
For all function keys, the parameter "Check on time authorization" is **not** regarded in the time profile.

Info	Topic
3.17	Time profiles
3.10.34	Necessary download record

3.10.4 Individual master record required

The parameter defines that an individual master record must be stored in the terminal for each ID number which is read from a badge. If the parameter "Individual master record required" is set, but no individual master record is available for the ID number which is read from a badge, the error identification 4 "No master record found" is entered into the recording data record. In the operating modes offline/autonomous the booking is rejected with the message "Kein Stammsatz" (No master record).

Standard factory setting
For **no** function key an individual master record is required.

Info	Topic
3.10.34	Necessary download record
13.5	Table of error identifiers

3.10.5 Bookings allowed in online mode

The parameter defines whether bookings are allowed in operating mode online. If bookings are not allowed in online/offline mode, the function key is not displayed on the screen.

Standard factory setting

Function key	Bookings allowed in online mode
F1	Yes
F2	Yes
F3	Yes
F4	Yes
F5	Yes
F31	No
F32	No
F33	No
F34	No
F35	No
F36	No
F37	No
F38	No
F39	No
F40	No

Info	Topic
3.10.34	Necessary download record

3.10.6 One/two badge inputs required

This parameter is only important if a badge input is required.

The parameter defines if one or two badge inputs are required for a booking. If two badge inputs are set as a parameter, a separator block must absolutely be defined via the badge definitions.

Standard factory setting
For **all** function keys one badge input is **required**.

Info	Topic
3.14	Badge definitions
3.10.12	Badge input required
3.10.34	Necessary download record

3.10.7 Bookings allowed in offline/ autonomous operation mode

The parameter defines if bookings are allowed in the offline/autonomous operating mode. If bookings are not allowed in the offline operating mode, the function key is not displayed in the touch screen.

Standard factory setting

Function key	Bookings allowed in offline/ autonomous operation mode
F1	Yes
F2	Yes
F3	Yes
F4	No
F5	Yes
F31	No
F32	No
F33	No
F34	No
F35	No
F36	No
F37	No
F38	No
F39	No
F40	No

Info	Topic
3.10.34	Necessary download record

3.10.8 Bookings allowed despite of a full buffer

The parameter defines if bookings are allowed if the buffer memory is full. Even with the parameter "Buffer Backup" set, recording data records are no longer stored in the terminal if the buffer memory is full. The access control functions are still available.

Standard factory setting

For **no** function key bookings are allowed if the buffer memory is full.

Info	Topic
3.10.34	Necessary download record

3.10.9 Suppress display info in master record

The parameter defines if the display info in the master record is suppressed. If the parameter is set, the display info is not displayed during a booking.

Standard factory setting
For **no** function key the display info is suppressed

Info	Topic
3.10.34	Necessary download record

3.10.10 Buffer authorized recording data

This parameter is mainly important in the "offline" or "autonomous" operating mode and also valid in the "online" operating mode if the "Buffer backup" parameter is set. The parameter defines if authorized recording data is buffered or not.

Standard factory setting

Function key	Buffer authorized recording data
F1	Yes
F2	Yes
F3	Yes
F4	No
F5	Yes
F31	Yes
F32	Yes
F33	Yes
F34	No
F35	Yes
F36	Yes
F37	Yes
F38	Yes
F39	Yes
F40	Yes

Info	Topic
3.10.34	Necessary download record

3.10.11 Buffer unauthorized recording data

This parameter is mainly important in the "offline" or "autonomous" operating mode and also valid in the "online" operating mode if the "Buffer backup" parameter is set.

The parameter defines if unauthorized recording data shall be buffered.

Standard factory setting
For all function keys: unauthorized recording data is not buffered.

Info	Topic
3.10.34	Necessary download record

3.10.12 Badge input required

The parameter defines if a badge input is required for a booking.

Standard factory setting
For all function keys a badge input is required.

Info	Topic
3.10.34	Necessary download record

3.10.13 Double access blocking

DC

This parameter is only important if an ID number is read from a badge or a smartphone.

The parameter defines if double access blocking is set.

If double access blocking is set, no further booking made with the same function key and the same badge or smartphone is authorized within a time slot of 6 minutes, e.g. after badge/ smartphone handover to an unauthorized person. If a double access has been attempted the error identification @ "Attempted double access" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Doppelzutrittsperre" (Double access blocking).

Up to 50 ID numbers can be managed. If double accesses occur more than 50 times, double access blocking will be quit for the ID number with the shortest time remaining before the 6 minutes will be expired.

Standard factory setting
For **no** function key a double access blocking is set.

Info	Topic
3.14	Badge definitions
3.10.34	Necessary download record

3.10.14 Activate relay

DC

The parameter defines if the door opener relay is activated after an authorized booking. Depending on the desired use case, the relay can be assigned via the sections [IOMapping] and [OutputFunctions] in the "application.ini" file.

Standard factory setting

Function key	RT/RTM	Activate relay
F0 (base state)	A0	Yes
F1	B1	No
F2	B2	No
F3	B3	No
F4	B0	No
F5	F0	No
F31	B1	No
F32	B2	No
F33	B4	No
F34	BS	No
F35	BE	No
F36	FA	No
F37	FB	No
F38	FC	No
F39	FD	No
F40	FE	No

Info	Topic
3.10.25	Record type of the function key
13.1	Table of RT/RTM
3.10.34	Necessary download record
14.3.1	File "application.ini"

3.10.15 Keyboard input

If data shall be entered via a keyboard, several parameters are to be defined prior. These are:

- Number of digits of the input (= activation) chapter [▶ 3.10.16]
- Kind of data type (code or data) chapter [▶ 3.10.17]
- Keyboard input rolling or fix chapter [▶ 3.10.18]
- Numeric or alphanumeric data chapter [▶ 3.10.19]
- Dialog text shown during the input procedure chapter [▶ 3.10.21]
- List numbers used for data input via lists chapter [▶ 3.10.22]

In a default booking procedure, max. 3 keyboard inputs may be requested. This is defined by setting the corresponding operating modes via parameters. For the use of keyboard inputs, the following definitions are valid:

1. keyboard input	Operating mode 3
2. keyboard input	Operating mode 4
3. keyboard input	Operating mode 5

The above mentioned parameters must be defined for each of these keyboard inputs. Some of the parameters thereby are equal for all 3 keyboard inputs, some of them are only partly valid. This will be stated at the parameter description in detail.

The definitions of the operation mode to be used take place via a download record. A parameter setting example will be given with the download record description.

Info	Topic
3.10.34	Necessary download record

3.10.16 Number of digits for keyboard input

This parameter is valid for all 3 keyboard inputs equally.
The parameter defines a keyboard input and how many digits to be entered. Up to 24 digits can be set. The **setting 00 deactivates** the keyboard input.

The "Number of digits for keyboard input" parameter defines the number of digits which are written into the recording data record, independently from the way of input (manually or via a list).

Standard factory setting
For no function key a keyboard or list input is set.

Info	Topic
3.10.34	Necessary download record

3.10.17 Type data/code 1 and type data/code 2

DC

This parameter is valid for the 1. keyboard input only.
This parameter is used to define the input procedure and the data type for the 1. keyboard input.

Possible settings

1. Keyboard input type "data" after badge input

Features

- A keyboard input **must** be made.
- The input takes place **after** the badge input.
- The operated character is displayed on the screen.
- When all characters have been entered, the keyboard input is confirmed via the key "OK".
- The keyboard input is **not** checked.
- The keyboard input is entered into the generated recording data record.

2. Keyboard input type "code" after badge input

Features

- A keyboard input **must** be made
- The input takes place **after** the badge input.
- The operated character is **not** displayed on the screen. For each input digit the wildcard character (*) is displayed as a placeholder.
- When all characters have been entered, the keyboard input is confirmed via the key "OK".
- The data is checked to the definitions of the parameters "Code control byte" or "Identification bytes".
- The keyboard input is entered into the generated recording data record.

3. Keyboard input type "code" possible before badge input

Features

- A keyboard input **may** be made
- The input takes place **before** the badge input.
- The operated character is **not** displayed on the screen. For each input digit the wildcard character (*) is displayed as a placeholder.
- When all characters have been entered, the keyboard input is confirmed via a badge input (booking).
- If no badge input has been set via parameter setting, the keyboard input is confirmed via the key "OK".
- The data is checked to the definitions of the parameters "Code control byte" or "Identification bytes".
- The keyboard input is entered into the generated recording data record.

4. Keyboard input type "code" before or after badge input

Features

- A keyboard input **must** be made
- The input takes place **before** or **alternatively after** the badge input.
- The operated character is **not** displayed on the screen. For each input digit the wildcard character (*) is displayed as a placeholder.
- If the keyboard input is made first the confirmation is the following badge input.
- If the badge input is made first the following keyboard input is confirmed via the key "OK".
- The data is checked to the definitions of the parameters "Code control byte" or "Identification bytes".
- The keyboard input is entered into the generated recording data record.

Standard factory setting
For all function keys the keyboard input type data after badge input is set.

Info	Topic
3.17.4	Time profiles: Identification bytes
3.9.19	Code control byte
3.10.34	Necessary download record

3.10.18 Keyboard input format

This parameter is valid for all 3 keyboard inputs equally
The parameter determines whether keyboard input must be rolling or fixed.

Rolling
Users can enter as many digits as defined by the "Number of digits for keyboard input" parameter. If fewer digits are entered, dummy characters are inserted. The dummy characters are zeros in the case of numeric characters (30hex) and blank characters (20hex) in the case of alpha-numeric characters.

Fixed
Users can enter as many digits as defined by the "Number of digits for keyboard input" parameter. If fewer digits are entered, the booking is rejected with an error message.

Info	Topic
3.10.15	Keyboard input
3.10.34	Necessary download record

3.10.19 Keyboard data numeric/alphanumeric

This parameter is valid for all 3 keyboard inputs equally.
The parameter defines if numeric or alphanumeric keyboard data are allowed for the keyboard input.

- **Numeric**

  For this kind of input, numeric data (numbers) is permitted only.

- **Alphanumeric**

  For this kind of input the following data is permitted:

  - numbers

  - letters a-z and A-Z

  - special characters : ; = < > ?

Standard factory setting
Numeric input

Info	Topic
3.10.15	Keyboard input
3.10.34	Necessary download record

3.10.20 Dialog text number for 1. keyboard input

This parameter is valid for the 1. keyboard input only.
The parameter defines a dialog text for the 1. badge input. 4 dialog texts are available (D00 to D03). The dialog texts can be modified via a parameter download. If the terminal is in operation, the dialog text for 1. keyboard input is displayed after the badge input.

Standard factory setting
For all function keys the dialog text D01 "Code eingeben" (Enter code) is set for the 1. keyboard input.

Info	Topic
3.19.7	Download record "Texts"
3.10.15	Keyboard input
3.10.34	Necessary download record

3.10.21 Dialog text number for 2. & 3. keyboard input

This parameter is valid for the 2. & 3. keyboard input only.
The parameter defines a dialog text for the 2. & 3. keyboard input which is shown on the display. Sixteen dialog texts are available (D00 to D15). The dialog texts can be modified via a download record.

Standard factory setting
The preset definitions for the 2. and 3. keyboard input must be adapted in conformity of the use case.

For all function keys the dialog text D00 "Ausweis bitte" (Present the badge) is fix set for a 2. keyboard input.

For all function keys the dialog text D00 "Ausweis bitte" (Present the badge) is fix set for a 3. keyboard input.

Info	Topic
3.19.7	Download record "Texts"
3.10.15	Keyboard input
3.10.34	Necessary download record

3.10.22 List number for keyboard input

This parameter is valid for all 3 keyboard inputs equally.
This parameter is only important if a keyboard input is. Only if a keyboard input is set, the list number is checked. If a list number is stored, the list will be displayed on the screen.

The parameter defines, that the keyboard input must not explicitly be entered but a list can be displayed on the touch screen. The booking employee can select an entry from the list. 20 lists (list numbers 01 to 20) are available. The contents of the lists can be modified via download.

The entry 00 as a list number means, that no list shall be displayed.

Standard factory setting
For all function keys **no** list is displayed with a keyboard input.

Info	Topic
3.11	Lists
3.10.15	Keyboard input
3.10.34	Necessary download record

3.10.23 Dialog text number for 1. badge input

The parameter defines a dialog text for the 1. badge input. Four dialog texts are available (D00 to D03). The dialog texts can be modified via a parameter download. If the terminal is in operation, the dialog text is displayed on the touch screen.

Standard factory setting

For all function keys the dialog text D00 "Ausweis bitte" ("Present the badge") is displayed.

Info	Topic
3.19.7	Download record "Texts"
3.10.34	Necessary download record

3.10.24 Dialog text number for 2. badge input

The dialog text number for the 2. badge input cannot be modified. The parameter is set fix but for complete information it is mentioned here.

For the 2. badge input the dialog text is set fix. The dialog text which is defined by the dialog text number is displayed on the touch screen, if a 2. badge input is defined.

Standard factory setting
For all function keys the dialog text D02 "Ausweis bitte" (Present the badge) is set fix for a 2. badge input.

Info	Topic
3.19.7	Download record "Texts"
3.10.34	Necessary download record

3.10.25 Record type / Record type modification

The parameter defines the record type and record type modification contained in a recording data record transmitted to the computer.

Standard factory setting

Function key	RT/RTM
F0 (base state)	A0
F1	B1
F2	B2
F3	B3
F4	B0
F5	F0
F31	B1
F32	B2
F33	B4
F34	BS
F35	BE
F36	FA
F37	FB
F38	FC
F39	FD
F40	FE

Info	Topic
3.10.34	Necessary download record

3.10.26 Reader number for 1. badge input

For all function keys the same reader must be defined for 1. badge input because of the common badge definitions.

This parameter defines the reader to be used for the 1. badge input.

Standard factory setting
For all function keys reader 1 is used for 1. badge input.

Info	Topic
3.14	Badge definition
3.10.12 and 3.10.6	Badge input required (1. & 2.)
3.10.34	Necessary download record

3.10.27 Reader number for 2. badge input

For all function keys the same reader must be defined for 2. badge input because of the common badge definitions.

This parameter defines the reader to be used for the 2. badge input.

Standard factory setting
For all function keys reader 1 is used for 2. badge input.

Info	Topic
3.14	Badge definition
3.10.12 and 3.10.6	Badge input required (1. & 2.)
3.10.34	Necessary download record

3.10.28 Browser

The parameter defines if the browser is started additionally after the settings for the function key have been processed. If so, the browser starts with a start page whose name could be added with parts of the generated recording data record.

If a partner application is parameterized, the following succession is valid:

1 Function key function

2 Browser

3 Partner application

To start the browser, the name of the start page must absolutely be entered in the parameter KeyXXURL in the [FunctionKey] section of the application.ini file.

Standard factory setting
For all function keys **no** browser is started.

Info	Topic
14.3.1	File "application.ini", section [FunctionKeys]
3.9.12	Operation timeout
3.10.29	Partner application
3.10.34	Necessary download record

3.10.29 Calling external applications

The B-Client HR40 device software enables external applications to be called after the parameterized function key functions have been processed.

The sequence of processing is as follows:

1 function key functions

2 browser (if activated)

3 external application.

These external applications can be:

A: Partner applications with an interface defined by dormakaba. These partner applications enable the data exchange between the B-Client and the partner application. The use of a partner application requires the licensing of the "Partner Interface" option.

B: General applications, so-called Native Apps. Here, no data exchange takes place between the B-Client and the application. Only one call of the application is possible. The application must be terminated by the user himself. The use of a Native App requires the licensing of the "Native App" option.

Only one partner application or any app can be defined for a function key.

Depending on which type of application (A or B) is to be called, the parameterization to be carried out differs.

Calling a partner application

1 Define the desired function key for the call.

For this purpose, bit 0 is set to the value "1" for this function key by either the PEC in B-COMM or with the download record Function key functions in operating mode 6 (extensions) in byte 2.

2 Connection of the partner application.

A separate documentation "Partner Interface Connection" number 04044924 is available for connecting the partner application.

A sample implementation of a demo partner application is available in source code on request. This can serve as a basis for own developments.

Calling a native app

1 The desired Native App must be installed locally

2 **No partner application** is specified or a partner application cannot be executed (e.g. missing license).

3 Define desired function key for the call. For this purpose, bit 0 is set to the value "1" for this function key by either the PEC in B-COMM or with the download record Function key functions in operating mode 6 (extensions) in byte 2.

4 Specify the app to be called. To do so, assign the name of the app to be called to the parameter "KeyXApp" in the section [FunctionKeys] of the file "application.ini".

Example: Key1App=<appPackageName>

Associates the function key 1 with the Native App named <appPackageName>.

Default factory setting
No function key starts a partner application or native app.

Info	Topic
3.10.34	Necessary download record

Also see about this

- Security hints for 3rd party apps [▶ 15]

3.10.30 Function key as dialog key

The parameter defines dialog mode for the function key.

For more detailed description of the dialog function (G dialog) see references.

If a function key is set as a dialog key, all other parameters for this function key are irrelevant.

If a function key is defined as a dialog key and the dialog is not blocked, the terminal transmits a G8 record to the computer and is then waiting for a G0 record. If no G0 record is transmitted to the terminal, it will exit the dialog mode with the message "Kein Dialog möglich" (Dialog impossible) after the response timeout expired. The dialog mode will be exit explicitly when a G9 record is transmitted to the terminal instead of a G0 record.

The booking employee can exit the dialog mode by touching the key again which has started the dialog mode.

Standard factory setting
No function key is defined as a dialog key.

Info	Topic
5.2.1	Blocking/releasing the dialog
7.7.5	G0 record
7.7.6	G8 record
7.7.7	G9 record
3.10.34	Necessary download record

3.10.31 Function key function for additional input steps

Function of the parameter

B-Client HR40 offers the option of configuring a maximum of 20 additional input steps for simple data acquisition applications, for example.

The additional input steps are called up via one or more correspondingly parameterized function key(s).

The implementation of the additional input steps is done via a modified G-Dialog. While in the conventional G-Dialog the parameters are transmitted online from the host via a G0 parameter record, the parameters for the additional input steps are stored in a file on the terminal and are called from there.

The parameterizations are stored in a separate file in the terminal for each function key. One parameter record is used for each input block. The number of input blocks per function key is limited to 20.

For the use of additional input steps, the desired function key must **not** be parameterized as a dialog key. This is only necessary for the use case of the online G-dialog.

Simultaneous use of online G-dialog and additional input steps is not possible.

The processing of the function key parameterization takes place as before. The processing of the dialog records for the additional input steps takes place at the end of the function key processing.

After processing all records in the file, the collected data is appended to the data entry record.

The following parameterization steps are required for the use of additional input steps.

1 Carry out the function key parameterization for each desired function key, see download record "Function key functions", operating mode 7, byte 2.

2 Create a corresponding file for each desired function key and save it in the terminal, see the following chapter.

Standard factory setting

No function key is parameterized for additional input steps.

Info	Topic
7.7.5	G0 record
3.10.34	Necessary download record

3.10.31.1 Definition file for additional input steps

The parameterization of the desired function key(s) as well as the booking procedure for the additional input steps is done via correspondingly created definition files in the directory /data/data/com.kaba.apps.hr/files/init.
The user creates these definition files himself, e.g. with an editor, and then transfers them to the specified directory. The definition files for additional input steps must be coded in Unicode Big Endian with the corresponding BOM (Byte Order Mark).

Fix texts can also be defined for the additional input steps.

See the following parameter example 2.

For each function key to be used, a separate definition file is created according to the pattern

addInputSteps_fk_XX.ini

The placeholder "XX" is filled with the number of the desired function key.

Examples for the function keys 4 and 5:

- addInputSteps_fk_04.ini
- addInputSteps_fk_05.ini

The actual parameters are specified in sections.

Section [General]

In the [General] section, general definitions for the input process are set. These definitions are based on the dialog record G0 with slight deviations.

The **DialogX** key defines the dialog mask for a user input including a user information. The user information from the [General] section is used if no language-specific user information is configured. For language-specific parameterization see chapter [▶ 3.10.31.3].

A DialogX key is defined for each input step. The placeholder "X" is given the number of the desired input step. A maximum of 20 input steps per function key are possible. The numbering must be continuously ascending (Dialog1 to max. Dialog20).
Gaps are not permitted and lead to an abort of the parameter set read-in process.

Parameter example 1 (without fix texts):

addInputSteps_fk_04.ini

[General]

Dialog1 = G010000010000Please enter 1. error reason 1-9

Dialog2 = G020000010000Please enter 2. error reason 1-9 a-z

Dialog3 = G010000030000Please enter quantity 1-999

Dialog4 = G001018050000Please present the badge

Dialog5 = G013101050000Please scan the barcode.

Use of fixed texts in additional input steps

Fix texts are used when additional information is to be added to certain data in the data record. In this way, when evaluating a data record, a specific type of data can easily be identified by the detailed description in the fix text, e.g. as a malfunction message or similar.

Another possible use is to always prefix the captured data with a recurring value by means of the fixed text.

The specified fixed texts are transferred directly in the data record.

In the definition file in the section [General] a fix text with the syntax "DialogX =Fixtext: <fixtext to be entered>" is noted, see example.

Parameter example 2 (with fix texts):

addInputSteps_fk_04.ini

[General]

Dialog1 = Fixtext: 1. Error reason

Dialog2 = G010000010000Please enter 1. error reason 1-9

Dialog3 = Fixtext: 2. Error reason

Dialog4 = G020000010000Please enter 2. error reason 1-9 a-z

Dialog5 = Fixtext: Quantity

Dialog6 = G010000030000Please enter quantity 1-999

.....

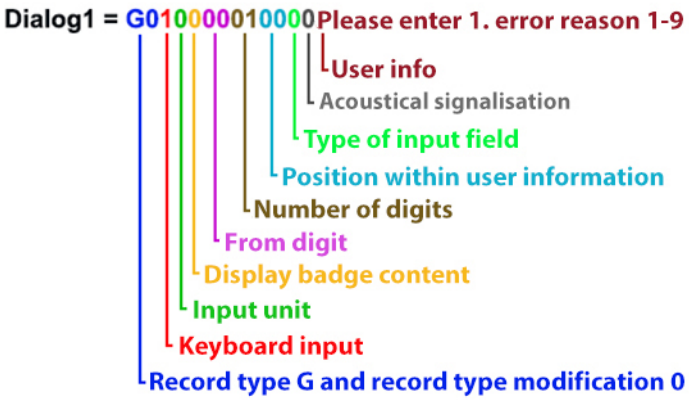
Resulting recording data record (excerpt)

GG F012303061650002000000000073 1. Error reason3 2. Error reasonh Quantity025....

Meaning of the entries in the recording data record

GG	Terminal GID and DID (07/07)
<Space character>	Identifier for "Online" operating mode
F0	RT/RTM for Function key F4
1	Identifier for "Transmit time without seconds"
230306	Date = March 6, 2023
1650	Recording time without seconds
0	Error identifier
0200000000073	Badge data according to function key functions
<Space character> 1. Error reason	1. Fix text with leading space character
3	Data input from Dialog 2
<Space character> 2. Error reason	2. Fix text with leading space character
h	Data input from Dialog 4
<Space character> Quantity	3. Fix text with leading space character
025	Data input from Dialog 6

The individual parameters of the definition have the following meaning (Example: Key Dialog1)



Parameter values

Keyboard input/list input

This entry defines if a keyboard- or list input is necessary.

Entry	Meaning
0	No keyboard input
1	Numeric keyboard input
2	Alphanumeric keyboard input
6	List input

Input unit

This entry defines if a reader input is necessary.

Entry	Meaning
0	No reader input
1	Reader 1
2	Reader 1
3	Reader 2

Display badge contents

This entry defines if the badge contents is displayed. The range limited by the entries "From digit with badge input" and "Number of digits" is displayed. The displayed data must be confirmed with the "OK" key.

Entry	Meaning
0	Do not display badge content
1	Display badge content

From digit with badge input

This entry defines from which digit the badge contents is entered into the G1 record. If the badge contents shall be entered from the first digit, from digit with badge input must be set to 01.

Number of digits

For a badge input, this entry is used to define how many digits (01 to 40) of the read badge contents have to be entered into the recording record.

For a keyboard input, this entry defines how many digits (01 to 40) must be entered and then written into the recording record. The entry 00 causes that the first key press is entered into the recording record and immediately transmitted to the computer.

Position within the user information used for default value

This parameter defines from which digit the characters of the user information will be entered as a default value into the input field. If no preset is desired, 00 can be parameterized.

Type of input field

Entry in the "keyboard input" field determines the type of the entry. This can be:

- via list selection (static or dynamic)
- via keyboard.

If the entry type "Lists" has been set, the desired list 1-15 can be selected here by entering 1- F.

If the entry type "Keyboard" has been set, the keyboard input can be specified here. At the beginning of the input, the cursor is positioned at the left. With each input, the cursor moves one digit to the right. When the input field is filled, the cursor is positioned at the rightmost digit.

If "Rolling keyboard input" has been set, the additionally entered characters are rolled through the display from right to left, see the following example a1. If fewer digits are entered as defined via "Number of digits",

- for numeric input: zeros or

- for alphanumeric input: blanks (20hex)

are entered into the G1 record as leading digits.

If "Fixed keyboard input" has been set, the cursor is also located on the left-hand side at the beginning of the input. With each input, the cursor moves one digit to the right. If other characters are entered, the right-hand character will be overwritten with the new character.

If fewer characters are entered than defined in the "Number of digits", an input error is displayed.

The following input options can be set:

Entry	Meaning
0	Rolling keyboard input; The characters entered are displayed
1	Fixed keyboard input; The characters entered are displayed
2 or 4	Rolling keyboard input; hidden input. The characters are displayed according to the Android setting in the parameter "Location & Security Settings: Visible Password".
3 or 5	Fixed keyboard input; hidden input. The characters are displayed according to the Android setting in the parameter "Location & Security Settings: Visible Password".

Acoustical signalization

This entry defines, if the user information is displayed with an additional acoustic signal.

Entry	Meaning
0	No acoustic signal
1	User information with alternating acoustic signal.
2	User information with constant acoustic signal.

User information

Text output up to max. 40 characters. This entry can be used for operator guidance texts, see examples.

3.10.31.2 Recording data

After processing the last input step, the captured data is entered into the recording record and transmitted to the host.

The total length of an recording record is determined by the A02 parameter record and the number of digits specified in the dialog records.

If the length of the recording data record exceeds 249 characters, a parameter setting error "A" meaning "Recording record too long" is generated, see also chapter.

The following operating scenarios can occur:

Scenario	Data in recording record
Input was canceled with the "Cancel" key	Cancelling function key processing: no input data
Input was terminated with the "OK" key	According to user input
Operation timeout expired	Input timeout: No input data
Reader input	According to user input

Info	Topic
3.14	A02 record
13.2	Parameter setting errors

3.10.31.3 Language specific settings

The user information can also be stored in multiple languages. For this purpose, a separate section with the corresponding language abbreviation is created in the definition file for the function key (addInputSteps_fk_XX.ini) for each desired language. In this section, the key **UserInfoX** is used to specify the corresponding user information in the desired language for each input step. The placeholder "X" is filled with the number of the desired input step.

The section name is the language abbreviation, e.g. [de], [fr], [es]. For certain countries, a country-specific identifier can also be specified in addition to the language identifier, e.g. [en_US] or [en_GB].

For details about the possible language identifiers see chapter [▶ 13.8].

The user information can have a length of max. 40 characters, longer texts are truncated. Leading blanks (ASCII 20hex) are taken into account in the texts of the UserInfo and are not truncated. This means that all characters after the equal sign are included in the UserInfo.

UserInfo4 of the language [de] shows an example for the use of spaces after the equal sign.

If no language-specific text is defined, the default text from the [General] section is used. If no text is defined there either, the user information remains empty.

Example for section [de] (can be omitted if German user info texts are already defined in section [General])

[de]

UserInfo1 =Bitte 1. Störgrund eingeben 1-9

UserInfo2 =Bitte 2. Störgrund eingeben 1-9 a-z

UserInfo3 =Bitte Stückzahl eingeben 1-999

UserInfo4 = 2. Ausweis-Eingabe

UserInfo5 =Bitte Barcode scannen

Example for section [en_US]

[en_US]

UserInfo1 =Please enter 1. error reason 1-9

UserInfo2 =Please enter 2. error reason 1-9 a-z

UserInfo3 =Please enter quantity 1-999

UserInfo4 =Please present the badge

UserInfo5 =Please scan the barcode

3.10.31.4 Parameterization example

The following parameterization example for the FK04 function key can be used as a basis for your own configurations.

addInputSteps_fk_04.ini

[General]

Dialog1 = G0100000010000Bitte 1. Störgrund eingeben 1-9

Dialog2 = G0200000010000Bitte 2. Störgrund eingeben 1-9 a-z

Dialog3 = G0100000030000Bitte Stückzahl eingeben 1-999

Dialog4 = G0010180500002. Ausweis-Eingabe

Dialog5 = G0131010500000Bitte Barcode scannen

[de]

UserInfo1 =Bitte 1. Störgrund eingeben 1-9

UserInfo2 =Bitte 2. Störgrund eingeben 1-9 a-z

UserInfo3 =Bitte Stückzahl eingeben 1-999

UserInfo4 =2. Ausweis-Eingabe

UserInfo5 =Bitte Barcode scannen

[en_US]

UserInfo1 =Please enter 1. error reason 1-9

UserInfo2 =Please enter 2. error reason 1-9 a-z

UserInfo3 =Please enter quantity 1-999

UserInfo4 =Please present the badge

UserInfo5 =Please scan the barcode

[fr]

UserInfo1 =s.v.p. entrer raison de la faute 1-9

UserInfo2 =s.v.p. entrer raison de la faute 1-9 a-z

UserInfo3 =s.v.p. entrer le nombre de pièces 1-999

UserInfo4 =veuillez entrer la badge

UserInfo5 =veuillez scanner le code-barres

[es]

UserInfo1 =p.f. entrar el motivo del fallo 1-9

UserInfo2 =p.f. entrar el motivo del fallo 1-9 a-z

UserInfo3 =p.f. entrar el número de piezas 1-999

UserInfo4 =p.f. presente tarjeta

UserInfo5 =escanee el código de barras

3.10.32 Show person related text

This parameter is only important with a positive answer ("Thank you") if a master record has been found and no mailbox text is shown.

This parameter defines whether personal text from master record shall be shown.

Standard factory setting
For no function key a personal text is shown.

Info	Topic
3.10.34	Necessary download record

3.10.33 Number of display info

This parameter is only important for showing display info from master records.

This parameter defines whether 4 or up to 10 display info shall be shown.

Standard factory setting
For all function keys 4 display info are shown.

Info	Topic
3.10.34	Necessary download record

3.10.34 Download record Function key functions

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID	Group address
DID	Device address
ON/OFF	Operating mode
>	Record type
3	Record type modification
F	Block identification
0	F00 to F05; F31 to F40
0	see following description
1	Operating mode 1;
3	see following description
1	Operating mode 2;
3	see following description
0	Number of digits 1. keyboard input, max. 24 digits
0	00 = No keyboard input/no list
2	Operating mode 3; (1. keyboard input)
1	see following description
A	Record type and record type modification,
0	see following description
0	Number of digits 2. keyboard input, max. 24 digits
0	00 = No keyboard input
0	Operating mode 4; (2. keyboard input)
0	see following description
0	Number of digits 3. keyboard input, max. 24 digits
0	00 = No keyboard input
0	Operating mode 5; (3. keyboard input)
0	see following description
0	List no. for 1. keyboard input, 01-20;
0	00 = No list
0	List no. for 2. keyboard input, 01-20;
0	00 = No list
0	List no. for 3. keyboard input, 01-20;
0	00 = No list
0	Operating mode 6; (extensions);
0	see following description
0	1. byte: Operating mode 7; (LEGIC Write function);
0	2. byte: Additional input steps

Block identification

Entry	Meaning
F00	Function key F0 (base state)
F01	Function key F1
F02	Function key F2
F03	Function key F3
F04	Function key F4
F05	Function key F5
F31	Function key F31
F32	Function key F32
F33	Function key F33
F34	Function key F34
F35	Function key F35
F36	Function key F36
F37	Function key F37
F38	Function key F38
F39	Function key F39
F40	Function key F40

Operating mode 1, byte 1

8	4	2	1	Hex value

3	2	1	0	Bit
				Function key memorizing 0 = No 1 = Yes
				Terminal awaits logical booking response in offline/autonomous mode 0 = No 1 = Yes
				Check on time authorization 0 = No 1 = Yes
				Individual master record required 0 = No 1 = Yes

8	4	2	1	Hex value

3	2	1	0	Bit
				Bookings allowed in online mode 0 = No 1 = Yes
				Bookings allowed in offline/autonomous mode 0 = No 1 = Yes
				Bookings allowed despite of full buffer memory 0 = No 1 = Yes
				Suppress display info in master record 0 = No 1 = Yes

Parameter example for operating mode 1

Function	0/1	Decimal	Hex
Function key memorizing	1	13	D
Terminal awaits logical booking response in offline mode	0		
Check on time authorization	1		
Individual master record required	1		
Bookings allowed in online mode	1	11	B
Bookings allowed in offline/autonomous mode	1		
Bookings allowed despite of full buffer	0		
Suppress display info	1		

Operating mode 2, byte 1

8	4	2	1	Hex value

3	2	1	0	Bit
				Buffer authorized recording data 00 = No 01 = Yes 10 = Depending on the entry in the time profile
				Buffer unauthorized recording data 00 = No 01 = Yes 10 = Depending on the entry in the time profile

Operating mode 2, byte 2

8	4	2	1	Hex value

3	2	1	0	Bit
				Badge input required 0 = No 1 = Yes
				Activate relay 0 = No 1 = Yes
				One / two badge inputs required 0 = One badge input required 1 = Two badge inputs required
				Double access blocking 0 = No 1 = Yes

Info	Topic
3.5.1	Hexadecimal coding

Operating mode 3 (1. keyboard input), byte 1

8	4	2	1	Hex value

3	2	1	0	Bit

Type data/code 1 for 1. keyboard input
see following table
Keyboard input format for 1. keyboard input
The parameter is set in the system and will not be evaluated at this point
Keyboard data numeric/alphanumeric for 1. keyboard input
0 = Numeric
1 = Alphanumeric
Type data/code 2 for 1. keyboard input; see following table

Type Data/code 1	Type Data/code 2	Meaning
0	0	Keyboard input type data after badge input
0	1	Keyboard input type code possible before badge input
1	0	Keyboard input type code after badge input
1	1	Keyboard input type code before or after badge input

Operating mode 3, byte 2

8	4	2	1	Hex value

3	2	1	0	Bit

Dialog text no. for 1. keyboard input
00 = Dialog text no. 0
01 = Dialog text no. 1
10 = Dialog text no. 2
11 = Dialog text no. 3
Dialog text no. for 1. badge input
00 = Dialog text no. 0
01 = Dialog text no. 1
10 = Dialog text no. 2
11 = Dialog text no. 3

Record type/Record type modification

The parameters define the record type and record type modification used to transmit a data record to the computer. All RT/RTM shown in the table in chapter 10.2 section "Recording data" can be defined.

Operating mode 4 (2. keyboard input), byte 1

8	4	2	1	Hex value

3	2	1	0	Bit

Reserved
Keyboard input format for 2. keyboard input
The parameter is set in the system and will not be evaluated at this point
Keyboard data numeric/alphanumeric for 2. keyboard input
0 = Numeric
1 = Alphanumeric
Reserved

Operating mode 4, byte 2

8	4	2	1	Hex value

3	2	1	0	Bit

Dialog text no. for 2. keyboard input)
0000 = Dialog text no. 0
0001 = Dialog text no. 1
0010 = Dialog text no. 2
0011 = Dialog text no. 3
0100 = Dialog text no. 4
0101 = Dialog text no. 5
0110 = Dialog text no. 6
0111 = Dialog text no. 7
1000 = Dialog text no. 8
1001 = Dialog text no. 9
1010 = Dialog text no. 10
1011 = Dialog text no. 11
1100 = Dialog text no. 12
1101 = Dialog text no. 13
1110 = Dialog text no. 14
1111 = Dialog text no. 15

Operating mode 5 (3. keyboard input)
The settings equate the settings of operating mode 4.

Info	Topic
3.5.1	Hexadecimal coding

Operating mode 6 (extensions), 1. byte

8	4	2	1	Hex value

3	2	1	0	Bit

Activate browser
0 = No
1 = Yes
The parameter will not be evaluated at this point
Reader no. for 1. badge input
0 = Reader 1
1 = Reader 2
Reader no. for 2. badge input
0 = Reader 1
1 = Reader 2

Operating mode 6, 2. byte

8	4	2	1	Hex value

3	2	1	0	Bit

Activate external application
0 = No
1 = Yes
Activate dialog mode
0 = No
1 = Yes
Display person related text
0 = No
1 = Yes
Number of display info
0 = 4 display info
1 = 10 display info

Parameter example for operating mode 6

Function	0/1	Decimal	Hex
Activate browser	1	9	9
Not evaluated	0		
Reader no. for 1. badge input	0		
Reader no. for 2. badge input	1		
Activate external application	0	4	4
Activate dialog mode	0		
Display person related text	1		
Number of display info (4)	0		

CL

For function variant 9x00 K7-40 und 9x00 K7-60 only.

Operating mode 7 (LEGIC Write)

8	4	2	1	Hex value
3	2	1	0	Bit

Write update data
0 = No
1 = Yes
Reserved 0
Reserved 0
No writing in the online mode
0 = No
1 = Yes

Aditional input steps (extended G dialog), Operating mode 7, 2. byte

8	4	2	1	Hex value
3	2	1	0	Bit

Reserved 0
Reserved 0
Reserved 0
Start additional input steps
0 = No
1 = Yes

Info	Topic
3.5.1	Hexadecimal coding

3.11 Lists

Lists are important when the parameter setting of a function key function is referring to a list number.

With every booking according to the function key functions, the terminal checks whether a keyboard input is set. If so, the terminal checks whether a list number is stored. If a list number is stored in the function key functions and the list does exist, the list will displayed to the booking employee.

Lists can also be implemented via G dialogs. In doing so, the list definitions are placed in the ini files. When booking, they are used to generate the list to be displayed for the runtime. This allows dynamic lists with variable contents to be displayed.

Info	Topic
3.10.22	List number for keyboard input
7.7.8	Definition of dynamic lists

3.11.1 List number

The list number defines a list explicitly. The number is necessary during the parameter setting of the function key functions to find a stored list. Up to 20 lists can be created (01 to 20).

Standard factory setting

No lists are stored.

Info	Topic
3.11.5	Necessary download record

3.11.2 Index in list

A list may contain up to 30 entries. The index represents the position within the list. The numbering must absolutely begin with 00 and must be in a continuous row; if any break occurs, the list will end there. The entry — — means the index is not used.

Standard factory setting

The entry — — is stored.

Info	Topic
3.11.5	Necessary download record

3.11.3 List text number

The list text number (00 to 99) defines the text which is displayed to the booking employee.

Standard factory setting

The list text number 00 is stored.

Info	Topic
3.19.6	List texts
3.11.5	Necessary download record

3.11.4 Data field

Any data can be entered in the data field (24 characters). The content of a data field is the data which is entered into the recording data record. The parameter "Number of digits with keyboard input" defines, how many digits of the data field will be entered into the recording data record.

Standard factory setting

The character — is entered in the data fields.

Info	Topic
3.11.5	Necessary download record

3.11.5 Download record Lists

The parameters described in the previous chapters are modified in the terminal with the following download record. Each download record may contain up to 8 blocks. So, e.g. for 30 entries within one list, **four** download records are necessary. The block identifications W01 to W20 serves only as a counter.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
W 0 1	Block identification W01 to W20	
0 0	List no. (01 to 20) 00 = no list	
— —	Index in List (00 to 29) — — = entry not defined	**1. block**
0 0	Text no. (00 to 99)	
— ... —	Data field; 24 digits (Entry in the recording data record)	
... ...	Up to 7 more blocks	

3.11.6 List definitions- summary

The following chapter summarizes once again all required parameterizations for displaying a list using download data records.

1. Parameterization in the function key functions download data record

Parameter	Meaning
Block identification	The block identification defines the function key to be used for loading the list.
Number of digits 1. keyboard input Number of digits 2. keyboard input Number of digits 3. keyboard input	If the value for the number of digits is between 01 and 24, the program will check whether a list number has been defined. If a list number has been defined, the associated list will be displayed, if it has previously been loaded to the terminal. The format of the data to be entered is defined by means of the parameters Operating mode 3 (for 1. keyboard entry) Operating mode 4 (for 2. keyboard entry) Operating mode 5 (for 3. keyboard entry).
List no. 1. keyboard input List no. 2. keyboard input List no. 3. keyboard input	Specifies a number used for selecting the associated list to be displayed. The list itself is defined via the "Lists" download data record.
Record type	The record type defines which record type is used for transmitting the registration record to the host.

2. Parameterization in the Lists download data record

Parameter	Meaning
Block identification	The block identification is used for counting the download data records if the list is supposed to contain more than 8 entries.
List number	If the parameter value is between 01 and 20, it will be interpreted as list number. Reference to this list number will be made in the "Function key functions" download data record.
Index in list	The index defines the number that precedes the list entry.
Text number	The text number defines the list text to be displayed. The list text itself is transmitted in the "Texts" download data record.
Data field	Input field for the person booking. The data entered here is entered in the registration record. The number of digits entered in the registration record is defined via the "Number of digits" parameter in the "Function key functions" download data record.

3. Parameterization in the Texts download data record

Parameter	Meaning
Block identification	The block identification defines the type of text to be transmitted. For list texts, the block identifications N00-N99 and n00-n99 are valid according to the language mode.
Text	Text to be displayed, 20 digits max.

Info	Topic
3.10.34	Download record Function key functions
3.11.5	Download record Lists
3.19.7	Download record Texts
3.19.6	Definition of list texts

3.12 Function key control

The parameters of the "Function key control" parameter group define the time a function key is automatically active from. Any other function keys may also be operated. For the function key control, up to 12 blocks can be stored in the terminal.

Each block consists of the parameters

- function key switchover time and
- function key number.

3.12.1 Function key switchover

The parameter defines the time a function key is active from.

Standard factory setting

No switchover time is stored.

Info	Topic
3.12.3	Necessary download record

3.12.2 Function key number

The parameter defines the function key to be activated.

Standard factory setting

No function key is stored.

Info	Topic
3.12.3	Necessary download record

3.12.3 Download record Function key control

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
Z 0 1	Block identification Z01, Z21 to Z27 See following table	
— — — —	Hour (Time of activation) Hour (4 — = no activation) Minute Minute	**1. block**
— —	Function key number See following table	
... ...	Up to zu 11 more blocks	

Block identification

The block identification defines for which weekday the parameters are valid.

Example

If no activation has been specified for Wednesday in the parameter "Function key activation", the parameter Z01 is valid on this day.

If no entry exists in the Z01 parameter record either, no function key activation is active for this day.

Input	**Meaning**
Z21	Parameters valid for Monday
Z22	Parameters valid for Tuesday
Z23	Parameters valid for Wednesday
Z24	Parameters valid for Thursday
Z25	Parameters valid for Friday
Z26	Parameters valid for Saturday
Z27	Parameters valid for Sunday
Z01	Parameters valid only for weekdays for which **no** weekday-dependent function key activation (Z21 – Z27) has been specified.

Function key number

Entry	Meaning
— —	No function key/Function key F0 (base state)
00	Function key F0/base state
01	Function key F1
02	Function key F2
03	Function key F3
04	Function key F4
05	Function key F5
31	Function key F31
32	Function key F32
33	Function key F33
34	Function key F34
35	Function key F35
36	Function key F36
37	Function key F37
38	Function key F38
39	Function key F39
40	Function key F40
09	Break signal control*

* If the function key 09 is set and the door-opener relay is not used, an intermission signal control can be realized. In the moment of the function key switchover, exclusively relay R1 is selected for the active time defined in the parameter "Operation time for relay". However, this does not apply to weekday-dependent function key control Z21 – Z27.

Info	Topic
3.9.11	Operation time relay

3.13 Relay control

The relay control is only available

- in the ONE-K7 terminal or
- with the function variant 9x00 K7-60.

Depending on the desired use case, the relay can be assigned via the sections [IOMapping] and [OutputFunctions] in the "application.ini" file.

The parameters of the "Relay control" group are used to define,

- whether
- when
- and-how long

the relay is to be activated outside its normal function as a door opener after a booking.

This can be used for pause signal control, for example. Up to 60 different periods with the precision of a minute can be defined.

Info	Topic
2.3	Function variants
14.3.1	File "application.ini"

3.13.1 Switch-on time

The parameter defines the point in time when the relay is switched on.

Standard factory setting

No switch-on time is defined.

Info	Topic
3.13.4	Necessary download record

3.13.2 Relay number

B-Client HR40 supports one relay only. Therefore, the factory setting for relay 1 (setting 00) may not be changed.

Standard factory setting

The setting 00 for relay 1 is defined.

Info	Topic
3.13.4	Necessary download record

3.13.3 Operation time

The parameter defines the operation time for the activated relay.

Standard factory setting

No operation time is defined.

Info	Topic
3.13.4	Necessary download record

3.13.4 Download record Relay control

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
Z 1 n	Block identification Z11 to Z13 See following table.	
— — — —	Hour (Switch-on time) Hour (4 mal — = no switch-on time) Minute Minute	**1. block**
0 0	Relay number Only 00 permitted.	
0 0 0	Operation time 000-999 = 0-99,9 seconds Value x 0,1 seconds	
0	Reserved	
... ...	Up to 19 more blocks	

Block identification

The use of the block identification each with 20 possible blocks allows to define up to 60 times accurately.

Entry	Meaning
Z11	Block 1 to 20
Z12	Block 21 to 40
Z13	Block 41 to 60

3.14 Badge definitions

The parameters of the "Badge definitions" parameter group define the evaluation process of the data that has been read from a badge. For this purpose, the badge content is separated in so-called blocks. Each block consists of the parameters

- from digit
- number of digits
- identification.

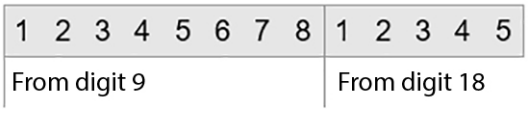
3.14.1 From digit

The parameter defines from which digit on the badge content is evaluated.

Standard factory setting

From digit 9

From digit 18



Info	Topic
3.14.7	Necessary download record

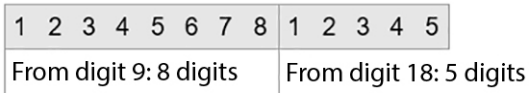
3.14.2 Number of digits

The parameter defines how many digits of the badge contents are evaluated.

Standard factory setting

From digit 9: 8

From digit 18: Number of digits 5



Info	Topic
3.14.7	Necessary download record

3.14.3 Identifier

The parameter defines how the badge data is interpreted and which kinds of checks are made.

Identifier	Meaning
0=Neutral	The data read from a badge will not be checked.
1= Customer number	If a customer number is stored in the terminal, it is compared with the data read from the badge. If the customer number is not identical with the stored customer number or a stored additional customer number, the error identification 2 "Customer number error" is entered into the recording data record. In the operating modes offline/autonomous, the booking is rejected with the message "Kunden-Nr. Fehler" (Customer number error).
3= ID number	The data read from a badge is interpreted as an ID number. It is used for access to an individual master record. It is also necessary if the function double access blocking shall be used.
4= Version number	The data read from the badge is compared with the version number in the individual master record. If the version number is not identical, the error identification ; "Version number false" is entered into the recording data record. In the operating modes offline/ autonomous, the booking is rejected with the message "Versions-Nr. falsch" (Version number wrong).
7=End of validity 8= Begin of validity	The data read from the badge is interpreted as date and time. The sequence YY/MM/DD/hh/mm (max. 10 digits) must be observed. The current date and time is compared with the data read from the badge. If date and time read from the badge has passed, error identification < "Time exceeded" is entered into the recording data record. In the operating modes offline/autonomous, the booking is rejected with the message "Zeitpunkt ueberschr" (Time exceeded).

Standard factory setting

From digit 9: Number of digits 8, identifier: 1

From digit 18: Number of digits 5, identifier: 3

1 2 3 4 5 6 7 8	1 2 3 4 5
From digit: 9 Number of digits: 8 Identifier: 1	From digit: 18 Number of digits: 5 Identifier: 3

Info	Topic
3.17	Time profiles
3.9.16	Customer number
5.5.1	Individual master record
3.14.7	Necessary download record

3.14.4 Separator block

A separator block must be defined if two badge inputs have been set via the parameter "One/ two badge inputs required". This way, the terminal will find the badge definitions which are valid for the second badge. A separator block is entered into the download record as follows:

From digit: 00

Number of digits: 00

Identifier: 0

The following is a complete parameter setting example for 2 badge inputs.

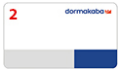
Info	Topic
3.10.6	2. badge input (1 & 2 badge inputs)

Example definition for 2 badge input with separator block

Badge content 1. badge

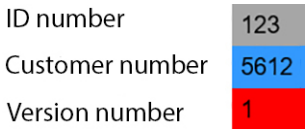
Badge content 2. badge

The following badge definitions are stored in the terminal via download.

Badge	From digit	Number of digits	Identifier	Meaning
	01	03	3	ID number
	05	04	1	Customer number
	09	01	4	Version number
	00	00	0	Separator block
	01	10	7	Validity

The terminal will read the following data from the badges:

1. badge:



2. badge:

The following badge data is entered into the recording data record:

.	
. . . .	Time Seconds and user language can be set additionally
.	Error identification
1 2 3	ID number
5 6 1 2	Customer number
1	Version number
9 9 0 4 1 8 1 2 0 0	End of validity. YYMMDD hhmm
:	

3.14.5 Filler

Fillers can be inserted between the badge blocks and will be transmitted in the recording data record. The value is 0. Fillers are entered into the download record as follows:

From digit: 00

Number of digits: Desired number of fillers

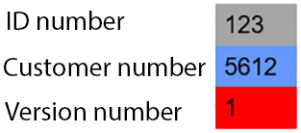
Identifier: 0

The following is a complete parameter setting example for fillers.

The following badge definitions are stored in the terminal via download:

From digit	Number of digits	Identifier	Meaning
01	03	3	ID number
05	04	1	Customer number
00	04	0	Filler
09	01	4	Version number

The terminal will read the following data from the badge:



The following badge data is entered into the recording data record:

.	
. . . .	Time Seconds and user language can be set additionally
.	Error identification
1 2 3	ID number
5 6 1 2	Customer number
0 **0** **0** **0**	**4 digits filler**
1	Version number
:	

3.14.6 Fix text

Fix text can be inserted between the badge blocks and will be transmitted in the recording data record. Fix texts is stored as a customer number prior in the terminal by use of an X02 record. Compared to the regular customer number (indicated by the identifier "1"), the customer number in the fix text will not be checked against the badge data. The fix text will be entered into the recording data record unchecked.

For this reason, the use of fix text and customer number simultaneously is not reasonable.

Fix text is entered into the download record as follows:

From digit: 00

Number of digits: Desired number of digits 1-8

Identifier: 1

The following is a complete parameter setting example for fix text.

The following badge definitions are stored in the terminal via download:

From digit	Number of digits	Identifier	Meaning
01	03	3	ID number
05	04	1	Fix text
09	01	4	Version number

The terminal will read the following data from the badge:

ID number	123
Customer number	5612
Version number	1

The following badge data is entered into the recording data record:

.	
. . . .	Time Seconds and user language can be set additionally
.	Error identification
1 2 3	ID number
5 **6** **1** **2**	**4 digits fix text from X02 record**
1	Version number
:	

3.14.7 Download record Badge definitions

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
A 0 2	Block identification	
0 9	From digit	**1. block**
0 8	Number of digits	
0	Reserved (0)	
1	Identifier	
... ...	Up to 9 more blocks.	

Identifier	**Meaning**
0	Neutral
1	Customer number
3	ID number
4	Version number
7	End of validity
8	Begin of validity

All defined blocks are entered in the order of their definition into the recording data record which is transmitted to the computer. The identifier itself is not included in the recording data record.

If the parameter "Fix record length for badge data" is set, blanks may be added or digits may be cut off in the recording data record. This will happen independently from the badge definitions.

3.15 Conversion of badge data

The parameters of the "Conversion of badge data" parameter group are used to define conversion rules for the badge data. The length of the converted data may differ from original badge data.

The conversion is done immediately after reading the badge and therefore before evaluating the data with the parameters of the "Badge definitions" parameter group. If the length of badge data has been changed due to the conversion, this must be considered.

A block with all parameters set to 00000000 indicates a block separator. The following blocks then belong to the next badge.

If only one separator block is defined for a badge, all data remain unconverted (standard setting). If for a badge at least one block is defined, only the data defined within a block are converted, undefined data are not output. If it is necessary to put all data to the output you have to define some blocks with output format "No conversion".

Info	Topic
3.14	Badge definitions

3.15.1 Conversion from digit

The parameter defines the starting position of the conversion, first digit is position 1. The value 00 indicates an unused block.

Standard factory setting

No conversion of badge data takes place.

Info	Topic
3.15.5	Necessary download record

3.15.2 Number of digits before conversion

The parameter defines the number of digits to be converted.

Standard factory setting

No conversion of badge data takes place.

Info	Topic
3.15.5	Necessary download record

3.15.3 Number of digits after conversion

The parameter defines the number of digits giving back to the output after conversion.

If the length of converted data is greater than indicated, the digits are cut off beginning on the left hand side. If the length of converted data is less than indicated, the digits on the left hand side are filled with ASCII-Character "0" (30hex).

Standard factory setting

No conversion of badge data takes place.

Info	Topic
3.15.5	Necessary download record

3.15.4 Mode of conversion

The parameter defines the mode of the conversion.

The following modes are possible:

Mode	Input format	Output format
0	Any	No conversion
6	Hexadecimal (0 - F)	Decimal (0 - 9)
7	Hexadecimal (0 - F)	Hexadecimal Special (0 - ?)
11	Hexadecimal Special (0 - ?)	Decimal (0 - 9)
12	Hexadecimal Special (0 - ?)	Hexadecimal (0 - F)

Standard factory setting

No conversion of badge data takes place.

Info	Topic
3.15.5	Necessary download record

3.15.5 Download record Conversion of badge data

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
A 1 2	Block identification	
0 0	Conversion from digit 00 = block not used (badge separator)	**1. block**
0 0	Number of digits before conversion	
0 0	Number of digits after conversion	
0 0	Mode of conversion	
...	Up to 9 more blocks.	

3.16 Contact inputs

The contact inputs are only available

- in the ONE-K7 terminal or
- with the function variant 9x00 K7-60.

Depending on the desired use case, the relay can be assigned via the sections [IOMapping] and [InputFunctions] in the "application.ini" file.

Info	Topic
2.3	Function variants
14.3.1	File "application.ini"
3.16.3	Example configuration

3.16.1 Allowed maximum door opening time

The parameter defines how long a door is allowed to be open after activating the door opener relay without sending an alarm record.

A door frame contact allows to supervise whether the allowed maximum door opening time is exceeded. The door frame contact is connected to one of the inputs.

To do so, the desired assignments in the sections

- [IOMapping]
- [InputFunctions]
- [OutputFunctions]

must be defined for the inputs and/or the relay.

If the door is still open after expiry of the allowed maximum door opening time, the terminal transmits an X2 alarm record to the computer. Only if the relay R1 is currently selected with static mode, the alarm record will not be generated.

It is also possible, if the maximum permitted length of time for the doors being open is exceeded, to activate the relay. This can be done by setting the corresponding parameter in the download record to "1".

Standard factory setting

The allowed maximum door opening time is 10.0 seconds.

Info	Topic
3.16.4	Necessary download record

3.16.2 Relay, active time after push button operation

This parameter defines how long the relay is activated after the push button was actuated. Pressing the push button also starts the allowed maximum door-opening time.

In cases of no terminal and no door handle are mounted inside a room, the door can be opened using a door opener push button.

To do so, the desired assignments in the sections

- [IOMapping]
- [InputFunctions]
- [OutputFunctions]

must be defined for the inputs and/or the relay.

When the button is pushed, the relay will be activated and the room can be left.

Standard factory setting

The relay is activated for 3.0 seconds after the button has been pushed.

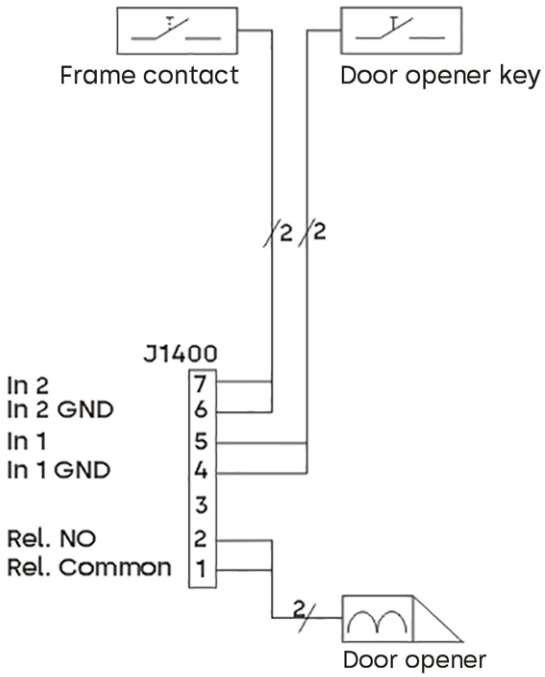
Info	Topic
3.16.4	Necessary download record

3.16.3 Wiring and configuration example

The following shows a possible wiring and .ini file configuration example to realize

- a terminal at the outside
- a door frame contact and
- a push button at the inside.

The technical data (contact load capacity and others) of the inputs/outputs can be found in the technical manuals of the terminals used.



[IOMapping]
BI01 = Int I01
BI02 = Int I02
BO01 = Int O01

[InputFunctions]
DoorOpenerKeyInside = BI01
FrameContact = BI02

[OutputFunctions]
DoorOpenerRelayOutside = BO01
DoorOpenerRelayInside = BO01

3.16.4 Download record Allowed max. door open time and relay

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
K 0 1	Block identification
1 0 0	Allowed maximum door opening time 000-99.9 seconds (100x0.1 seconds = 10.0 seconds)
0	Reserved 0
0 3 0	Operation time relay 000-99.9 seconds (030x0.1 seconds = 3.0 seconds
0 0 0	3 digits reserved 0
0	Signaling when exceeding the maximum permitted door opening time. 0= Only alarm record X2 is transmitted 1= In addition to alarm record X2, the relay is activated. The relay remains activated until the door is closed again.

3.17 Time profiles

Time profiles are only available

- in the ONE-K7 terminal or
- with the function variant 9x00 K7-60.

See also the overview of the function variants in chapter [▶ 2.3].

To effect time profile checks during an door control, the master record must refer to a time profile. For this purpose, the profile name is entered into the master record.

A maximum of 254 time profiles can be stored in the terminal. A time profile may contain up to seven time pairs.

Each time pair includes the parameters

- profile name
- time span (24h format)
- validity days
- identification byte.

3.17.1 Profile name

The profile name identifies each time profile stored in the terminal. Allowed profile names are 01hex to FEhex. The entry **00** means that no valid time profile is saved for this employee and a check for time authorization provides a negative test result.

Standard factory setting

No time profiles are stored in the terminal.

Info	Topic
5.5.1	Individual master record (Profile name)
3.17.5	Necessary download record

3.17.2 Time span

Please observe, that the time definitions are in 24h-format. This means, e.g. 1 pm=13:00, 2 pm=14:00 and so on.

Each time pair is valid for a certain time span, e.g. from 8:00 until 17:00. The time span finishes with the **elapse of the defined minute.**

Example: the time span 8:00 until 17:00 finishes at 17:00:59.

If the time span shall end accurately defined at 17:00, the time span parameter must be set to 16:59. (08:00 am to 4:59 pm)

The desired time span is defined by entering hours and minutes for the "start" and for the "end" of validity. For all undefined time spans no time authorization exists.

If no time authorization exists, the error identification 9 "No time authorization (profile)" is entered into the recording data record.

In the "offline/autonomous" operating modes the booking is rejected with the message "Keine Zeitber." (No time authorization).

Possible values: 00:00 to 23:59.

The entry "24:00" is not permitted!

Time spans may not overlap.

If, for example, a time span finishes at 05.00 pm, the next following one may not start at 05.00 pm but earliest at 05.01 pm.

Standard factory setting

No time spans are defined.

Info	Topic
3.17.5	Necessary download record

3.17.3 Validity days

The parameter states the days of the week the defined time pair will be valid for. During a booking, all time pairs are checked until time span and date will correspond to the current time and date.

Standard factory setting

No validity days are defined.

Info	Topic
3.17.5	Necessary download record

3.17.4 Identification bytes

The identification bytes contain the following parameters:

- Special day identification
- Time authorization
- Buffer authorized recording data
- Buffer not authorized recording data
- PIN check

Special day identification

The special day identification assigns the time pair to a special day. If a booking is made and the current day is detected as a special day, the time pairs are only checked on the special day identification. In this case, the check on validity days is not made. All time pairs are checked until time span and special day identification are in accordance with the current time and day.

The definition of the special days and its special day identifications are made via the special day table.

Time authorization

This parameter is only important, if the parameter "Check on time authorization" is set.

This parameter is used to assign the time authorization to the defined time span. If no time authorization is assigned, error identification 9 "No time authorization (profile)" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Keine Zeitber." (No time authorization).

Buffer authorized recording data

The parameter defines, if authorized recording data is buffered.

This parameter is mainly important if the parameter "Buffer authorized recording data" in the "Function key functions" parameter group is set to the value "Depending on entry in time profile", and the terminal is in the "offline/autonomous" operating mode. If buffer backup is set, the parameter is also valid for the "online" operating mode.

Buffer unauthorized recording data

The parameter defines, if unauthorized recording data is buffered.

This parameter is mainly important when the parameter "Buffer not authorized recording data" in the parameter group "Function key functions" is set to "Depending on entry in time profile" and the terminal is in the "offline/autonomous" operating mode. If buffer backup is set, the parameter is also valid for the "online" operating mode.

PIN check

This parameter is only important if a keyboard input is set. Additionally, the data type code must be set.

The parameter defines, whether the entered keyboard data is compared with the PIN stored in the master record.

If the entered keyboard data is compared with the PIN and the data differs, the error identification : "PIN wrong or missing" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Code falsch" (Code false).

Standard factory setting

- No special day identification is defined
- No time authorization is defined
- Authorized recording data is not buffered
- Not authorized recording data is not buffered
- No PIN check is defined.

Info	Topic
3.18	Special day table
3.10.3	Check on time authorization
3.10.15	Keyboard input
3.10.17	Data type "Code"
3.17.5	Necessary download record

3.17.5 Download record Time profiles

The parameters described in the previous chapters are modified in the terminal with the following download record.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
P n n	Block identification P01 to PFE See following description	
n n	Profile name 01hex to FEhex 00 = no time profile	
h h	Hour "Begin"	**1. time pair**
m m	Minute "Begin"	
h h	Hour "End"	
m m	Minute "End"	
n n	Validity days see following description	
n n	Identification bytes see following description	
...	Up to 6 more time pairs	

Block identification

During download/upload, the time profiles are distinguished by means of their block identifications. If the time profiles stored in the terminal are used, they are distinguished by means of their profile name.

Profile name

For a better overview it is useful, to have the second and the third byte of the block identification identical to the profile name.
Example: the block identification P01 refers to profile name 01, a.s.o.

Validity days, 1. byte

8	4	2	1	Hex value

3	2	1	0	Bit
				Wednesday 0 = Invalid 1 = Valid
				Tuesday 0 = Invalid 1 = Valid
				Monday 0 = Invalid 1 = Valid
				Reserved 0

2. byte

8	4	2	1	Hex value

3	2	1	0	Bit
				Sunday 0 = Invalid 1 = Valid
				Saturday 0 = Invalid 1 = Valid
				Friday 0 = Invalid 1 = Valid
				Thursday 0 = Invalid 1 = Valid

Info	Topic
3.5.1	Hexadecimal coding

Identification bytes, 1. byte

These bits are checked for conformity against the special day identification bits defined in the special day table.

8	4	2	1	Hex value
3	2	1	0	Bit

Special day identification

Reserved 0

2. byte

8	4	2	1	Hex value
3	2	1	0	Bit

PIN check
0 = No
1 = Yes

Buffer unauthorized data
0 = No
1 = Yes

Buffer authorized data
0 = No
1 = Yes

Check for time authorization
0 = No
1 = Yes

Info	Topic
3.5.1	Hexadecimal coding
3.18.2	Special day identification

3.18 Special day table

DC

The special days are only available

- in the ONE-K7 terminal or
- with the function variant 9x00 K7-60.

The special days are important if the master record refers to a time profile.

If a booking is made according to a time profile, the terminal will check whether the current day is defined as a special day. If so, the terminal will check whether a time pair is existing for this special day.

This check will compare

• The special day identification of the special day and

• The special day identification in the identification bytes of the time pairs.

If the special day identification of any time pair is identical, this time pair is used.

A maximum of 32 special days may be defined in the special day table.

Info	Topic
3.17	Time profiles
2.3	Function variants

3.18.1 Date indication

The terminal interprets a day with the date defined here as a special day.

Standard factory setting

No special day is entered in the special day table.

Info	Topic
3.18.3	Necessary download record

3.18.2 Special day identification

The special day identification is important in the identification bytes of the time profiles to assign the time pair.

The special day identification identifies a valid time pair for a special day.

Standard factory setting

No special day identification is stored.

Info	Topic
3.17.5	Time pairs
3.18.3	Necessary download record

3.18.3 Download record Special day table

The parameters described in previous chapters are modified in the terminal with the following download record. The 32 possible special days are transmitted to the terminal in **four** download records which have the block identification **L01 to L04**.

GID DID ON/OFF	Group address Device address Operating mode	
> 3	Record type Record type modification	
L 0 n	Block identification L01 to L04	
M M	Date: Month	**1. Special day**
D D	Date: Day	
n	Special day identification see following description	
0	Reserved 0	
...	Up to 7 more special days	

Special day identification

When comparing the special day identification in the time profile and the special day identification in the special day table, only one bit must match to get a positive check result.

8	4	2	1	Hex value
3	2	1	0	Bit

Special day 1
1 = valid
0 = invalid

Special day 2
1 = valid
0 = invalid

Special day 3
1 = valid
0 = invalid

Reserved 0

3.19 Texts

3.19.1 Function key texts

When the terminal is ready for bookings, the function key text is displayed in the upper line of the booking window. The text will also be displayed at the icon. Any function key text can be defined.

Standard factory setting

Function key	BI (comp.)	BI (multi)	Function key text
F0 (base state)	M00	m00	Access
F1	M01	m01	In
F2	M02	m02	Out
F3	M03	m03	Official absence
F4	M04	m04	Interrogation
F5	M05	m05	Special function
F31	M31	m31	In with reason
F32	M32	m32	Out with reason
F33	M33	m33	End official absence
F34	M34	m34	Begin intermission
F35	M35	m35	End intermission
F36	M36	m36	Special function
F37	M37	m37	Special function
F38	M38	m38	Special function
F39	M39	m39	Special function
F40	M40	m40	Special function

Info	Topic
3.19.7	Necessary download record

3.19.2 Dialog texts

When the terminal is ready for bookings, the dialog text is displayed in the lower line of the display/screen. Any dialog text may be defined.

Standard factory setting

Dialog text no.	BI (comp.)	BI (multi)	Dialog text
0	D00	d00	Present the badge
1	D01	d01	Enter code
2	D02	d02	Enter second badge
3	D03	d03	Enter identification
4	D04	d04	Enter personnel no.
5	D05	d05	Enter reason
6	D06	d06	Enter post
7	D07	d07	Enter cost center
8	D08	d08	Enter project
9	D09	d09	Enter internal order
10	D10	d10	Enter object type
11	D11	d11	Enter object id
12	D12	d12	Dialog text 12
13	D13	d13	Dialog text t 13
14	D14	d14	Dialog text 14
15	D15	d15	Dialog text 15

Info	Topic
3.19.7	Necessary download record

3.19.3 General texts

General texts are displayed after any errors occurred. If the terminal is in the offline or autonomous operating mode, the general text "Danke" (Thank you) is displayed after an authorized booking. The general text "Danke" (Thank you) is also displayed in the online operating mode if the computer has transmitted an R1-, R4 -R6 record as a logical booking response to the terminal.

The general text is an information for the booking employee and consists of max. 20 characters. Any general text can be defined. As some of the texts are assigned to possible errors, we recommend to change the texts only for translation into other languages to maintain their sense.

Standard factory setting

BI (comp.)	BI (multi)	General text
T01	t01	Key not allowed
T02	t02	Terminal blocked
T03	t03	Input time-out
T04	t04	Input error
T05	t05	Please wait
T06	t06	Parameter error
T07	t07	No data transmission
T08	t08	Thank you
T09	t09	Customer no. error
T10	t10	Status error
T11	t11	No master record
T12	t12	Master rec. blocked
T13	t13	No correction auth.
T14	t14	No off. abs. auth.
T15	t15	In/Out error
T16	t16	No time auth.
T17	t17	Code no. false
T18	t18	Version no. false
T19	t19	Not authorized
T20	t20	Repeat input
T21	t21	Read error
T22	t22	Secured error

BI (comp.)	BI (multi)	General text
T23	t23	Double access block
T24	t24	Time exceeded
T25	t25	Terminal occupied
T26	t26	End of dialog
T27	t27	Dialog impossible
T28	t28	Dialog request

Info	Topic
5.3	R1 record
5.3.1	R4-R6 record
3.19.7	Necessary download record

3.19.4 Mailbox texts

The mailbox text is an information for the booking employee and consists of max. 20 characters. Any mailbox text may be defined.

To display a mailbox text after an authorized booking, the number of the desired mailbox text must be entered in the logical booking response (R4/R5/R6 record) and/or in the master record.

Standard factory setting

Mailbox text no.	BI (comp.)	BI (multi)	Mailbox text
1	B01	b01	Mailbox text 1
2	B02	b02	Mailbox text 2
3	B03	b03	Mailbox text 3
4	B04	b04	Mailbox text 4
5	B05	b05	Mailbox text 5
6	B06	b06	Mailbox text 6
7	B07	b07	Mailbox text 7
8	B08	b08	Mailbox text 8
9	B09	b09	Mailbox text 9
10	B10	b10	Mailbox text 10
11	B11	b11	Mailbox text 11
12	B12	b12	Mailbox text 12
13	B13	b13	Mailbox text 13
14	B14	b14	Mailbox text 14
15	B15	b15	Mailbox text 15

Info	Topic
5.3.1	R4-R6 record
5.5.1	Individual master record
3.19.7	Necessary download record

3.19.5 Display info texts

The display info text is an information for the booking employee and consists of max. 20 characters. Any display info text may be defined.

To display a display info text after an authorized booking, this must be entered in the logical booking response (R4 –R6 record) and/or in the master record.

Standard factory setting

Display info text no.	BI (comp.)	BI (multi)	Display info text
1	I01	i01	Regular hours
2	I02	i02	Holidays
3	I03	i03	Overtime
4	I04	i04	Extra work
5	I05	i05	Display info text 5
6	I06	i06	Display info text 6
7	I07	i07	Display info text 7
8	I08	i08	Display info text 8
9	I09	i09	Display info text 9
10	I10	i10	Display info text 10

Info	Topic
5.3	R1 record
5.3.1	R4-R6 record
3.19.7	Necessary download record

3.19.6 List texts

Keyboard inputs during a booking can be realized with lists. In this case, a list is displayed and the booking employee can choose from any text.

A list text is a defined proposal for the booking employee and consists of 20 characters. Any list text may be defined.

Standard factory setting

List text no.	BI (comp.)	BI (multi)	List text
0	N00	n00	List text 0
1	N01	n10	List text 1
2	N02	n02	List text 2
...	...	...	...
...	...	...	...
...	...	...	...
97	N97	n97	List text 97
98	N98	n98	List text 98
99	N99	n99	List text 99

Info	Topic
3.11	Lists
3.19.7	Necessary download record

3.19.7 Download record Compatible and multilingual texts

The compatible and multilingual texts described in the previous chapters are modified in the terminal with the following download records.

The block identification will distinguish the different text types.

Texts with compatible block identification will be used only if no entry was made in the .ini file "interface.ini" in the [Languages] section.

In the following download record, "Danke" (Thank you) would be stored as the general text T08 in the terminal.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
T 0 8	Block identification M00 to M05, M31 to M40 D00 to D15, T01 to T28, B01 to B15, I01 to I10, N00 to N99
D a n k e	Text, 20 characters max.

Info	**Topic**
14.3.4	interface.ini file

In the terminal a multilingual function can be used. When using this function, the language for any texts will change in accordance with the flag choose on the touch screen of the terminal. The corresponding texts are stored in the terminal with the following download record.

This download records contain an additional language identification and use small letters in the first digit of the block identification.

In the following download record, "Thank you!" would be stored as the general text t08 in the terminal. The language identification for English "Great Britain" is used.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
t 0 8	Block identification m00 to m05, m31 to m40, d00 to d15, t01 to t28, b01 to b15, i01 to i10, n00 to n99
e n	Language identification: Digit 1 and 2 represent the language.
G B	Country identification: Digit 3 and 4 represent the country.
... ...	Digit 5 and 6 are currently irrelevant and filled up with blanks (20hex).
T h a n k y o u	Text, 20 characters max

If no country identification is needed, digit 3 and 4 can also be filled with blanks.

Info	**Topic**
13.8	Language identifications

3.20 Download record Settings in the .ini files

The following download record offers the possibility to change the settings in the terminal´s ini-files. The formatting stated below must be adhered.

For better understanding, some example parameters and values are given.

NOTICE

Data loss in .ini files

The "text.ini" files and the customer-specific files for "Additional input steps" (for example, "addInputSteps_fk_04.ini") contain characters in Unicode Big Endian format. The z00 record uses ASCII characters. Therefore, the mentioned files cannot be set or edited by use of the z00 record.

Edit the mentioned files with the PEC or a suitable editor.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification
z 0 0	Block identification
/ d a t a / ... / d e b u g . i n i	Path and file name (relative or absolutely) a) Relative (to the current directory) First character unlike / Example: files/init/interface.ini Current directory: data/data/com.kaba.apps.hr b) Absolutely: First character / Example: /data/data/com.kaba.apps.ba/files/init/debug.ini

[F i l e]	Name of the section in square brackets (mandatory). Example: [File]
N a m e	Name of the parameter. Example: Name
=	Mandatory: equal sign
d e b _ l o g	Parameter value deb_log1 Using this example creates a debug file named deb_log1 in the defined directory.

4 Communication via HTTP/HTTPS/XML

HTTP/HTTPS/XML communication

This is intended for the online communication between terminal and server. The occurring data records are marked up as XML files and sent to the server and/or terminal via HTTP/HTTPS.

Unlike the other communication paths (e.g. UDP) the terminal is the active side in the HTTP/HTTPS/XML communication and initializes the communication. That means that data records or requests cannot be sent simply from the server to the terminal. In fact, they must be sent from the terminal in a booking response to a request.

Requests to the server are sent using the HTTP methods "GET" or "POST". The method to be used can be defined by the HTTP_REQUESTMETHODE_FOR_BOOKING parameter in the [Common] section of the "b client_xml.ini" file.

The communication examples in this manual use the "GET" method.

Basic authentication

The basic authentication regarding RFC 2617 is realized via the declaration of username and password. Username and password are transmitted Base64 coded in the header of the request or post. When using SSL/TSL encryption with HTTPS the secured connection will be established prior to the transmission of any password. This way it is assured that the passwort cannot be monitored when using base authentication.

For this purpose, 2 new XML system variables are defined and stored in the b Client _xml.ini file. These are:

AUTHENTICATIONNAME=User name

AUTHENTICATIONPASSWORD=Password.

If one or both of these variables are not defined, the authorization will not be set. This is also the default status of the XML communication.

he coding of password and user name take place according to ISO 8859-1 character set. More character sets are not supported.

If no authorization is available the server responds with code "401: Unauthorized".

Setting of the HTTP/HTTPS/XML communication
This is done by means of the service interface of the terminal, local parameter setting or configuration of the "communication.ini" file in the path /data/data/com.kaba.apps.ba/files/init.

More detailed information about the service interface is to be found in the manual of the used device in the "Service interface" chapter.

Before an HTTP/HTTPS/XML communication to the server can be established the following files must exist on the server:

- the translation file (default name: xmltrans.xml, see chapter [▶ 4.8] XML translation file.
- the configuration file (default name: xmlconfig.xml, see chapter [▶ 4.9] XML configuration file.

The file names in this documentation always refer to the default settings. File names can deviate in real installations.

In the examples, URLs are used in the HTTP format. In case of encrypted data traffic, this must be HTTPS.

The HTTP/HTTPS/XML communication can also be established via a proxy server. In this case, the corresponding parameters for the proxy server (port and IP address) must be defined. This can be done in one of the following ways:

- in the Service Interface of the terminals or by
- local parameter setting or
- in the "communication.ini" file in the path /data/data/com.kaba.apps.ba/files/init.

4.1 HTTP/HTTPS/XML: Abbreviations used in the text

HTTP/HTTPS	Hypertext transfer protocols (unsecured/secured).
XML	Markup language for encoding data in a textual format.
RRT	Asymmetric encryption system used for data encryption and digital signatures.
TLS	Network protocol for secured data transmission. Currently the versions TLS 1.2 and TLS 1.3 are supported.
RFC	Numbered series of documents concerning internet related technical and organizational topics.
PKI	System that issues, distributes and verifies certificates.
DRT	US standard about digital signatures.
DH	Diffie-Hellman key exchange algorithm.
CRL	List of invalid certificates.
SHA-1	Standardized, cryptologic hash functions.
SHA-2	Standardized, cryptologic hash functions, successor to SHA-1. SHA-2 comprises the four cryptologic hash functions SHA-224, SHA-256, SHA-384 and SHA-512.

4.2 Data encryption via HTTPS

4.3 HTTPS: General

The internet protocol HTTPS was designed to provide secure data exchange via the internet. It serves for the encryption of the communication between web servers and clients. Compared to the unsecured protocol HTTP, HTTPS uses SSL (Secure Socket Layer) or its successor TLS (Transport Layer Security) as transport layer protocols.

The data encryption between dormakaba terminals and the web servers take place in two steps:

- Exchange of certificates to authorize the communication partners
- Exchange of session keys to encrypt the user data.

In the following chapters you will find information about requirements and limitations on the certificate exchange. The encryption of the user data is done in the background. For maximum security we recommend the use of 128-bit encryption algorithms if the system performance allows this. During the operation, a combination of secured and unsecured data traffic is possible at the same time.

Before establishing the secured data traffic it is recommended to check all the connections without encryption first.

4.4 HTTPS: Properties of the terminal

In the dormakaba terminal the following requirements must be met for the encryption of the data traffic:

- The XML communication must be activated in the terminal (ETH/XML interface).
- A subdirectory **/SSL** must exist in the terminals /data/data/com.kaba.apps.hr/files directory. If necessary, this directory must be generated:
  /data/data/com.kaba.apps.hr/files/SSL.
- The certificate authority root certificate must be stored in the /data/data/com.kaba.apps.hr/files/SSL directory.
  The file name **root.pem** or **root.crt** for the certificate is mandatory. The certificate root.crt is searched first, after that the certificate root.pem.

The various Android versions have different API levels and hence different implementations of SSL/TLS protocol versions. Further information can be found in the documentations about the various Android versions, API levels and protocol implementations of the manufacturer (Google Inc).

4.5 Server authentication

Please observe the following limitations when creating and using certificates. These limitations are summarized in two groups and are shown in the following tables.

Limitations regarding the TLS implementation

- dormakaba terminals can only work as TLS clients.

Limitations regarding the PKI certificates

1.	dormakaba terminals support only X.509v3 certificates.
2.	dormakaba terminals support only RSA and RSA_EXPORT key exchange methods and consequentially all certificates must be signed with and contain only RSA keys. DSA keys and DH key exchange methods are not supported.
3.	dormakaba terminals support RSA keys of 512, 1024, 2048 and 4096 bits.
4.	dormakaba terminals ignore "Unique Identifiers" as defined in RFC 2459 but will accept certificates holding these properties.
5.	dormakaba terminals support only "BasicConstraint" extensions as defined in RFC 2459.
6.	dormakaba terminals will ignore the following extensions but will accept certificates holding these extensions: a. Authority key identifier b. Subject key identifier c. Private key usage period d. Subject directory attributes e. Policy mappings f. Key usage extension (if not marked as critical) g. Certificate policies (if not marked as critical) h. Subject alternative name (if not marked as critical) i. Issues alternative name (if not marked as critical) j. Extended key usage (if not marked as critical)
7.	CRL functionality will not be supported.

4.6 Client authentication

The client authentication is a component of the HTTPS communication. The client authentication is the process by which the terminal (client) has to confirm its communication authorization to the server. This is done via a certificate with a private key which is stored on the terminal. During the initial SSL handshake, the public part of the certificate is transferred to the server for testing. The server validates the certificate by checking whether the transferred certificate "matches" a trusted root or CA certificate which was been configured in the keystore of the server.

The configuration of the server is server-specific for this function. For more details, please refer to the respective server documentation.

Saving a corresponding certificate named client.p12 in PKCS#12 format to the specified directory is sufficient for the configuration of the terminal.

The terminal certificate is "personalized" when the IP address of the terminal is created.

/data/data/com.kaba.apps.hr/files/SSL/

Name	Größe	Geändert	Rechte	Besitzer
..		14.05.2019 23:18:20	rwxrwxrwx	root
client.p12	6 KB	14.05.2019 17:09:09	rw-rw-rw-	root
root.crt	3 KB	15.03.2018 10:04:54	rw-rw-rw-	root

The certificate organization on the terminal is as follows:

Path: /data/data/com.kaba.apps.hr/files/SSL

Name of certificate: client.p12

File format of certificate: PKCS#12

The password for the certificate is saved to the b-client_xml.ini file in the parameter CLIENT_CERT_PASSWORD=.

4.7 Specification of system variables

The terminal provides a set of pre-defined system variables with fixed functions for the HTTP/XML communication. The system variables are stored in the section **Common** of the **b-client_xml.ini file**, to have the settings available if the system must be restarted.

All system variables can be changed at runtime via XML pages except the variables SERIALNUMBER and MACADDRESS which are read-only.

When using the system variables, please observe the notation!

When using encrypted communication, only syntax 1 is permitted!

The system variables can also be used in URL requests to transmit information to the server. In these cases, the fact that a variable and not a text is concerned must be made clear by the introductory character "$" and the name of the variable in brackets.

Example:

"$(GID)" is replaced by the value of the GID system variable.

The following list contains all system variables and their meaning.

System variables in the section "Common" of the b-client_xml.ini file

System variable	Possible values	Description
GID	00-29 Default: 0	The GID is used for the identification of a terminal group. It must be set with initial start-up of the terminal.
DID	00-59 Default: 0	The GID is used for the identification of an individual terminal within a terminal group. It must be set with initial start-up of the terminal.
PROXY	IP Address:Port number Default: empty	IP address of the proxy server, followed by a colon and specification of the port number for communication with the terminal.
ONLINEIP	IP Address:Port number/Directory	The parameter configures the online server. It must be set with initial start-up of the terminal. The definition of the directory and the port is optional. If no definitions are made for the directory, the requested files are searched for in the root directory of the server.

System variables in the section "Common" of the b-client_xml.ini file

System variable	Possible values	Description
TERMTYP	Character string	The specification in TERMTYP defines the terminal type and is necessary for program start.
SERIALNUMBER	Character string	Serial number of the terminal. This value is read-only.
MACADDRESS	Character string in the format: xx-xx-xx-xx-xx-xx Example: 0C-FF-34-5A-23-1B	MAC address of the terminal. This value is read-only.
SYSTIME	YYYYMMDDHHMMSS	The time on the terminal is set by means of the SYSTIME variables. This corresponds to sending of a U1 record.
INITIAL_XMLTRANSLATION	**Syntax 1:** http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: xmltrans.xml. The rest of the call will be added automatically.	Name of the XML translation page which is requested by the server during the program start. **Syntax 1:** Defines a complete URL in the format HTTP or HTTPS. System variables may be used with this call as needed. The ampersand character (&) must be entered as a <&> string. **Syntax 2:** The call has the following fixed structure: http://$(ONLINEIP)/$(INITIAL_XMLTRANSLATION)?GID=$(GID)&DID=$(DID)&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS).
INITIAL_XMLCONFIG	**Syntax 1:** http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: xmlconfig.xml. The rest of the call will be added automatically.	Name of the XML configuration page which is requested by the server during the program start. **Syntax 1:** Defines a complete URL in the format HTTP or HTTPS. System variables may be used with this call as needed. The ampersand character (&) must be entered as a <&> string. **Syntax 2:** The call has the following fixed structure: http://$(ONLINEIP)/$(INITIAL_XMLCONFIG)?GID=$(GID)&DID=$(DID)&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS).

System variables in the section "Common" of the b-client_xml.ini file

System variable	Possible values	Description
CHECKONLINETIME	Numerical data Default: 60	Time in seconds within which the terminal checks the availability of the server in operating mode "Online".
CHECKOFFLINETIME	Numerical data Default: 10	Time in seconds within which the terminal checks the availability of the server in operating mode "Offline".
CHECKONLINEPAGE	**Syntax 1**: http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: unknown.htm. The rest of the call will be added automatically.	Defines the page which is requested periodically by means of CHECKONLINETIME in terminal operating mode "Online". **Syntax 1:** Defines a complete URL in the format HTTP or HTTPS. System variables may be used with this call as needed. The ampersand character (&) must be entered as a <&> string. **Syntax 2:** The terminal software prefixes the file name with the call http://$(ONLINEIP)/.
CHECKOFFLINEPAGE	**Syntax 1:** http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: unknown.htm. The rest of the call will be added automatically.	Defines the page which is requested periodically by means of CHECKOFFLINETIME in terminal operating mode "Offline". Syntax 1 and syntax 2 as for "CHECKONLINEPAGE".
ERRORPAGE	**Syntax 1:** http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: error.xml. The rest of the call will be added automatically.	Name of the XML error page which is requested by the terminal when an error occurs. **Syntax 1:** Defines a complete URL in the format HTTP or HTTPS. System variables may be used with this call as needed. The ampersand character (&) must be entered as a <&> string. http://$(ONLINEIP)/error.xml?GID=$(GID)&DID=$(DID)&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS). The following data will be addend to the end of the call: &DATA=$(BPA,1)&ERROR=<Text> **Syntax 2:** The call has the following fixed structure: http://$(ONLINEIP)/$(ERRORPAGE)?GID=$(GID)&DID=$(DID)&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS)&DATA=$(BPA,1)&ERROR=<Text>

System variables in the section "Common" of the b-client_xml.ini file

System variable	Possible values	Description
HTTP_REQUESTMETHODE_FOR_BOOKING	GET (Default) POST	Defines the HTTP request (GET or POST) that is used to transmit the data records from the terminal. See also chapter chapter.
CONNECTIONTIMEOUT	Numerical data Default: 0	Defines the time in the "Longpolling" operation mode, a request remains active. After elapse of this time the terminal aborts the longpolling page request. The value "0" means, the waiting time is unlimited.
LONGPOLLINGPAGE	**Syntax 1:** http://IPADDRESS/xxx or https://IPADDRESS/xxx **Syntax 2:** File name default: long_polling.xml. The rest of the call will be added automatically.	Defines the page which is requested periodically in the "Longpolling" operation mode. More information about this operation mode can be found in chapter chapter. Syntax 1 and syntax 2 as for "CHECKONLINEPAGE".
AUTHENTICATIONNAME	Characters according to ISO 8859-1	Username for Basic Authentication
AUTHENTICATIONPASSWORD	Characters according to ISO 8859-1	Password for Basic Authentication
CLIENT_CERT_PASSWORD	Characters according to ISO 8859-1	Password for Client Authentication

4.8 XML translation file

The translation file (default name: xmltrans.xml) serves as translation table for the occurring data records. This table must contain a definition for each RT/RTM to be edited. The translation table must be available on the server with establishment of the communication with the terminal. If the translation file does not exist at this point, the terminal continues to establish the communication every 60 seconds.

The terminal requests the translation table via the system variable "INITIAL_XMLTRANSLATION" while starting the terminal application. The translation table can be edited within the operating time and downloaded to the terminal as an attachment of **any** response. It must be assured to download **only complete** translation tables, not only the changes made.

Editing the translation table may be necessary in the following cases:

- changes in the structure of the booking record
- changes in the assignment of the function keys.

A new translation table downloaded with a response replaces the existing translation table.

The translation table must be introduced with the tag **<trans>** and terminated with the tag **</trans>**. The actual translation regulations are in a CDATA area.

The CDATA area is within the character string **<![CDATA[Content]]>** .

The reaction on data records based on the RT/RTM is defined in the content of the translation table. For this, the translation definitions are specified by XML tags which consist of the RT/RTM of a data record and the URL statement, see example definition.

A copyable text version of this example is given on he following page.

```
<?xml version ="1.0"?
    <trans>
        <![CDATA[
            <B1>
                <URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B1>
            <B2>
                <URL=http://$(ONLINEIP)/out.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B2>
            <B3>
                <URL=http://$(ONLINEIP)/officialabsence.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B3>
            <B0>
                <URL=http://$(ONLINEIP)/inquiry.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B0>
            <F0>
                <URL=http://$(ONLINEIP)/special.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </F0>
        ]]>
    </trans>
```

Copyable text version of the previous example.

Depending on the used text editor, occuring tabs or space characters or line feeds must be deleted before using this example.

```
<?xml version ="1.0"?
<trans>
<![CDATA[
<B1>
<URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B1>
<B2>
<URL=http://$ (ONLINEIP)/out.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B2>
<B3>
<URL=http://$ (ONLINEIP)/officialabsence.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B3>
<B0>
<URL=http://$ (ONLINEIP)/inquiry.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B0>
<F0>
<URL=http://$ (ONLINEIP)/special.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</F0>
]]>
</trans>
```

Suppress data records

Data records can be suppressed, depending on their RT/RTM. To do so, the RT/RTM to be suppressed must be defined with an **empty** URL, see the examples below. If an RT/RTM with an empty URL occurs, no HTTP request will be send to the server. The generated data record will be deleted.

Example:
Suppressed RT/RTM with empty URL.

A copyable text version of this example is given on he following page.

```
<?xml version ="1.0"?
    <trans>
        <![CDATA[
            <B1>
                <URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B1>
            <B2>
                <URL=http://$(ONLINEIP)/out.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B2>

            Beispiele/Examples:
            <A0>
                <URL=/>
            </A0>

            <B3>
                <<URL=/>
            </B3>

        ]]>
    </trans>
```

Copyable text version of the previous example.

Depending on the used text editor, occuring tabs or space characters or line feeds must be deleted before using this example.

```
<?xml version ="1.0"?
<trans>
<![CDATA[
<B1>
<URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B1>
<B2>
<URL=http://$(ONLINEIP)/out.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B2>
Beispiele/Examples:
<A0>
<URL=/>
</A0>
<B3>
<URL=/>
</B3>
]]>
</trans>
```

Defining a general operating rule

All undefined RT/RTMs can be treated according to a general operating rule. This rule is defined by a tag with the wildcard characters " * * " and a respective URL. A **single** wildcard character in connection with **another single character** is not permitted.

Example 1:

All undefined RT/RTMs are treated according to the definitions which take place in the defined URL (page "unknown.xml").

Example 2:

The URL definition in example 2 is **empty**. This way, all undefined RT/RTMs are suppressed. If any of these RT/RTM occurs, no HTTP request will be send to the server. The generated data records will be deleted.

A copyable text version of this example is given on he following page.

```
<?xml version ="1.0"?
    <trans>
        <![CDATA[
            <B1>
                <URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
                &SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
            </B1>

            Beispiel/Example 1:
            <**>
                <URL=http://$(ONLINEIP)/unknown.xml?DATA=$(BPA,1)/>
            </**>

            Beispiel/Example 2:
            <**>
                <URL=/>
            </**>

        ]]>
    </trans>
```

Copyable text version of the previous example.

Depending on the used text editor, occuring tabs or space characters or line feeds must be deleted before using this example.

```
<?xml version ="1.0"?
<trans>
<![CDATA[
<B1>
<URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&SASAM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18,13)/>
</B1>
Beispiel/Example 1:
<**>
<URL=http://$(ONLINEIP)/unknown.xml?DATA=$(BPA,1)/>
</**>
Beispiel/Example 2:
<**>
<URL=/>
</**>
]]>
</trans>
```

The data itself is specified in a code in the URL request to the server. In addition to the system variables (e.g. GID/ DID, ONLINEIP etc.) the keyword "BPA" is available. It specifies

- from which digit and
- how many digits

of the data within the data record are inserted in the URL request.

Example:

- `RTRTM=$(BPA,7,10)`: the record type/record type modification is inserted at digit 7 and comprises 10 digits.

If **no** number of digits is specified, **all** the digits are inserted, starting with the specified digit to the end of the data record.

Example:

- `ID=$(BPA,7)`: all the digits of the ID number are inserted starting with digit 7 to end of the data record.

Example definition for a function key with RT/RTM "In" (B1):

```
<B1>

<URL=http://$(ONLINEIP)/in.xml?GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)
&RTRTM=$(BPA,4,2)&TIMESTAMP=$(BPA,7,10)&ID=$(BPA,18)/>

</B1>
```

The B1 booking on the terminal results in the following URL request on the server (assuming the terminal is ready for operation, that means the configuration file has already been requested and transmitted, see chapter [▶ 4.9].

http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&RTRTM=B1
&TIMESTAMP=0801171523&ID=0200000000001

Definition entry	Meaning
http://	Protocol definition
12.12.50.158	Server IP address
8080	Server port number
in.xml	Name of the XML file for the booking response to be called
GID=5	Group address of the terminal
DID=5	Device address of the terminal
MODE=0	Operating mode of the terminal
RTRTM=B1	Record type/record type modification of the actuated function key (B1="In")
TIMESTAMP=0801171523	Time stamp, 10 digits starting with digit 7 (BPA,7,10)
ID=0200000000001	Badge data, all digits starting with digit 18 (BPA,18)

4.9 XML configuration file

The configuration file (default name: xmlconfig.xml) contains configuration data of the terminal. It must be available on the server with establishment of the communication with the terminal. If the configuration file does not exist at this point, the terminal continues to establish the communication every 60 seconds.

The terminal requests this file with start of the terminal application via the system variable "INITIAL_XMLCONFIG".

The configuration file must be introduced with the tag **<cfg>** and terminated with the tag **</cfg>**.

- The **system variables** can be specified straight as variables in the **sysvar** objects by means of `name` and `value`.
- The **parameters** and **data records** of the terminal application are specified in the **sysdef** objects by means of the variables `TERMREC` and `value`.

```
<?xml version="1.0"?>
<cfg>
        <sysvar name="GID" value="00"/>
        <sysvar name="DID" value="00"/>
        <sysvar name="ONLINEIP" value="123.0.0.1:8080"/>
        <sysvar name="CHECKONLINETIME" value="60"/>
        <sysvar name="CHECKOFFLINETIME" value="10"/>
        <sysdef name="TERMREC" value="@@0>3F0103100720XX"/>
        <sysdef name="TERMREC" value="@@0>3F0200110021B2"/>
        <sysdef name="TERMREC" value="@@0>3F0300110021B3"/>
        <sysdef name="TERMREC" value="@@0>3F0400010021B0"/>
        <sysdef name="TERMREC" value="@@0>3F0500110521F0"/>
        <sysdef name="TERMREC" value="@@0>3M00REGISTRATION      "/>
        <sysdef name="TERMREC" value="@@0>3M01REGISTRATION      "/>
        <sysdef name="TERMREC" value="@@0>3T05PLEASE WAIT       "/>
        <sysdef name="TERMREC" value="@@0>3T08THANK YOU         "/>
        <sysdef name="TERMREC" value="@@0>3D00ENTER CODE        "/>
        <sysdef name="TERMREC" value="@@0>3D03PRESS GREEN KEY   "/>
        <sysdef name="TERMREC" value="@@ D1Configured"/>
</cfg>
```

If you use a system variable in a configuration file that contains an URL, the following characters require a specific notation.

Character	Notation
<	<
>	>
&	&
"	"
'	'

Please see the following code example.

Code example for an xmlconfig.xml file.

Please note the notation of the ampersand character inside the URL.

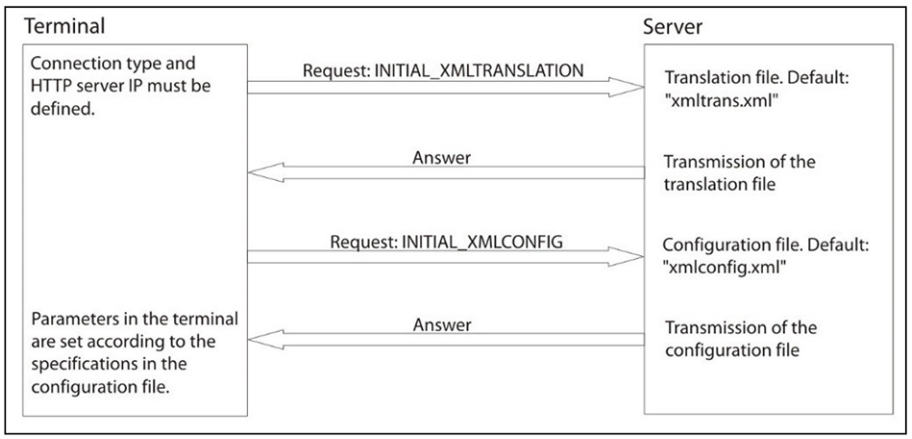
```
<?xml version="1.0" encoding="iso-8859-1"?>
<cfg>
  <sysvar name="CHECKONLINETIME" value="0"/>
  <sysvar name="CHECKOFFLINETIME" value="30"/>
  <sysvar name="CHECKONLINEPAGE"
  value="https://$(ONLINEIP):8443/Hostsimulation/
  PollingServlet?GID=$(GID)&amp;DID=$(DID)&amp;
  TERMTYP=$(TERMTYP)&amp;SERIALNUMBER=$(SERIALNUMBER)
  &amp;MACADDRESS=$(MACADDRESS)"/>
  <sysdef name="TERMREC" value="@@ T0" />
  <sysdef name="TERMREC" value="@@ D1Terminal Configured" />
</cfg>
```

4.10 XML: Establishing the communication

For the establishment of a communication between terminal and server the following prerequisites must be met:

- The communication type "HTTP/XML" has been selected in the terminal
- The translation file is on the server, see chapter [▶ 4.8].
- The configuration file is on the server, see chapter [▶ 4.9].

The communication is established based on the following scheme:



After ending the communication the terminal has received the XML translation file and the communication parameters and is ready for operation.

Errors during the XML startup process

The XML communication will not be started if an error occurs during the start process, e.g. the server answer to the request of the XML translation file is missing. In this case, the start process and the server request is repeated every 60 seconds.

4.11 Initializing the b-client_xml.ini file

During the installation process of the B-Client, among others the following files are copied to the terminal´s file system

1. b-client_xml.ini file into the

\data\data\com.kaba.apps.hr\files\coldstart\delivery\init directory.

This file contains the factory settings. During a cold restart, this file will not be deleted or overwritten.

2. b-client_xml.ini file into the \data\data\com.kaba.apps.hr\files\init directory.

In delivery state, this file is a copy of the file described in 1. In this copy file, customer specific settings are carried out.

During a cold restart, this file will not be deleted or overwritten.

Reproducing the delivery status
Select "Restart terminal software into delivery state" in the tab "System" in the terminal menu "Settings".

This will overwrite the file

(2) b-client_xml.ini

in the directory \data\data\com.kaba.apps.hr\files\init

with the content of the file

(1) b-client_xml.ini

in the directory \data\data\com.kaba.apps.hr\files\coldstart\delivery\init.

4.12 XML: Online/Offline mode

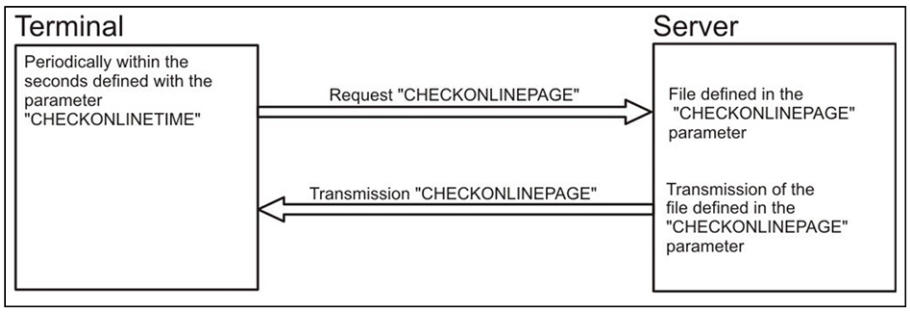
4.12.1 XML: Online mode

In an established connection between terminal and server (terminal mode online), the terminal checks the availability of the server periodically. This is done by means of an URL request (CHECKONLINEPAGE) to the server. The time interval (in seconds), in which the terminal transmits the URL request is defined by the system variable "CHECKONLINETIME".

With these URL requests the terminal attempts to establish a connection to the server and requests the file which is stored in the system variable "CHECKONLINEPAGE".

Subsequently the connection is closed. The graph below illustrates this process.

If the system variable "CHECKONLINETIME" is set to 0, the terminals switches to the "Longpolling" mode, see chapter [▶ 4.13].



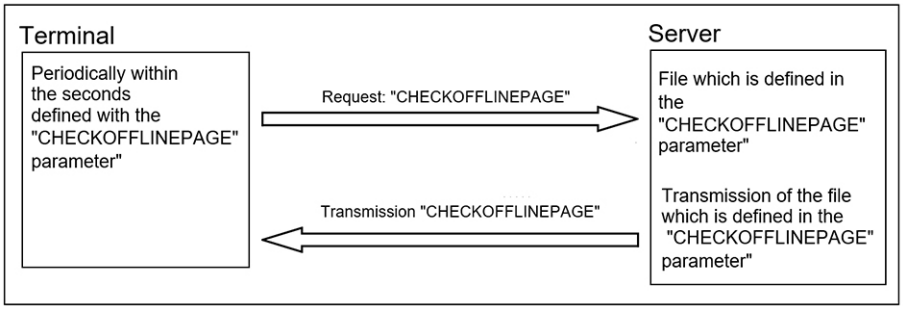
4.12.2 XML: Offline mode

If there is no connection between the terminal and the server any longer, the terminal switches temporarily to offline mode. This happens automatically after elapse of the time "CHECKONLINETIME" or if there is no booking response from the server.

The terminal now records all booking data in offline mode and attempts at regular intervals to establish a connection with the server. This is done by means of a URL request (CHECKOFFLINEPAGE) to the server to the port which is defined in the parameter "ONLINEIP". The terminal transmits data using the "GET" method and requests the file which is stored in the system variable "CHECKOFFLINEPAGE". Subsequently the connection is closed. The time interval (in seconds) in which the terminal transmits the URL requests is defined by means of the system variable "CHECKOFFLINETIME".

When the server is available again, the terminal gets a response to the "CHECKOFFLINEPAGE" request and switches to online mode.

Subsequently the connection is closed. The graph below illustrates this process.



4.13 XML: Longpolling mode

4.13.1 Activation of Longpolling

For the activation of Longpolling at the B-Client (terminal), the CHECKONLINETIME variable is used. The variable is entered in the b-client_xml.ini file.

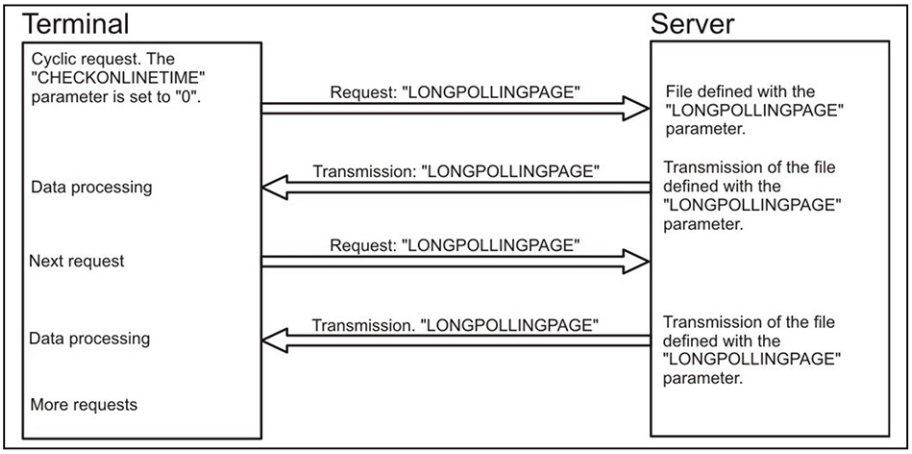
CHECKONLINETIME=0	Activation of Longpolling The file defined by the LONGPOLLINGPAGE parameter is requested via GET.
CHECKONLINETIME=>0	Normal polling The file defined by the CHECKONLINEPAGE parameter is requested via GET.

4.13.2 Definition of the "LONGPOLLINGPAGE" variable

In the terminal, the LONGPOLLINGPAGE variable is implemented. The variable is entered in the b-client_xml.ini file. By default the LONGPOLLINGPAGE variable has the value "long_polling.xml". The value can be changed. You have to make sure that the host carries out the longpolling treatment for the page defined by this variable. Depending on the implementation on the host, this can be, for example, an ASP or even a servlet.

The variable specifies the name of the XML file requested during Longpolling by the terminal from the host via HTTP(S)/GET. All XML elements supported by the terminal can be entered in this file by the host.

The host may reply to a GET only it has new data. Since the terminal requests the next GET directly after a data transfer, this would generate unnecessary network load due to "empty data".



4.14 Booking Request

By use of the system variable HTTP_REQUESTMETHODE_FOR_BOOKING it is possible to define the HTTP request (GET or POST) that is used to transmit data records from the terminal. This is valid for all data records, not only for booking data records.

The system variable HTTP_REQUESTMETHODE_FOR_BOOKING can be used in the .ini file and in the configuration file. The system variable is stored in the b-client_xml.ini file in the [Common] section. Permitted values are:

- GET
- POST

The default value is GET.

To transmit data for a certain RT/RTM via the POST method it is necessary to have a relating entry for this RT/RTM in the translation table that contains the keyword "DATA".

Further information about data coding can be found in chapter [▶ 4.15] . For the data format, there is no difference if they are sent in the URL of the GET or in the data segment of the POST request.

Example 1: Definition for A0 recording record

```
<A0>
<URL=http://$(ONLINEIP)/Hostsimulation/BookingsServlet?
GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)&RTRTM=$(BPA,4,2)
&TIMESTAMP=$(BPA,7,10)&DATA=$(BPA,18)/>
</A0>
```

Data in the POST in this example:

&DATA=0000000001984

Example 2: Definition for parameter upload record:

```
<>3>
<URL=http://$(ONLINEIP)/Hostsimulation/ParameterUploadServlet?
GID=$(GID)&DID=$(DID)&MODE=$(BPA,3,1)&RTRTM=$(BPA,4,2)&DATA=$
(BPA,1)/>
</>3>
```

Data in the POST in this example:

&DATA=FF%20%3E3F0103110021B10000000000000000000

All entries defined after this keyword are transmitted in the data segment of the HTTP POST. If the keyword "DATA" is not set, the data segment of the HTTP POST remains empty, the content length then is 0.

4.15 XML: Data coding in requests

The format of data record results from the definition in the translation file; see example in chapter . With post-processing of the data it must be taken into account that the terminal application performs a so-called "percentage coding" of all characters with the exception of the non-reserved characters.

Table of the not-reserved characters

Hex value	Character
41-5A	Characters "A"-"Z"
61-7A	Characters "a"-"z"
30-39	Characters "0"-"9"
2D	"-"
2E	"."
5F	"_"
7E	"~"

The following applies
Coding is done for all characters which

- are other characters than "not reserved" (reserved or language-related special characters) and
- occur in the so-called "Query block" of the recorded data record, i.e. after the question mark and
- occur **within the recorded data.**

These characters are converted "percentage-coded" that means the character is represented by specification of its hexadecimal values with prefixed "%" character.

XML: Examples of a recorded data record:

Example 1:

http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&RTRTM=B1
&TIMESTAMP=0801171523&ID=020&00000)001

- In the represented URL request the query block starts with the question mark in front of the GID.
- In the recorded data the characters "&" and ")" are displayed
- Coding: & is coded: %26;) is coded: %2A

The character "&" occurring in the character string "&" is **not** converted as it is not within the recorded data.

Result:

http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&SASAM=B1&
TIMESTAMP=0801171523&ID=020%2600000%2A001

Example 2:

http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&SASAM=B1&
TIMESTAMP=0801171523&DATA=BB >9 (space and >)

Result:

http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&SASAM=B1&
TIMESTAMP=0801171523&Data=BB%20%3E9

4.16 XML: Booking responses

4.16.1 XML: Online responses

In "online" mode all data which were recorded with a booking are transmitted as XML page to the server by using the GET method. If the parameter HTTP_REQUESTMETHODE_FOR_BOOKING in the [Common] section of the b client_xml.ini file is set to POST, the transmission takes place via the POST method. In this process a definition is made in the translation table, which XML page is requested as booking response for this booking.

Example with the data record from chapter [▶ 4.15]:

```
http://12.12.50.158:8080/in.xml?GID=5&DID=5&MODE=0&SASAM=B1&
TIMESTAMP=0801171523&ID=0200000000001
```

In the example the XML page "in.xml" is requested as booking response for a booking with RT/RTM B1 (In).

The page "in.xml" might have the following contents:

```
<?xml version="1.0"?>
<dialog>
        <sysdef name="TERMREC" value="@@ R1Thank you"/>
        <sysvar name="SYSTIME" value="20070509190533"/>
</dialog>
```

The booking answer must be introduced with the tag **<dialog>** and terminated with the tag **</dialog>**.

- The **system variables** can be specified straight as variables in the **sysvar** objects by means of `name` and `value`.
- The **parameters** and **data records** of the terminal application are specified in the **sysdef** objects by means of the variables `TERMREC` and `value`. In this manner, requests e.g. for program number or master records to the terminal can be transmitted together with the booking answer, and text can be edited, etc.

If there is no definition for the corresponding RT/RTM stored in the translation file, the terminal application sends the data by means of the error page ERRORPAGE (default: `error.xml`) to the server, see chapter [▶ 4.20.1].

In this case, the behavior of the terminal application is the same as for a missing booking response. The terminal waits the time defined in the parameter "Response time-out" (approx. 12 seconds by default, see Response time out) and then decides on the authorization.

4.16.2 XML: Offline responses

The data arisen in "offline" mode are transmitted to the server by using the GET or POST method after the server is available again.

The XML page requested as booking answer is the same as in "online" mode because the booking answer is defined in the translation table by the RT/RTM.

If there is no definition for the corresponding RT/RTM stored in the translation file, the terminal application sends the data by means of the error page ERRORPAGE (default: `error.xml`) to the server, see chapter [▶ 4.20.1].

4.16.3 XML: Dialog responses

In dialog mode the dialog records are transmitted to the server by means of the GET or POST method. The requested XML page is described by a definition in the translation table, like for the function keys. The RT/RTM of the (G) dialog records are used for this.

```
<trans>
  <![CDATA[
      <G8>
          <URL=http://$(ONLINEIP)/g8.xml?GID=$(GID)
          &DID=$(DID)&SASAM=$(BPA,3,2)
          &DATA=$(BPA,5,2)/>
      </G8>

      <G1>
          <URL=http://$(ONLINEIP)/g1.xml?GID=$(GID)
          &DID=$(DID)&amp;1)&SASAM=$(BPA,3,2)
          &DATA=$ (BPA,5,5)/>
      </G1>
]]>
</trans>
```

The server transmits the XML page which is specified in the translation table as dialog answer.

The page "g8.xml" (see example definition) might contain the following:

```
<?xml version="1.0"?>
<dialog>
       <sysdef name="TERMREC" value="@@ G010000040000Enter Code"/>
</dialog>
```

The booking answer must be introduced with the tag **<dialog>** and terminated with the tag **</dialog>**.

- The **system variables** can be specified straight as variables in the **sysvar** objects by means of `name` and `value`.
- The **parameters** and **data records** of the terminal application are specified in the **sysdef** objects by means of the variables `TERMREC` and `value`. In this manner, requests e.g. for program number or master records to the terminal can be transmitted together with the booking answer, and text can be edited, etc.

If there is no definition for the corresponding RT/RTM stored in the translation file, the terminal application sends the data by means of the error page ERRORPAGE (default: `error.xml`) to the server, see chapter [▶ 4.20.1].

In this case, the behavior of the terminal application is the same as for a missing booking response. The terminal waits the time defined in the parameter "Response time-out" (approx. 12 seconds by default) and then decides on the authorization.

Info	Topic
4.20.1	XML: RT/RTM errors
3.9.6	Response time out

4.17 XML: File transfer

4.17.1 Overview

This chapter describes the transfer of files between a host (e.g. B-COMM) and a terminal of the NexT generation using the HTTP(S) communication channel. This function is supported from the following B-Client program versions:

- B-Client HR40 (K7-Android 12): 739-00-X-K01

An interface with the following properties is implemented in the HTTP(S) communication channel for this purpose.

- No restriction for the file format, see also chapter [▶ 4.17.1.1].
- The file transfer is always initiated by the host in response to a request from the terminal (longpolling request, check online or offline request, or a GET when transferring other data).
- Only one file transfer transaction is allowed at a time.
- The host works as an HTTP(S) server.
- Terminal and access manager work as HTTP(S) client.

4.17.1.1 Supported file formats

Data transfer via HTTP(S) basically supports all file formats. The main benefit of the function is to be able to transfer all files that are used by the clients in their context. The implementation was done without any restrictions, but not all possible file formats were tested.

When downloading, the "content-type" field in the response header should either be empty or set to "application/xml" or "text/xml".

The following file formats/encodings can be handled at least via this file transfer:

- . ini files (ASCII and Unicode inclusive BOM (**B**yte **O**rder **M**ark)
- Log files
- Image files
- Multimedia files (e. g. ogg format)
- Binary files
- Zip files
- Files without extension e. g."eepromsettings".

4.17.1.2 Allowed directories for file transfer

The following directories are permitted for file transfer on the client side.

Terminal

Full access to all directories of the application

- /data/data/com.apps.kaba/ba/...
- /data/data/com.apps.kaba/hr/...
- /data/data/com.apps.kaba/tp/...

Read access is possible to some other directories, but in general the rights are limited to the "root" user.

4.17.2 System variables in the xmlconfig.xml file

For file transfer, new system variables are implemented that must be defined on the host server. These variables do not have to be present when the communication is initialized, they can be set at any time to start the desired action.

Please note: The transfer direction (upload/download) is specified from the server's point of view.

- Upload = from client to server.
- Download = from server to client.

For more information and examples, see the following table.

<sysvar name="FILE_TO_DOWNLOAD" value="file name"/>

The file name must be specified with absolute path.

<sysvar name="FILE_TO_UPLOAD" value="file name"/>

The file name must be specified with absolute path.

<sysvar name="FILE_UPLOAD_PAGE" value="URL"/>

The format of the URL is analogous to the format of the system variables CHECKONLINEPAGE, LONGPOLLINGPAGE.

The URL can also contain the file name parameter as specified in the FILE_TO_UPLOAD variable.

Example:

<sysvar name ="FILE_UPLOAD_PAGE" value="http://$(ONLINEIP)/upload.xml?GID=$(GID)&DID=$(D ID)&TERMTYP=$(TERMTY P)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS) &FILE_TO_UPLOAD=$(FILE_TO_UPLOAD)"/>

<sysvar name="FILE_DOWNLOAD_PAGE" value="URL"/>

The format of the URL is analogous to the format of the system variables CHECKONLINEPAGE, LONGPOLLINGPAGE.

The URL can also contain the file name parameter as specified in the FILE_TO_DOWNLOAD variable.

Example:

<sysvar name ="FILE_DOWNLOAD_PAGE value="http://$(ONLINEIP)/download.xml?GID=$(GID)&DID=$(DID)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS)&FILE_TO_DOWNLOAD=$(FILE_TO_DOWNLOAD)"/>

System variables in the xmlconfig.xml file for file transfer

System variable	Possible values	Description

FILE_UPLOAD_PAGE	**Example 1:**
	`http://$(ONLINEIP)/b-comm-file-upload-page.xml`
	Example 2:
	`http://$(ONLINEIP)/b-comm-file-upload-page.xml/?&GID=$(GID)&DID=$(DID)&PROGNUM=$(PROGNUM)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS)`
		This parameter value must fit to the implementation on server side!
		The variable contains the URL of the POST to upload a file. The requested file is included in the response. The variable can be changed at any time.
		Must be set before initiating an upload with "FILE_TO_UPLOAD"

FILE_DOWNLOAD_PAGE	**Example:** `http://$(ONLINEIP)/b-comm-file-download-page.xml/?&FILE_TO_DOWNLOAD=$(FILE_TO_DOWNLOAD)`	This parameter value must fit to the implementation on server side! The variable contains the URL of the GET for the file to be downloaded. The requested file must be included in the response. The variable can be changed at any time. Must be set before initiating a download with "FILE_TO_DOWNLOAD".

FILE_TO_UPLOAD	**Example:**
	`<?xml version="1.0"?>`
	`<dialog>`
	`<sysvar name="FILE_TO_UPLOAD" value="/data/data/com.kaba.apps.ba/files/init/system.ini"/ >`
	`</dialog>`
		Absolute path and name of the file which shall be uploaded to the server.
		Must be send for every file to upload. Setting this variable will initiate the upload.

FILE_TO_DOWNLOAD	**Example:**
	`<?xml version="1.0"?>`
	`<dialog>`
	`<sysvar name="FILE_TO_DOWNLOAD" value="/data/data/com.kaba.apps.ba/files/init/system.ini"/ >`
	`</dialog>`
		Absolute path and name where the file will be stored on the terminal.
		Must be send for every file to download. Setting this variable will initiate the download

4.17.3 File upload: graphic overview

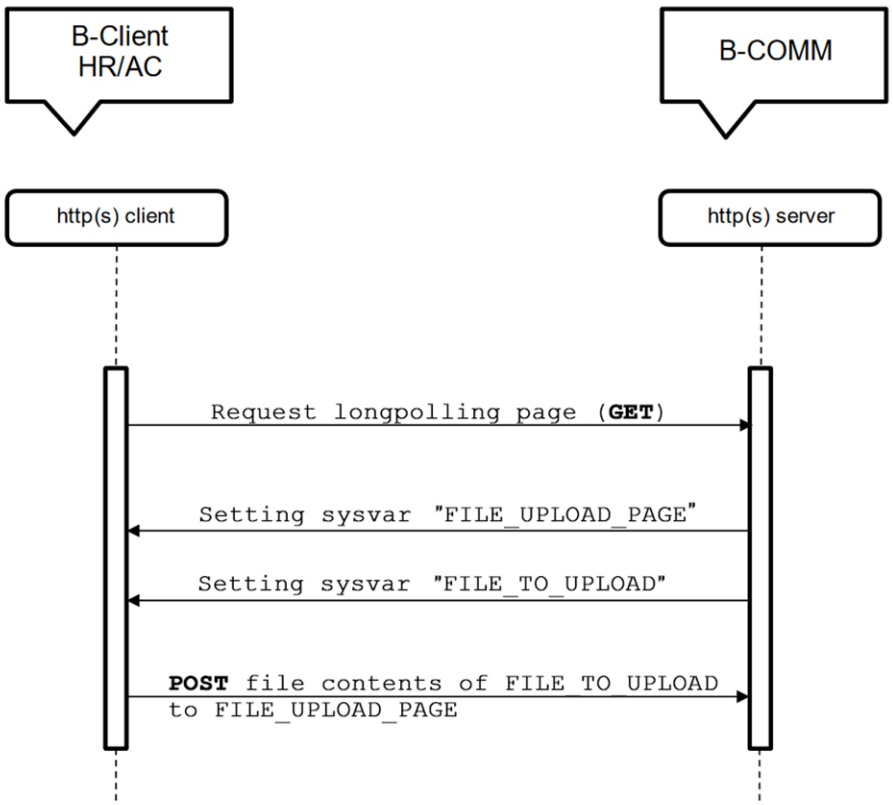
The following overview shows an example of the process of a file transfer from the client to the host (upload) with the "Longpolling" function. The process is the same for online or offline requests.

For more information on the topic of "Longpolling" see chapter [▶ 4.13].

For the file upload the following applies:

- If the file defined in the variable "FILE_TO_UPLOAD" is empty, an empty POST message is transmitted.
- If the file defined in the variable "FILE_TO_UPLOAD" does not exist, an empty POST message is also transmitted.

In both cases, details about this transfer can be viewed in the debug.log (data/data/com.kaba.apps.ba/files/diagnosis/debug.log). The debug level must be set to "5" for this, see chapter [▶ 14.1.3] debug.ini, section [Special].



4.17.3.1 Server response - Upload

A complete response of a server to a GET request could look like the following example:

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

<dialog>

<sysvar name="FILE_UPLOAD_PAGE" value="http://$(ONLINEIP)/b-comm-file-
upload-page.xml"/>

<sysvar name="FILE_TO_UPLOAD" value="/data/data/com.kaba.apps.ba/files/init/
system.ini"/>

</dialog>
```

The variable `sysvar` "FILE_UPLOAD_PAGE" must be noted with the name of the desired XML page of the processing server and can contain additional parameters as in this example:

```
http://$(ONLINEIP)/b-comm-file-upload-page.xml/?&GID=$(GID)&DID=$(DID)&
PROGNUM=$(PROGNUM)&SERIALNUMBER=$(SERIALNUMBER)&MACADDRESS=$(MACADDRESS)
```

The client sends the file content to this URL via HTTP POST. The server must ensure that the file content is stored in the desired file.

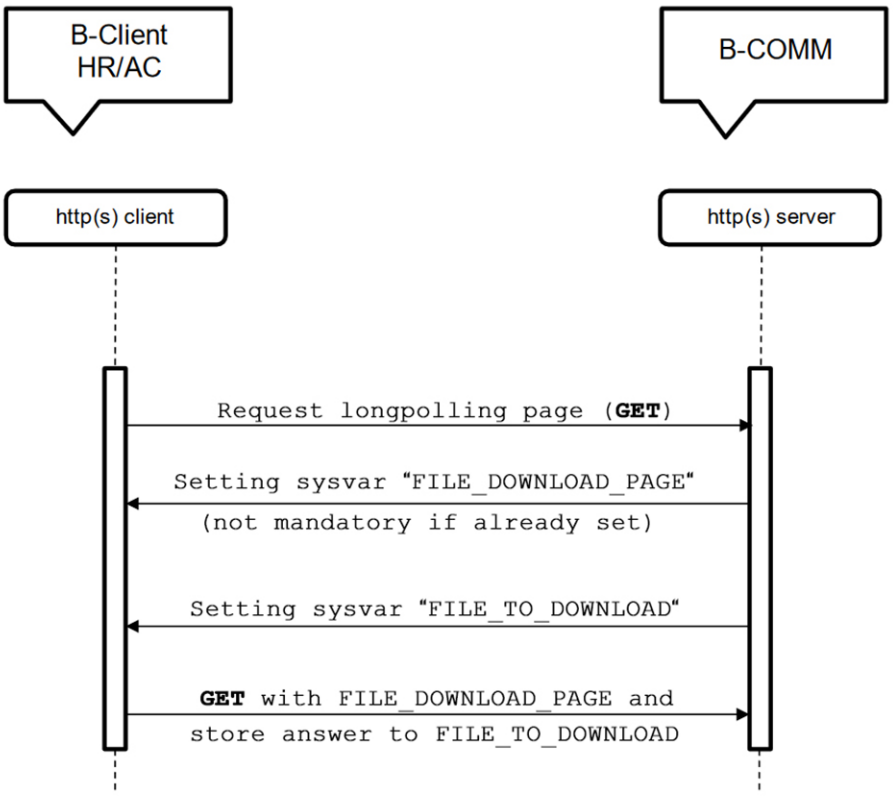
4.17.4 File download: graphic overview

The following overview shows an example of the process of a file transfer from the host to the client (download) with the "Longpolling" function. The process is the same for online or offline requests.

For more information on the topic of "Longpolling" see chapter [▶ 4.13].

For the file download the following applies:

- If the file defined in the "FILE_TO_DOWNLOAD" variable already exists, it will be overwritten.
- If the file defined in the "FILE_TO_DOWNLOAD" variable does not exist, it will be created.



4.17.4.1 Server response - Download

A complete response of a server to a GET request could look like the following example:

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

<dialog>

<sysvar name="FILE_DOWNLOAD_PAGE=http://$(ONLINEIP)/b-comm-file-download-
page.xml/?&FILE_TO_DOWNLOAD=$(FILE_TO_DOWNLOAD)"/>

<sysvar name="FILE_TO_DOWNLOAD" value="/data/data/com.kaba.apps.ba/files/
init/system.ini"/>

</dialog>
```

The client sends an HTTP GET request to the page defined by "FILE_DOWNLOAD_PAGE". The requested file is specified by the "FILE_TO_DOWNLOAD" system variable. The host returns the file in the response to the GET request.

The variable `sysvar` "FILE_DOWNLAD_PAGE" must be noted with the name of the desired XML page of the processing server and can contain additional parameters as in this example:

```
http://$(ONLINEIP)/b-comm-file-download-page.xml/?GID=$(GID)&DID=$
(DID)&PROGNUM=$(PROGNUM)&SERIALNUMBER=$(SERIALNUMBER)&FILE_TO_DOWNLOAD=$
(FILE_TO_DOWNLOAD)
```

4.18 Batch processing

Batch processing can be used for automated download of templates, parameters and master records.

XML files containing batch instructions have lower priority and require less computing power on the terminal side. This allows other processes such as bookings to be carried out in parallel to batch instructions, without too much load on system performance.

For batch processing, XML files are created on the application/web server which are transmitted via longpolling or normal polling.

The XML file must carry the <batch> tag at the beginning and the </batch> tag at the end as root element.

```
<?xml version="1.0" encoding="iso-8859-1"?>
 <batch>
   <dialog>
      <sysdef name="TERMREC" value="** D0deleting  master
      records .."/>
      <sysdef name="TERMREC" value="** Y1*******"/>
      <sysdef name="TERMREC" value="** D0processing  master
      records ..."/>
      <sysdef name="TERMREC" value="**
      Y000019830300098760011111111222222223333333344444444
      00000000000Persönliche Info
      11111111111111111111222222222222222222255555555
      66666666777777778888888899999999000000003"/>
      <sysdef name="TERMREC" value="**
      Y000000010100012340 0DspInf01DspInf02DspInf03DspInf0400
      000000000Persönliche Info Z01Persönliche Info Z02
      DspInf05DspInf06DspInf07DspInf08DspInf09DspInf100"/>
      <sysdef name="TERMREC" value="**
      Y000001520300098760011111111222222223333333344444444
      00000000000Persönliche Info
      11111111111111111111222222222222222222255555555
      66666666777777778888888899999999000000002"/>
      <sysdef name="TERMREC" value="**0D0processing of all
      records finished..."/>
   </dialog>
 </batch>
```

There will be no direct response of the terminal to state whether the download record has been processed successfully.

4.19 Maximum XML file size

When transmitting XML pages to the RAM of the terminal, the overall size of the XML page must be taken into account. Several factors have to be taken into account when determining the maximum file size of the XML pages.

- Memory extension of the terminal
- Configuration of the B-Client terminal software (e.g. software options)
- Installed apps
- Required memory capacity for the actual XML page held completely in the memory for parsing.

Since no absolutely valid size for the XML file can be derived from the above-mentioned conditions, dormakaba recommends a standard value for the size of the XML file.

The maximum size of an XML file should not exceed 2.5 MB.

4.20 XML: Troubleshooting

Troubleshooting is done by the XML error page which is defined in the system variable "ERRORPAGE" (default name: `error.xml`). The terminal requests the XML error page from the server when an error occurs. The recorded data are transmitted together with the request. For XML/HTTP communication 2 different error types are defined:

- Error in the RT/RTM definition of the translation table
- BPA error in the translation table.

After request of the error page the data record is deleted from the terminal buffer, no matter whether the server answered.

The terminal logs all errors which could not be treated by means of the "ERRORPAGE" in the debug.log file.

4.20.1 XML: RT/RTM errors

If the definition for RT/RTM is not found in the translation table or if it is faulty, the terminal application requests the error page as follows:

```
http://$(ONLINEIP)/$(ERRORPAGE)?GID=$(GID)&DID=$(DID)
&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)
&MACADDRESS=$(MACADDRESS)&DATA=$(BPA,1)
&ERROR=SASAMNotFound(<SASAM>)
```

In this process the recorded data are transferred with the keyword "DATA" starting with digit 1 (=$[BPA,1]). The request contains an additional descriptive text (ERROR=RTRTMNotFound) and an indication on the RT/RTM that initialized the error.

Example:

```
http://123.0.0.1:8080/error.xml?GID=5&DID=5
&TERMTYP=752-02-X-K01&SERIALNUMBER=0723233
&MACADDRESS=0C-FF-34-5A-23-1B
&DATA=@@ A000712051537002000000000013
&ERROR=SASAMNotFound(A0)
```

4.20.2 XML: BPA errors

If an error occurs while coding the recording data with the keyword "BPA" (e.g. the specified digit does not exist), the terminal application requests the error page as follows:

```
http://$(ONLINEIP)/$(ERRORPAGE)?GID=$(GID)&DID=$(DID)
&TERMTYP=$(TERMTYP)&SERIALNUMBER=$(SERIALNUMBER)
&MACADDRESS=$(MACADDRESS)&DATA=$(BPA,1)
&ERROR=BPAError(<BPA>)
```

In this process the recorded data are transferred with the keyword "DATA" starting with digit 1 (=$[BPA,1]). The request contains an additional descriptive text (ERROR=BPAError) and an indication on the BPA definition that initialized the error.

Example:

```
http://123.0.0.1:8080/error.xml?GID=5&DID=5
&TERMTYP=752-02-X-K01&SERIALNUMBER=0723233
&MACADDRESS=0C-FF-34-5A-23-1B
&DATA=@@ A000712051537002000000000013
&ERROR=BPAError(BPA,-4,10)
```

5 Data records host to terminal

5.1 Display records

The display records allow to display any user information on the display. Depending on the record type modification, three different display records are available.

GID DID ON/OFF	Group address Device address Operating mode
D n	Record type Record type modification
n .. n	User information, 120 characters max. If the user information contains more than 120 characters, the super numerous characters are cut off.

Record type/Record type modification

RT/RTM	Meaning/description
D0	**Static display** A user information can be displayed independently from a booking. The user information is displayed as long as a new information, e.g. a logical booking response is displayed.
D1	**Dynamic display "Authorized"** A user information can be displayed independently from a booking. The user information is deleted from the touch screen after a defined time and the function key text is displayed again. The display time is defined with the parameter Display duration "Authorized".
D2	**Dynamic display "Not authorized"** A user information can be displayed independently from a booking. Additionally the beeper will be activated. The user information is deleted from the touch screen after a defined time and the function key text is displayed again. The display time is defined with the parameter "Display duration "Not authorized".

Info	Topic
3.9.9	Display duration "Authorized"
3.9.10	Display duration "Not authorized"

5.2 Controlling the terminal

5.2.1 Block/release bookings, local parameter setting and dialogs

The following data records enables the following functions according to the record type modification:

- Block bookings
- Release bookings
- Enable local parameter setting
- Block local parameter setting
- Block dialog
- Release dialog

GID	Group address
DID	Device address
ON/OFF	Operating mode
I	Record type
n	Record type modification

Record type/Record type modification

RT/RTM	Meaning/description
I0	**Release bookings** Bookings are possible at the terminal.
I1	**Block bookings** The terminal is blocked, no bookings are possible. If this data record is sent while a booking procedure is currently active, the booking can be finished. If the terminal is blocked, "Terminal gesperrt" (Terminal blocked) will be displayed. A device restart does **not** revoke the blocking. The terminal can also be blocked via an input. The input intended for this must be configured accordingly in the [IOMapping] and [InputFunctions] sections in the "application.ini" file. Both blocking features have the **same priority**.
I2	**Enable local parameter setting.** Local parameter setting at the terminal is enabled (default).
I3	**Block local parameter setting.** Local parameter setting at the terminal is blocked. A device restart does **not** revoke the blocking.

Info	Topic
14.3.1	"application.ini"

RT/RTM	Meaning/description
I6	**Release dialogs** Dialogs are possible at the terminal.
I7	**Block dialogs** The dialog function is blocked. If this data record is sent while a dialog procedure is active, the current dialog can be finished. If the dialog is locked, the corresponding function key can no longer be selected. A device restart does **not** revoke the blocking.

5.2.2 Controlling the relays

With the following data record the relays of the terminal can be activated independently from a booking. If a relay is activated dynamically, the operation time is defined by the "Relay operation time" parameter.

Depending on the desired use case, the relay can be assigned via the sections [IOMapping] and [OutputFunctions] in the "application.ini" file.

GID DID ON/OFF	Group address Device address Operating mode
J n	Record type Record type modification
0	Relay number. Only "0" permitted.

Record type/Record type modification

RT/RTM	Meaning/description
J0	**Set relay statically** The relay remains operated until it is reset via a J2 record
J1	**Set relay dynamically** The relay will be activated. The operation time is defined via the "Relay operation time" parameter.
J2	**Reset relay** A relay which was activated by a J0 record can be reset by a J2 record.

Info	Topic
3.9.11	Relay operation time
14.3.1	"application.ini"

5.2.3 Operating mode, recording data and program version changing/ requesting

According to the record type modification, several operations can be carried out in the terminal.

GID	Group address
DID	Device address
ON/OFF	Operating mode
T	Record type
n	Record type modification, see following description

Record type/Record type modification

RT/RTM	Meaning/description
T0	**Switch terminal to "Online" mode** In the online operating mode, the terminal has a permanent data communication to the computer. That means, generated data records are immediately transmitted to the computer. If a recording data record cannot be transmitted to the computer the terminal changes to operating mode offline. When the connection to the computer is set up again, the terminal returns to operating mode online in the moment when the computer requests for recording data records.
T1	**Switch terminal to "Offline" mode** In the offline operating mode, the generated data records are stored in the terminal. When the computer will set up the connection, all the stored data records are transmitted to the computer. Alarm records, error records and status records are transmitted immediately to the computer when a connection is set up.
T2	**Switch terminal to "Autonomous" mode** In the autonomous operating mode, the generated recording data records are stored in the terminal. The computer must request this data records with a T3 record. Alarm records, error records and status records are transmitted immediately to the computer when a connection is set up.
T3	**Request recording data in "Autonomous" mode** This data record requests the recording data records buffered in a terminal while in the autonomous operating mode.
T4	**Request terminal status** This data record requests the current terminal status. The terminal responses with a S7 record.
T8	**Request program number** This data record requests the terminal´s program number. The terminal responses with a SV record.

Info	Topic
7.3.3	S7 record
7.3.1	SV record

5.2.4 Request terminal info

The following data record allows to request technical information from the addressed terminal.

The response is a TI record "Report terminal info" containing the related information.

The measuring unit of the temperature (°Celsius, °Fahrenheit or ° Kelvin) is defined by the "Temperature=" parameter in the "[Measurement]" section of the "application.ini" file.

GID DID ON/OFF	Group address Device address Operating mode
T I	Record type Record type modification
n n n	Information identifier, see following description.

Information identifier

Identifier	Meaning/description
001	Request the temperature in the terminal
002	Request the device type
003	Request the display resolution

Info	Topic
7.3.8	Report terminal info
14.3.1	"application.ini" file

5.2.5 Request backup state

The following data record allows to request the backup state of the terminal. The backup state is used by the terminal to report which files have been changed since the last backup modifications.

The terminal sends an **St record** with the identification **B0** as a response to the computer.

GID DID ON/OFF	Group address Device address Operating mode
T t	Record type Record type modification
B 0	Identification for backup request record

Info	Topic
7.3.4	St record with identification B0

5.2.6 Delete backup status

The following data record allows to delete the backup status of the terminal. For this purpose the bit values for RESET1 and RESET2 are specified in the hexadecimal coding in which they have been reported by the terminal in the St record with identification B0.

The FTP server **cannot** be disabled by means of this deletion record.

GID DID ON/OFF	Group address Device address Operating mode
T t	Record type Record type modification
B 1	Identification for backup deletion record
n	Bit values RESET1 The coding corresponds to BUSTAT1
n	Bit values RESET2 The coding corresponds to BUSTAT2

Info	Topic
7.3.4	St record with identification B0
3.5.1	Hexadecimal coding

5.2.7 Set terminal volume

With the following data record the volume of the terminal´s speaker can be set. Values between "00" (speaker off) and "15" (maximum volume) are accepted.

The default value is "08".

A reboot of the teminal is not necessary. The volume will be set immediately after transmitting the data record.

GID DID ON/OFF	Group address Device address Operating mode
T V	Record type Record type modification
s	Identifier for **s**etting the value.
n n	Value range between 00 = speaker off and 15 = maximum volume.

Info	**Topic**
5.2.8	Request the current volume value by use of a TV record with identifier "g".
7.3.9	Report the volume value via TV record.

5.2.8 Request terminal volume

With the following data record the volume value of the terminal´s speaker can be requested.

The response is a TV record containing the currently set volume value.

GID DID ON/OFF	Group address Device address Operating mode
T V	Record type Record type modification
g	Identifier for **g**etting (request) the value.

Info	**Topic**
5.2.7	Set the current volume value by use of a TV record with identifier "s".
7.3.9	Report the volume value via TV record.

5.2.9 Request reader information

The following data record is used to request information on the following reader types:

- LEGIC
- Mifare

As identification for the reader, the number under which the requested reader is configured in the addressed terminal must be entered.

Possible values for the identification: 1 or 2.

The terminal sends an **St record** with the identification **Tn** (n= reader number) as a response to the computer.

For all other than the named reader types or readers that respond a "3" as an identifer, the terminal will send an O3 record.

GID	Group address
DID	Device address
ON/OFF	Operating mode
T	Record type
t	Record type modification
T	n = reader identification.
n	

Info	Topic
7.3.5	St record with identification Tn
7.2	Error records, O3 record

5.2.10 Request launch data

The launch data records 1 to 16 of the LEGIC reader are requested using the "Tt" request record. There is no request record which allows to request all launch data records at once. Therefore, the launch data records must be requested individually from the reader.

The terminal response is an "St" record.

GID	Group address
DID	Device address
ON/OFF	Operating mode
T	Record type
t	Record type modification
n	Number of the launch data record be read
n	01 to 16

Info	Topic
7.3.7	St record

5.2.11 Controlling the buffer memory

The following data records are only important if buffer backup is set in the terminal. The function of this data records depends on the record type modification, see table below.

GID	Group address
DID	Device address
ON/OFF	Operating mode
M	Record type
n	Record type modification, see following description.

Record type/Record type modification

RT/RTM	Meaning/description
M0	**Read buffer memory** All recording data records stored in the buffer memory of the terminal are transmitted to the computer again.
M1	**Delete buffer memory** All recording data records stored in the buffer memory and transmitted once to the computer are deleted in the buffer memory of the terminal.
M2	**Initialize buffer memory** All recording data records are deleted in the buffer memory of the terminal.

Info	Topic
3.9.1	Buffer backup

5.3 Logical booking response

After a booking in operating mode online, the terminal transmits the recording data record to the computer and is awaiting a logical booking response. The logical booking response can be transmitted to the terminal as a dialog-response or as a data record. If the terminal receives no logical booking response it changes to operating mode offline and displays the internal booking response "Danke" (Thank you).

GID DID ON/OFF	Group address Device address Operating mode
R (n)	Record type Record type modification, see following description.
n n	User information, 120 characters max.

Record type/Record type modification

RT/RTM	Meaning/description
R0	**Silent confirmation** The following data record can be transmitted to the terminal as a silent confirmation. The door opener relay is **not** activated. The entered user information is displayed. The display duration is defined with the "Display duration authorized" parameter.
R1	**Booking authorized** The following data record can be transmitted to the terminal after an authorized booking. The door opener relay is activated according to the "Select relay" parameter. The entered user information is displayed. If no user information is entered, the internal booking response "Danke" (Thank you) is displayed. The display duration is defined with the "Display duration authorized" parameter.
RU	**Authorized booking with individual operator's information** The data record is analogical to the R1 data record. The RU data record is used to interpret the operator's information it contains in UTF-8 format and display the corresponding Unicode characters. Application example: Display of characters which are not stored in the SpecialEncoding table, for example for the names of persons.

RT/RTM	Meaning/description
R2	**Booking not authorized** This data record can be transmitted to the terminal after an unauthorized booking. The door opener relay is **not** activated. The entered user information is displayed. If no user information is entered, the internal booking response "Nicht berechtigt" (Not authorized) is displayed. The display duration is defined with the "Display duration not authorized" parameter.
RV	**Unauthorized booking with individual operator's information** The data record is analogical to the R2 data record. The RV data record is used to interpret the operator's information it contains in UTF-8 format and display the corresponding Unicode characters. Application example: Display of characters which are not stored in the SpecialEncoding table, for example for the names of persons.
R3	**Repeat input** This data record can be transmitted to the terminal if the input shall be repeated. The booking is not authorized. The door opener relay is **not** activated. The entered user information is displayed. If no user information is entered, the internal booking response "Eingabe wiederholen" (Repeat input) is displayed. The display duration is defined with the "Display duration not authorized".

Info	Topic
3.10.14	Activate relay
3.9.9	Display duration "Authorized"
3.9.10	Display duration "Not authorized"

5.3.1 Booking authorized with display info (R4/R5/R6 record)

The following data record can be transmitted to the terminal after an authorized booking. The door opener relay is activated according to the "Select relay" parameter.

With the mailbox text no. which is entered in the data record the corresponding mailbox text is displayed. If the mailbox text number 0 is entered, the display info texts and its corresponding display info entered here are displayed, but no mailbox text. Further display info texts and the corresponding display info can be requested by pressing the "Enter"-key.

If neither mailbox text nor display info is entered, the internal booking response "Danke" (Thank you) is displayed. The display duration is defined with the "Display duration authorized" parameter. The response specific text can be used if no mailbox text number is entered and is displayed instead of "Thank You".

GID DID ON/OFF	Group address Device address Operating mode
R (n)	Record type Record type modification, see following description.
n	Mailbox text no. 0-F (for mailbox text no. 1-15)
n n	Display info 1; number of characters see following table
n n	Up to 9 more display info, each of X characters
n n	Response specific text; 40 characters

Record type modification and number of characters

RT	RTM	Characters in the display info
R	4	8
R	5	13
R	6	20

Info	Topic
3.10.14	Activate relay
3.9.9	Display duration "Authorized"

5.3.2 Prolongation of response time out

With the following data record the response time-out can be restarted. After that, the terminal awaits another logical booking response within the restarted response timeout.

GID DID ON/OFF	Group address Device address Operating mode
R 9	Record type Record type modification

5.4 Date/time setting

The following data record is used to set the date and the time in the terminal. If also the seconds have to be set, the RT/RTM U1 must be used.

GID DID ON/OFF	Group address Device address Operating mode
U (n)	Record type Record type modification, see following description.
Y Y M M D D	Date identification YY = 90 to 89 equals 1990 to 2089. If only the time is to be set, the date declaration can be omitted.
h h m m (s) (s)	Time declaration Defining the seconds only with the RT/RTM U1.

Record type modification and number of characters

RT	RTM	Setting the seconds
U	0	No
U	1	Yes

5.5 Master records

The master records are the fundament for all time recording tasks to be fulfilled using dormakaba terminals. All necessary data and parameters of the individual employee are defined in a master record.

In default applications, every single employee has got his own, individual master record.

Different master record may be stored in the terminal:

- Individual master records
- Pseudo master records

While checking a booking, an individual master record is searched first. If no individual master record is found, a pseudo master record is searched. The master record found first is used for checking. If the parameter "Individual master record required" is set, only the individual master record is searched.

Individual master record

This master record is assigned to an ID which has been read from a badge. All checks and authorizations contained in the individual master record are valid for this badge.

Pseudo master record

If no individual master record can be assigned to a badge the pseudo master record is assigned. Only one pseudo master record can be stored in the terminal.

Info	Topic
3.10.4	Individual master record required
5.5.1	Individual master record
5.5.4	Pseudo master record

5.5.1 Individual master record download

Format of the individual master record.

<table>
<tr><td>GID
DID
ON/OFF</td><td colspan="2">Group address
Device address
Operating mode</td></tr>
<tr><td>Y
0</td><td colspan="2">Record type
Record type modification</td></tr>
<tr><td>n
.
n</td><td colspan="2">ID no.
variable length 5 to 20 digits (default: 7 digits)
see following description</td></tr>
<tr><td>n</td><td>Version no.; see following description</td><td rowspan="8">The wildcard character * can be used in these parameters.
The current parameter value in the terminal will remain unchanged.</td></tr>
<tr><td>n
n</td><td>Employee status;
see following description</td></tr>
<tr><td>n
n</td><td>Profile name;
see following description</td></tr>
<tr><td>n
.
n</td><td>PIN no.;
variable length 1 to 10 digits (default: 4 digits)
see following description</td></tr>
<tr><td>0</td><td>Reserved 0</td></tr>
<tr><td>n</td><td>Mailbox text no.; 0-F (for mailbox text no. 1-15);
see following description</td></tr>
<tr><td>n
.
n</td><td colspan="2">Display info 1; 8 digits;
see following description</td></tr>
<tr><td>n
.
n</td><td colspan="2">Display info 2; 8 digits;
see following description</td></tr>
<tr><td>n
.
n</td><td colspan="2">Display info 3; 8 digits;
see following description</td></tr>
<tr><td>n
.
n</td><td colspan="2">Display info 4; 8 digits;
see following description</td></tr>
<tr><td>0
.
0</td><td colspan="2">11 digits reserved 0</td></tr>
<tr><td>n
.
n</td><td colspan="2">Personal text 40 characters
see following description</td></tr>
</table>

n . n	Text without defined meaning; 20 characters available e. g. for partner applications
n . n	Display info 5; 8 digits; see following description
n . n	Display info 6; 8 digits; see following description
n . n	Display info 7; 8 digits; see following description
n . n	Display info 8; 8 digits; see following description
n . n	Display info 9; 8 digits; see following description
n . n	Display info 10; 8 digits; see following description
n	Biometric status, see following description.

ID no.

The individual master record is valid for the badge with the ID number entered in the master record. The length of the ID number is adjustable from 5 to 20 digits. The setting can be done in the file data/data/com.kaba.apps.hr/files/init/application.ini.

The factory default setting is 7 digits.

It is possible to use numeric values as well as hexadecimal values. The use of a UID is possible up to 20 digits (= 10 byte UID). The conversion of a 10 byte UID to a numeric value (HexToDecASCII) is not possible. The conversion is restricted to 63 bit.

If the ID number has less than 7 digits, the character zero must be entered ahead, e.g. 0012345 with a 5 digit ID number.

The values 0 to 9 are allowed as ID number.

Please note, that ID numbers containing only 9 (99...99) are reserved for the pseudo master record.

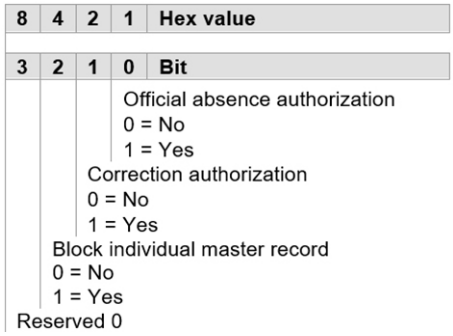
Version no.

The version number read from the badge must correspond to the ID number entered in the master record.

Employee status

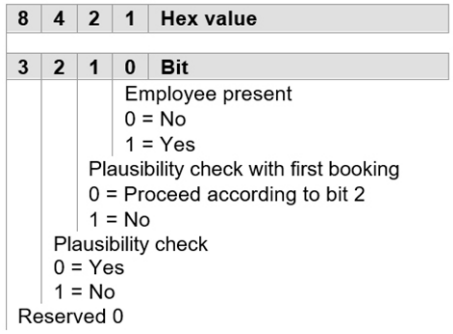
The employee status is transmitted in two bytes. The terminal interprets this characters entered in the download record as a hex code.

Employee status: 1. byte



Bit	Function
Bit 0	Assigns an official absence authorization to the employee. If an employee with no official absence authorization executes a booking on a function key which has the RT/RTM B3, B4 or B5 the error identification 7 "No official absence authorization" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Keine Dienstgangber." (No official absence authorization).
Bit 1	Assigns a correction authorization to the employee. If an employee with no correction authorization executes a booking on a function key which has the RT/RTM B6 the error identification 6 "No correction authorization" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Keine Korrekturber." (No correction authorization).
Bit 2	Blocks the individual master record. If the master record is blocked, the error identification 5 "Master record blocked" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Stammsatz gesperrt" (Master record blocked).
Bit 3	Reserved

Employee status: 2. byte



Bit	Function
Bit 0	Stating if the employee is present. This bit must be observed if the individual master record is changed during the working hours. After an upload of the individual master record, this bit is stating if the employee is currently present or not.
Bit 1	Defines if a plausibility check will be executed with the first booking after a download of the individual master record.
Bit 2	Defines if the plausibility check is executed continuously. If the plausibility check is set and a present employee executes an "IN" booking or an absent employee executes an "OUT" booking, the error identification 8 "In/out error (plausibility error)" is entered into the recording data record. In the "offline/autonomous" operating modes the booking is rejected with the message "Kommen/Gehen-Fehler" (In/out error).
Bit 3	Reserved

Profile name

To effect checks according to time profiles, the profile name of the desired time profile must be entered. The entry **00** means that no valid time profile is saved for this employee and a check for time authorization provides a negative test result. Irrespective of this entry, a check for time authorization **always** takes place if it is set in the "Function key functions".

PIN

In the time profile and/or the code-control-byte, the PIN check can be entered. During the PIN check, the PIN entered via the touch screen is compared with the PIN entered here in the individual master record. The number of digits is variable and can be set in the range from 1 to 10. The setting can be done in the file application.ini in the [MasterRecordDefinition] section. The factory default setting is 4 digits. The characters 0 to 9 are allowed entries for the PIN.

Mailbox text number and display info 1 to 10

These entries are only important in the following two cases. Additionally the entries display info 1 to 10 are only important if the display info in the master record is not suppressed.

- After an authorized booking in the operating modes offline/autonomous.
- After an authorized booking in operating mode online when an R4 to R6 record without mailbox text and without display info is transmitted to the terminal.

With the entered mailbox text number the corresponding mailbox text is displayed. The mailbox text is displayed once with the first booking after download of the master record. If the mailbox text number 0 is entered, no mailbox text is displayed. But display info texts and the corresponding display info which are entered in the master record, are displayed. More display info and their display info texts can be requested with a soft key.

Personal text

This entry is only important if the parameter show personal text is set. The personal text is shown instead of a positive answer ("Thank you").

Text without defined meaning

This text is intended for program expansions and is not used otherwise.

Biometric byte

This byte defines, whether the employee has the right to enter his badge number manually using the keypad or a badge.

If a badge number is entered no master record is defined for, the booking is rejected with the message "Kein Stammsatz" (no master record). If the function key parameter "Buffer not authorized recording data" is set, the booking is stored with the error identification 4 "Kein Stammsatz" (no master record)".

If a badge number is entered a master record is defined for but it´s biometric status is set to "0", the booking is rejected with the message "Nicht berechtigt" (not authorized). If the function key parameter "Buffer Not Authorized Recording Data" is set, the booking is stored with the error identification E "Keine Ausweisnummer Eingabe Berechtigung" (no badge number entry authorization)".

Biometric byte

8	4	2	1	Hex value
3	2	1	0	Bit
				Authorization for badge number entry via keyboard 0 = No 1 = Yes
				Authorization for badge number entry via badge 0 = No 1 = Yes
				Reserved 0
				Reserved 0

Info	Topic
14.3.1	application.ini file
5.5.4	Pseudo master record
3.17	Time profiles
3.10.9	Suppress display info in the master record
3.10.32	Show personal text

5.5.2 Individual master record delete/request

Depending on the record type modification, this data record can be used to delete or request individual master records.

GID	Group address
DID	Device address
ON/OFF	Operating mode
Y	Record type
n	Record type modification, see following description.
n	ID no.
..	Number of digits see description in chapter [▶ 5.5.1]
n	

Record type/Record type modification

RT/RTM	Meaning/description
Y1	**Delete individual master record** To delete individual master records, the ID number of the individual master record which shall be deleted must be entered. If **all** individual master records in the terminal shall be deleted, wildcards (* *.. * *) in the defined number of digits can be entered instead of the ID no. The pseudo master record is also deleted. Please use wildcard characters always very carefully. All individual master records are lost after use of wildcards.
Y8	**Request individual master record** To request individual master records, the ID number of the individual master record which shall be requested must be entered. If all individual master records in the terminal shall be requested, also wildcards (* * * ...) can be entered instead of the ID number.

5.5.3 End of download of individual master record

This data record indicates the end of the individual master records and must be transmitted to the terminal as an end identification after the download of individual master records.

GID	Group address
DID	Device address
ON/OFF	Operating mode
Y	Record type
9	Record type modification, see following description.

5.5.4 Pseudo master record download

Format of the pseudo master record.

GID DID ON/OFF	Group address Device address Operating mode	
Y 0	Record type Record type modification	
9 . 9	Pseudo ID; only cipher **9** permitted variable length 5 to 20 digits (default: 7 digits) see following description.	
0 0 0	3 digits reserved 0	The wildcard character * can be used in these parameters. The current parameter value in the terminal will remain unchanged.
n n	Profile name; see following description	
n . n	PIN no.; variable length 1 to 10 digits (default: 4 digits) see following description	
0	Reserved 0	
n	Mailbox text no.; 0-F (for mailbox text no. 1-15); see following description	
n . n	Display info 1; 8 digits; see following description	
n . n	Display info 2; 8 digits; see following description	
n . n	Display info 3; 8 digits; see following description	
n . n	Display info 4; 8 digits; see following description	
0 . 0	11 digits reserved 0	
n . n	Personal text 40 characters see following description	

n . n	Text without defined meaning; 20 characters available e. g. for partner applications
n . n	Display info 5; 8 digits; see following description
n . n	Display info 6; 8 digits; see following description
n . n	Display info 7; 8 digits; see following description
n . n	Display info 8; 8 digits; see following description
n . n	Display info 9; 8 digits; see following description
n . n	Display info 10; 8 digits; see following description

Pseudo no.

The length of the pseudo number is adjustable from 5 to 20 digits. The setting can be done in the file application.ini in the section [MasterRecordDefinition]. The factory default setting is 7 digits.

It is mandatory for the pseudo ID to fill up the defined numbers of digits only with the cipher 9, e.g. 9999999 for 7 digits.

Profile name

To effect checks according to time profiles, the profile name of the desired time profile must be entered. The entry **00** means that no valid time profile is saved for this employee and a check for time authorization provides a negative test result. Irrespective of this entry, a check for time authorization **always** takes place if it is set in the "Function key functions".

PIN

In the time profile and/or the code-control-byte, the PIN check can be entered. During the PIN check, the PIN entered via the touch screen is compared with the PIN entered here in the individual master record. The number of digits is variable and can be set in the range from 1 to 10. The setting can be done in the file application.ini in the [MasterRecordDefinition] section. The factory default setting is 4 digits. The characters 0 to 9 are allowed entries for the PIN.

Mailbox text number and display info 1 to 10

These entries are only important in the following two cases. Additionally the entries display info 1 to 10 are only important if the display info in the master record is not suppressed.

- After an authorized booking in the operating modes offline/autonomous.
- After an authorized booking in operating mode online when an R4 to R6 record without mailbox text and without display info is transmitted to the terminal.

With the entered mailbox text number the corresponding mailbox text is displayed. The mailbox text is displayed once with the first booking after download of the master record. If the mailbox text number 0 is entered, no mailbox text is displayed. But display info texts and the corresponding display info which are entered in the master record, are displayed. More display info and their display info texts can be requested with a soft key.

Personal text

This entry is only important if the parameter show personal text is set. The personal text is shown instead of a positive answer ("Thank you").

Text without defined meaning

This text is intended for program expansions and is not used otherwise.

Info	Topic
14.3.1	application.ini file
3.17	Time profiles
3.10.9	Suppress display info in the master record
3.10.32	Show personal text

5.5.5 Pseudo master record deletion and requesting

With the following data record the pseudo master record can be deleted in the terminal.

GID	Group address
DID	Device address
ON/OFF	Operating mode
Y	Record type
n	Record type modification, see following description.
9	Pseudo ID. Only cipher 9 permitted
..	Number of digits see description in chapter [▶ 5.5.1]
9	

Record type/Record type modification

RT/RTM	Meaning/description
Y1	Delete pseudo master record in the terminal.
Y8	Request the pseudo master record.

5.5.6 Pseudo master record download end

The following data record indicates the end of the pseudo master record. It must be transmitted to the terminal as an end identification after the download of the pseudo master record.

GID	Group address
DID	Device address
ON/OFF	Operating mode
Y	Record type
9	Record type modification

6 Parameter download/upload

6.1 Parameter upload request

With the following upload request the terminal parameters can be requested. For this purpose, the block identification of the requested parameters must be entered.

The parameters can also be requested by entering wildcards * instead of the block identification.

Example:

The entry T * * as a block identification will transmit **all** general texts to the computer.

GID DID ON/OFF	Group address Device address Operating mode
> 8	Record type Record type modification
n n n	Block identification

Info	Topic
3	Parameters and download records
13.6	Table of block identifications

6.2 Request entries of the ini-files

With the following upload request, entries in the ini-files of the terminal can be requested completely or partly.

If all .ini-files shall be requested which are entered in the .ini-file "interface.ini" in the [Upload] section, only the block identification must be used. Via the path and the file name, the desired .ini-file is explicitly requested. Also an explicit section and the parameter itself can be requested if it is specified.

Example

The following upload request "z00/data/data/com.kaba.apps.hr/files/init/communication.ini[Host]IPAddress=" means, only the parameter value for "IPAddress" is transmitted to the computer.

Any other .ini-files, sections or parameters are not transmitted to the computer.

GID DID ON/OFF	Group address Device address Operating mode
> 8	Record type Record type modification
z 0 0	Block identification
d a t a / .. x x . i n i	Path and file name (relative or absolutely) a) Relative (to the current directory) First character unlike / **Example** data/data/com.kaba.apps.tp/files/init/debug.ini b) Absolutely: First character / **Example** /files/init/debug.ini If the path or the file name is missing, all .ini files are requested.
[StartUp]	Name of the section in square brackets (mandatory). Example: [StartUp] If the name of the section is missing, all sections and their entries are requested.
MainClass=	Parameter Example: MainClass=

Info	**Topic**
3.20	Download record .ini files
14	.ini files.

6.3 End of parameter download

The following data record indicates the end of a parameter download. The data record must be transmitted to the terminal as an end identification after a download of parameters

This is particularly important when downloading .ini file entries. Otherwise, some of the parameters in the .ini file parameters will not be activated.

GID	Group address
DID	Device address
ON/OFF	Operating mode
>	Record type
9	Record type modification

7 Data records terminal to host

7.1 Recording data

After a booking the terminal generates a data record. The operating mode of the terminal determines when this data record is transmitted to the host.

In the operating mode Online the procedure is as follows:

- the recording data are checked according to the defined check criteria.
- the check result is entered as error identification into the data record.
- the data record is immediately transmitted to the host.
- the host produces a logical booking response and transmits it to the terminal specifying whether the booking is authorized or not.
- if the terminal does not receive any logical booking response from the host within the response time-out, it changes to the offline operating mode.

In the operating mode Offline and Autonomous the procedure is as follows:

- the recording data are checked according to the defined check criteria.
- the check result is entered as error identification into the data record.
- according to the check result, the terminal itself decides if the booking is authorized or not and allows or blocks the access.
- the data record is not transmitted to the host, it is handled according to the parameters "Buffer authorized recording data" and "Buffer not-authorized recording data".

If the booking record is extended, the number of booking records that can be saved is reduced correspondingly.

Info	Topic
5.2.3	Operating mode
13.5	Table of error identifications
5.3	Logical booking response
3.10.10	Buffer authorized recording data
3.10.11	Buffer not authorized recording data

GID DID ON/OFF	Group address Device address Operating mode
n n	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
FK	Error identification
n .. n	Badge content and keyboard data according to parameter setting.

Info	**Topic**
13.4	Table of clock identifications
13.5	Table of error identifications
3.9.8.1	Set seconds in data record
3.9.8.2	Set user language in data record

Record type/Record type modification

The record type/record type modification is defined by the "record type/record type modification" parameter. If the plausibility checks, the official absence authorization, and the correction authorization shall be used, the following record types and record type modifications must be maintained.

RT/RTM	Meaning
A0	Access record
A1	Access record "In"
A2	Access record "Out"
B1	Time recording record "In"
B2	Time recording record "Out"
B3	Business errand "Auto In/Out"
B4	Business errand "In"
B5	Business errand "Out"
B6	Correction
B0	Interrogation
BS	Start of break
BE	End of break
F0	Special function

Info	Topic
3.10.25	Record type/Record type modifications

7.2 Error records

The error records **O1 to O3** contain the data record which has triggered the error record.

A list of the software options for which there is no license is added to the error record **O5**.

GID DID ON/OFF	Group address Device address Operating mode
O (n)	Record type Record type modification, see following description.
n .. n	Data record including RT/RTM which triggered the error record

Record type/Record type modification

RT/RTM	Meaning/description
1	Record contents false
2	Command cannot be executed at the moment
3	Command non-executable
5	No license

7.3 Status records

7.3.1 Reporting program number

The terminal transmits the following data record after the request from the computer with a T8 record "Request program no."

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S V	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
n .. n	Program number; 12 digits
.. .. ..	Reserved. 12 digits blanks.
n .. n	Serial number; 12 digits

Info	Topic
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
13.4	Table of clock identifications
5.2.3	T8 record

7.3.2 Buffer memory empty/buffer memory full

With the following data record the terminals reports to the computer whether the buffer memory is empty or full. The data record "Buffer memory empty" is only transmitted to the computer if the buffer memory content has been read out completely.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S n	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.

Record type/Record type modification

RT/RTM	Meaning/description
S0	Buffer memory empty
S2	Buffer memory full

Info	Topic
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
13.4	Table of clock identifications

7.3.3 Report terminal status

The terminal transmits the following data record after request from the computer with a T4 record.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S 7	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
SYSTAT1 SYSTAT2 SYSTAT3 SYSTAT4 SYSTAT5 SYSTAT6 SYSTAT7 SYSTAT8 SYSTAT9	System statuses see following description

SYSTAT1

The system status SYSTAT1 can represent 2 values.

After the terminal has reported its status to the host computer, the SYSTAT1 status is set to 0.

Entry	Meaning
0	Received data records have been acknowledged.
4	The computer transmitted a data record with unknown record type, record type modification or block identification.

SYSTAT2

The system status SYSTAT2 can represent 4 values.

Entry	Meaning
0	Bookings are not blocked currently, no booking is active
1	Bookings are blocked currently, no booking is active
2	Bookings are not blocked currently, a booking is active
3	Bookings are blocked currently, a booking is active

SYSTAT3

The system status SYSTAT3 represents the status of the relay.

Entry	Meaning
0	Relay not activated.
1	Relay activated.

SYSTAT4

The system status SYSTAT4 represents the status of the buffer memory.

Entry	Meaning
0	Buffer memory empty
1	Buffer memory not empty
3	Buffer memory full

SYSTAT5, SYSTAT6, SYSTAT7

The system statuses SYSTAT5, SYSTAT6 and SYSTAT7 are intended for extensions. A "0" is entered into the data record.

SYSTAT8

The system status SYSTAT8 represents the operating mode of the terminal.

Entry	Meaning
0	Operating mode Online
1	Operating mode Offline
3	Operating mode Autonomous

SYSTAT9

The description of SYSTAT9 can be found in chapter.

Info	Topic
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
13.4	Table of clock identifications
5.2.3	T4 record

7.3.4 Report backup state

The terminal records the changes for the selected files and operations if compared to the last message of the backup state. If this is the case, the corresponding hexadecimal value is written in the data record and reported to the host.

If there have been several changes, the corresponding hexadecimal values are added. The hexadecimal values are listed in 3 state descriptions BUSTAT1 to BUSTAT3, see the following example.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S t	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
B 0	Identifier for "Report backup state".
BUSTAT1	Hexcoded value for BUSTAT1
BUSTAT2	Hexcoded value for BUSTAT2
BUSTAT3	Hexcoded value for BUSTAT3

Info	Topic
5.2.5	Request backup state
5.2.6	Delete backup state
3.5.1	Hexadecimal coding
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record

BUSTAT1

8	4	2	1	Hex value
3	2	1	0	Bit
				/com.kaba.apps.hr/files/init
				/com.kaba.apps.ba/files/images
				/com.kaba.apps.hr/files/data
				/com.kaba.apps.ba/files/audio

BUSTAT2

8	4	2	1	Hex value
3	2	1	0	Bit
				/com.kaba.apps.ba/files/init
				/com.kaba.apps.hr/files
				Reserved 0
				Reserved 0

BUSTAT3

8	4	2	1	Hex value
3	2	1	0	Bit
				SFTP server active
				Reserved 0
				Reserved 0
				Reserved 0

Example:

The following files/states have been modified at a terminal:

- /com.kaba.apps.hr/files/init
- /com.kaba.apps.hr/files/data
- /com.kaba.apps.ba/files/audio
- /com.kaba.apps.hr/files
- SFTP server is active.

The following hexadecimal values in the data record result from these conditions:

- BUSTAT1: D_{hex} [Bit 3 = 1_{hex} + Bit 2 = 1_{hex} + Bit 0 = 1_{hex} ---> = D_{hex}]
- BUSTAT2: 2_{hex} [Bit 1 = 1_{hex}]
- BUSTAT3: 1_{hex} [Bit 0 = 1_{hex}]

The states are deleted again by transmitting the deletion record "Delete backup state" with identical hexadecimal values to the terminal. The SFTP server **cannot** be disabled by means of this deletion record.

7.3.5 Report reader information

The terminal sends the following data record as a response to an inquiry with a Tt record. The data record includes the number of the requested reader, as configured in the addressed terminal.

The reader information includes, among others, the boot loader version and the version specification of the reader operating systems.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S t	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
T n	Reader identification (reader ID) n= 1 or 2
n n	Reader type 03 = LEGIC 04 = MIFARE
n .. n	Reader information For internal use only

Info	**Topic**
5.2.9	Request reader information
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record

7.3.6 Report result of VCP processing

The terminal sends the following data record after a VCP processing in the mobile access context. The data record contains a result in form of a value (0 or 1). The included 2-digit status value is intended for internal purposes only.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S t	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
V R	Identifier for "Report result of VCP processing"
n	Result, see following table.
n n	2 digits status information. For internal use only.

Result	**Meaning**
0	VCP processing successful.
1	VCP processing not successful. Check of the VCP configuration required.

Info	**Topic**
9	Bookings via smartphone
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record

7.3.7 Report launch data

The launch data records available in the terminal are reported back by means of this status record.

The status record is requested by means of the data record "Requesting launch data" (Tt record).

GID DID ON/OFF/AUTO	Group address Device address Operating mode
S t	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day.
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
n n	Number of the read launch data record. 01 to 16
n .. n	Launch data (for internal use only).

- The request is acknowledged using an O3 record (command not executable) in case the reader system is not a LEGIC reader.
- The request is acknowledged using an O2 record (command currently not executable) in case a booking is being performed currently at the reader.

Info	Topic
5.2.10	Requesting launch data
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
13.4	Table of clock identifications

7.3.8 Report terminal info

The terminal sends the following data record as a response to an inquiry with a TI record "Request terminal info". This data record is used to request technical information from the addressed terminal.

The following technical information is reported:

- Identifier 001: Temperature values in the terminal
- Identifier 002: Device type
- Identifier 003: Display resolution.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
T I	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
n n n	3 digits information identifier, see following description.
.. .. ..	Terminal info, see following description. Flexible length, up to 100 characters).

Information identifier 001

With the Information identifier 001 the terminal´s temperature values are reported.

This terminal generation is equipped with three temperature sensors (CPU, PoE and Power management).

The reported message therefore contains measurement values from all three temperature sensors. The degree character (°) will not be transmitted in the data record due to possible incompatibilities with the used character sets.

Depending on the defined measurement unit, the following texts may occur:

CPU=65C;PoE=63C;Powermanagement=60C //Message in °Celsius

CPU=338K;PoE=336K;Powermanagement=333K //Message in °Kelvin

CPU=149F;PoE=145F;Powermanagement=140F //Message in Fahrenheit

The measuring unit of the temperature (°Celsius, °Fahrenheit or ° Kelvin) is defined by the "Temperature=" parameter in the "[Measurement]" section of the "application.ini" file.

If no temperature values could be found, an O2 error record (command currently not executable) will be generated instead of the TI record.

After starting the HR client, the current temperature value will be written to the debug.log file with a delay of approx. 10 seconds. For analysis purposes the temperature values can be written to the debug.log file hourly. To do so, the debug level must be set to "1".

If the temperature shall be requested more often during runtime, this can be done via a TI record with time scheduled orders, e. g. via B-COMM.

Information identifier 002

With the Information identifier 002 the device type of the addressed terminal is reported.

Possible values:

- ONE-K7
- 9600-K7
- 9700-K7
- 9800-K7

Information identifier 003

With the Information identifier 003 the display resolution of the addressed terminal is reported.

Possible values:

- 1280x800
- 1280x720
- 800x1280

Info	Topic
5.2.4	Request terminal info
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
7.2	O2 error record
14.3.1	"application.ini" file

7.3.9 Report terminal volume

The terminal sends the following data record after a request from the host with the TVg record "Request terminal volume". This data record contains the currently set volume value in the requested terminal.

GID DID ON/OFF	Group address Device address Operating mode
T V	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
n n	Currently set volume value. 00 to 15.

Info	Topic
5.2.7	Setting the current volume value by use of a TV record with identifier "s".
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
5.2.8	Requesting the current volume value by use of a TV record with identifier "g".

7.4 Alarm records

The terminal sends an alarm record in the following cases.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
X n	Record type Record type modification, see following table.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.

RT/RTM	Meaning/description
X1	**Door open time too long** The door is supervised with a door frame contact. An X1 data record is transmitted if the door is not closed after expiry of the allowed maximum door opening time.
X2	**Door break open** The door is supervised with a door frame contact. An X2 data record is transmitted to the computer if the door was broken open.
X3	**Door closed again** The door is supervised with a door frame contact. An X3 data record is transmitted to the computer, if an open door is closed again after • expiry of the allowed maximum door-opening time or • after the door was broken open.
X4	**Sabotage contact loop opened** A closed sabotage contact loop was opened.
X5	**Opened sabotage contact loop closed again** An open sabotage contact loop was closed again.

Info	Topic
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record

7.4.1 Service alarm records: Temperature alarm

The service alarm records are produced under certain operating conditions of the terminal. In this case, the terminal sets an alarm identification into the data record and transmits it to the host when a connection exists. Currently only temperature alarms are implemented.

GID DID ON/OFF/AUTO	Group address Device address Operating mode
X S	Record type Record type modification, see following table
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m	Time Seconds (2 digits) and user language (6 digits) only if set as parameter.
n n n n	4 digits alarm identification, see following description
T e x t ...	Description of the alarm as a text; variable length up to 100 characters. see following description.

Info	**Topic**
3.9.8.1	Set seconds in data record
3.9.8.2	User language in data record
13.4	Table of clock identifications

Alarm identifier	Meaning/description
0002	Temperature in the terminal too high. The maximal permitted temperature value has been reached or exceeded at at least one of the temperature sensors.
0003	Temperature in the terminal is within the permitted values again. Data records containing this identifier will be sent if the temperature has sunk to the lower limit again after it has exceeded the upper limit (data record with identifier 0002 was sent).

Description of the alarm as text

Depending of the measured temperature, the following texts may occur in the alarm record:

With identifier	Meaning/description
0002	Device is overheated
0003	Device temperature normal.

Temperature value limits and hysteresis for 97 00 K6 terminal.

This terminal is equipped with three temperature sensors.

1. CPU

Limit value for sending an XS record with identifier 0002: 90° Celsius (194 Fahrenheit).

Limit value for sending an XS record with identifier 0003: 88° Celsius (190.4 Fahrenheit).

2. PoE module

Limit value for sending an XS record with identifier 0002: 75° Celsius (167 Fahrenheit).

Limit value for sending an XS record with identifier 0003: 73° Celsius (163.4 Fahrenheit).

3. Power management

Limit value for sending an XS record with identifier 0002: 80° Celsius (176 Fahrenheit).

Limit value for sending an XS record with identifier 0003: 78° Celsius (172.4 Fahrenheit).

7.5 Master records

7.5.1 Upload of individual master records

The terminal transmits the following data record after request from the computer with an Y8 record.

GID DID ON/OFF	Group address Device address Operating mode
Y 0	Record type Record type modification
n ... n	ID no., variable length, 5 to 20 digits Default: 7 digits
n ...	Content of the individual master record

Info	Topic
5.5.1	Individual master record
5.5.2	Request individual master record

7.5.2 Upload of pseudo master record

The terminal transmits the following data record after request from the computer with an Y8 record with the pseudo number 99...99.

GID DID ON/OFF	Group address Device address Operating mode
Y 0	Record type Record type modification
9 ... 9	Pseudo no. variable length, 5 to 20 digits Default: 7 digits. 9 **Only cipher 9 permitted!**
n ...	Content of the pseudo master record

Info	Topic
5.5.4	Pseudo master record
5.5.5	Request pseudo master record

7.5.3 End of upload of individual/pseudo master record

The following data record marks the end of the upload of individual/pseudo master records and will be transmitted by the terminal to the computer as an end identification.

GID	Group address
DID	Device address
ON/OFF	Operating mode
Y	Record type
9	Record type modification

7.6 Parameter upload/download

7.6.1 Parameter upload record

The terminal transmits the following data record after request from the computer with a >8 record.

GID DID ON/OFF	Group address Device address Operating mode
> 3	Record type Record type modification, see following description.
n n n	Block identification of the requested parameter record.
n ... n	Parameters depending on the request.

Info	Topic
6.1	>8 record
3	Data records and parameters

7.6.2 End of parameter upload

The following data record marks the end of a parameter upload and will be transmitted by the terminal to the computer as an end identification.

GID DID ON/OFF	Group address Device address Operating mode
> 9	Record type Record type modification

7.7 Dialog records (G dialog)

The requirements on a data exchange between a terminal and a superior evaluation system are various. This is true not only for the number of input steps but also for the texts and information to be displayed at the terminal. The static booking procedure can, therefore, not always cover all requirements.

For this reason, self-defined booking sequences can be generated by the user by use of so-called G dialogs. G dialogs base on special kinds of data records, containing the record type G. These data records are linked to function keys with the corresponding parameter settings. They allow a question and answer communication with the host.

If a function key has been defined as dialog key, all user inputs are performed in a dialog with the host (G dialog). The host will decide dynamically whether and what inputs are required and which information must be displayed for the user.

This extends the design options of the user dialog significantly.

7.7.1 Terminal settings for G dialog

The following settings must take place before G dialog records can be used:

- the desired function key must be defined as a dialog key.
- record type and record type modification must be assigned to the function key using the single function blocks.
- the terminal must be in the "Online" operating mode.

Info	Topic
3.10.30	Function key as dialog key
3.10.25	Single function blocks
5.2.3	Switch over the operating mode

7.7.2 G dialog procedure

The G dialog is started by pressing a function key at the terminal. The terminal sends a G8 record (dialog request) containing the record type and record type modification set for the function key.

After receiving the G8 record, the host will send the first G0 record (dialog mask) to the terminal. The G0 record contains information on:

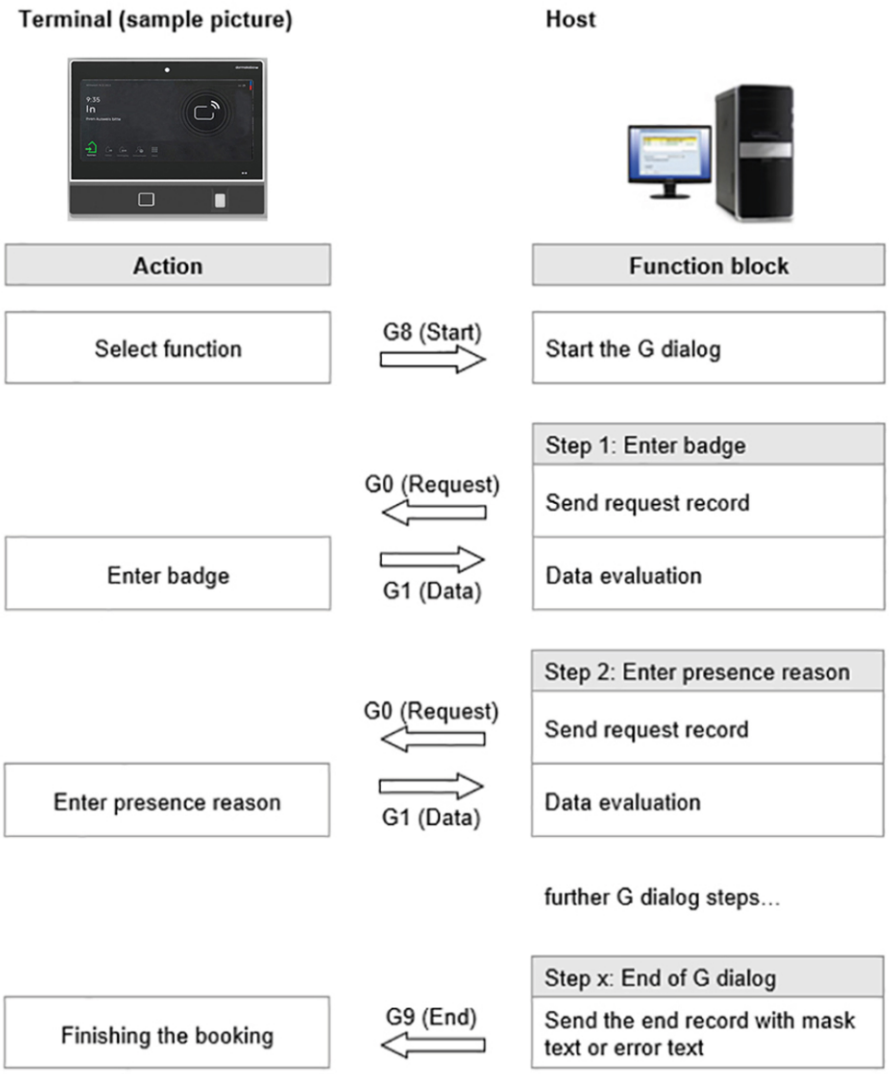
- the input value (type, number of characters, from which position, etc.)
- the reader type to be used (bar code, keyboard, etc.)
- entry, info and user guidance texts.

The terminal responds to the G0 request of the host by sending a G1 record (dialog input). The recorded input data are transmitted to the host according to the format specifications of the G0 record.

After evaluating the data, the host will decide whether further entries are required and, if necessary, send more requests to the host in the form of G0 records.

The dialog can be terminated either by the terminal or the host by sending the G9 record. The host will end the dialog if no more input steps are active or errors occur in the dialog. The terminal will end the dialog, for example, after the dialog key is pressed again by the user or after a time-out, etc.

7.7.3 G dialog example



7.7.4 Dialog request

If a function key is defined as dialog key and the dialog is not blocked, the terminal will transmit the following data record to the computer after the function key has been touched. After that, the terminal expects a G0-record from the host computer.

GID DID ON/OFF	Group address Device address Operating mode
G 8	Record type Record type modification
n n	Content: RT/RTM of the operated function key; see following description
(l) (l) (c) (c) (v) (v)	Identifier for user language Only if set as parameter.

Content: RT/RTM of the operated function key

The content is defined by the "Content of G8-record" parameter. It is either "G8" or the RT/RTM of the function key which has started the dialog request.

The RT/RTM of the function keys are defined by the parameters "record type" and "record type modification".

Identifier for user language

This tag contains the predefined or user chosen language. The tag will be transmitted only if the parameter "User language in data record" is set. The tag consists of the following parts:

- Language (l = Language)
- Country (c= Country)
- Version (v=Version)

Info	Topic
5.2.1	Block/release dialog
7.7.5	G0 record
3.10.25	Record type/Record type modification
3.9.8.2	User language in data record

7.7.5 Dialog mask

The terminal expects the following data record from the computer after a dialog request by means of the G8 record. The following data record can also be transmitted to the terminal without a dialog request. The dialog mask defines which inputs are required. The entered user information appears on the display.

The entries are transmitted from the terminal to the computer by means of a G1 record. After that it is either possible to transmit another G0 record or a G9 record to the terminal.

GID DID ON/OFF	Group address Device address Operating mode
G 0	Record type Record type modification
n	Keyboard entry; see following description
n	Input unit; see following description
n	Display badge content; see following description
n n	From digit with badge input; see following description
n n	Number of digits; see following description
n n	Position in the user information used for default value (00: no specification)
n	Type of the input field; see following description
n	Display of user information; see following description
n ..	User information (40 digits max.)
n ..	Extended user information (160 characters max.)

Keyboard/list input

This entry defines if a keyboard input is necessary.

Entry	Meaning
0	No keyboard input
1	Numeric keyboard input No alphanumeric default values are allowed here. If alphanumeric default values are specified, an O1 error record will be generated.
2	Alphanumeric keyboard input
6	List input
7	Dynamic list input Dynamic list via HTTP. The URL for the list is specified in the "Enhanced user information".
8	Dynamic list via HTTP. Time in the parameter "AutoConfirmationTimeout" is taken into account. Once the specified time has elapsed, the entered value is confirmed automatically. For dynamic lists see also chapter [▶ 7.7.8]. The URL for the list is specified in the "Enhanced user information".

Input unit

This entry defines if a reader input is necessary.

Entry	Meaning
0	No reader input
1 or 2	Reader 1 (for biometric reader no alternative reader permitted).
3	Reader 2 (optional)
4	Reader 1 biometric or alternative RFID reader

Display badge contents

This entry defines if the badge contents is displayed. The range limited by the entries "From digit with badge input" and "Number of digits" is displayed. The displayed data must be confirmed with the "OK" key.

Entry	Meaning
0	Do not display badge content
1	Display badge content

Info	Topic
7.7.4	G8 record
7.7.6	G1 record

From digit with badge input

This entry defines from which digit the badge contents is entered into the G1 record. If the badge contents shall be entered from the first digit, from digit with badge input must be set to 01.

Number of digits

For a badge input, this entry is used to define how many digits (01 to 40) of the read badge contents have to be entered into the G1 record. The G1 record has then a "6" as an identifier in the information byte, i.e. G16xx...

For a keyboard input, this entry defines how many digits (01 to 40) must be entered and then written into the G1 record. If the number of digits specified here is >00, first the data are entered completely until the number of digits is reached and the entry is confirmed with "Enter".After that, the entered data are applied and transmitted to the G1 record. The G1 record has then a "1" as an identifier in the information byte, i.e. G11xx...

The entry 00 causes that the first key press is entered into the G1 record and immediately transmitted to the computer. Each key press generates a G14 record with the additional contents of the corresponding key, for example G141, G142, G14A, G14B, etc.

Possible values:

- 01 to 40 (badge input)
- 01 to 40 (keypad input)
- 00: the first operated key will be entered into the G1-record and transmitted directly to the host. The entry must be confirmed with the "Enter" key.

Position within the user information used for default value

This parameter defines from which digit the characters of the user information will be entered as a default value into the input field.

Possible values:

- 01 to 40
- 00: no default.

Type of input field

Entry in the "keyboard input" field determines the type of the entry. This can be:

- via list selection (static or dynamic)
- via keyboard.

If the entry type "Lists" has been set, the desired list 1-15 can be selected here by entering 1-F.

If the entry type "Keyboard" has been set, the keyboard input can be specified here. At the beginning of the input, the cursor is positioned at the left. With each input, the cursor moves one digit to the right. When the input field is filled, the cursor is positioned at the rightmost digit.

If "Rolling keyboard input" has been set, the additionally entered characters are rolled through the display from right to left, see the following example a1. If fewer digits are entered as defined via "Number of digits",

- for numeric input: zeros or

- for alphanumeric input: blanks (20hex)

are entered into the G1 record as leading digits.

If "Fixed keyboard input" has been set, the cursor is also located on the left-hand side at the beginning of the input. With each input, the cursor moves one digit to the right. If other characters are entered, the right-hand character will be overwritten with the new character.

If fewer characters are entered than defined in the "Number of digits", an input error is displayed.

The following input options can be set:

Entry	Meaning
0	Rolling keyboard input; The characters entered are displayed
1	Fixed keyboard input; The characters entered are displayed
2 or 4	Rolling keyboard input; hidden input. The characters are displayed according to the Android setting in the parameter "Location & Security Settings: Visible Password".
3 or 5	Fixed keyboard input; hidden input. The characters are displayed according to the Android setting in the parameter "Location & Security Settings: Visible Password".

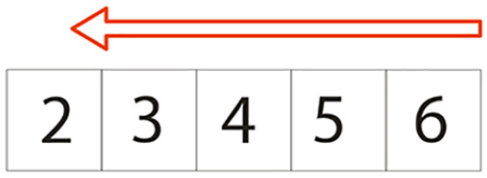
The following parameters affect the input:

- From digit
- Number of digits
- Type of input: rolling or fixed
- Default value defined or not defined

See the following display examples:

Display examples:

a) without default value, the maximum number of input digits reached.

Parameter	Representation
a1) Number of digits: 05 Type of the input field: 0 = - rolling representation - characters visible	 Representation of other characters:  The characters are rolled through the display from right to left.
a2) Number of digits: 05 Type of the input field: 1 = - fix representation - characters visible	 Representation of other characters:  The character on the right-hand side is overwritten with a new value.

G0 parameter record for the following display examples with a default value.

GID	Group address
DID	Device address
<Space>	Operating mode
G	Record type
0	Record type modification
1	Keyboard entry: Numeric
0	Input unit: No reader
1	Display badge content: Yes
0	From digit: 01
1	
0	Number of digits: 06
6	
0	Position in the user information used for default value. Value: 09
9	
0-3	Type of the input field; see examples
0	Display of user information: No acoustic signal
1 W	User information. Text: "Weight:<space>10."
2 e	
3 i	
4 g	
5 h	
6 t	
7 :	
8 <Space>	
9 1	
0	
.	

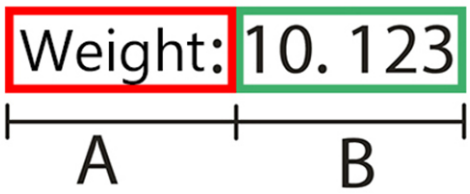
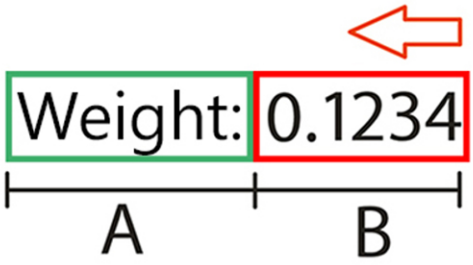
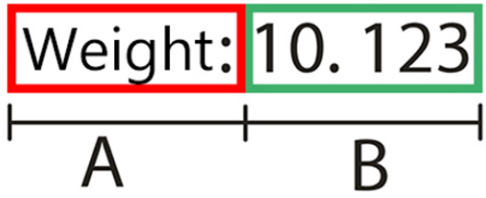
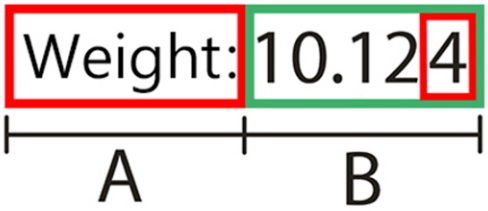
Explanation:

- From digit 01
- 6-digit input
- Default value: from digit 9 of the user information
- Type of the input field: see the following examples
- Default value is the text "10."

  Thus, these are already 3 digits of the total 6-digit input.

Display examples

b) With a default value for input of weight data.

Parameter	Representation
b1) Number of digits: 06 Type of the input field: 0= - rolling representation - characters visible User information in the data record: Weight:<blank> (=A) From digit 9 of the user information = default value: 10. Entered values: = B	 Representation of other characters:  The characters are rolled through the display from right to left.
b2) Number of digits: 06 Type of the input field: 1= - fix representation - characters visible User information in the data record: Weight:<blank> (=A) From digit 9 of the user information = default value: 10. Entered values: = B	 Representation of other characters:  The character on the right-hand side is overwritten with a new value.

Acoustical signalization

This entry defines, if the user information is displayed with an additional acoustic signal.

Entry	Meaning
0	No acoustic signal
1	User information with alternating acoustic signal.
2	User information with constant acoustic signal.

User information

Displaying of text up to 40 characters.

Extended user information

Additional output of text up to 160 characters in a separate window. When using a browser, a variable part of the URL to be called can be defined in the extended user information.

This may be used to call web pages user-specifically via the badge input.

Info	Topic
7.7.8	Dynamic lists

7.7.6 Dialog input

The terminal transmits the following data record to the computer after a G0 record has been received and when the required data has been entered. Afterwards, another G0 record or a G9 record can be transmitted to the terminal.

GID DID ON/OFF	Group address Device address Operating mode
G 1	Record type Record type modification
n	Information byte see following description
n ... n	Input data

Entry	Meaning	Input data in G1 record
0	Input was aborted with "C" key	No input data
1	Input was completed with "E" key	In accordance to user input
2	Key "->" was operated	No input data
3	Key "<-" was operated	No input data
4	Numeric key was operated	Cipher of the operated key
5	Operation timeout expired	In accordance to user input
6	Reader input	In accordance to user input

The entries 2, 3 and 4 can only appear if keyboard input with number of digits = 00 has been set in the terminal with a G0 record.

Info	Topic
7.7.5	G0 record
7.7.7	G9 record

7.7.7 Dialog end

The following data record signals the end of the dialog. This data record must be transmitted to the terminal after the end of the dialog to perform an end identification. The entered user information is displayed. If no user information is entered, the internal booking response "Dialog end" is displayed.

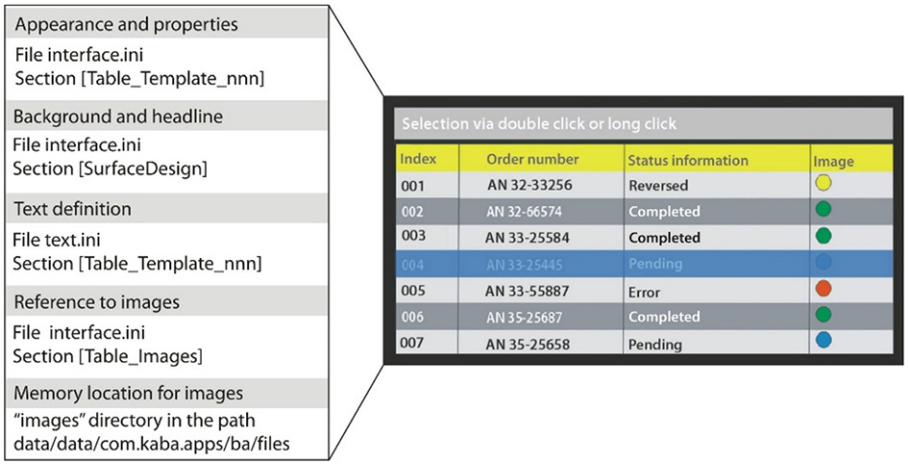
GID	Group address
DID	Device address
ON/OFF	Operating mode
G	Record type
9	Record type modification
n	User information (120 characters max.)
...	
n	

7.7.8 G dialog including dynamic lists

Dynamic lists in tabular form make data selection at the terminal easier for the person booking. After pressing a function key, the corresponding function key function will start a G dialog. In this G dialog, the HTTP address of the desired list is communicated to the terminal.

The labeling and the on-screen display can be edited by using an editor in the B-COMM communication software. Alternatively, the display can also be adjusted at the terminal via entries in the .ini files. All possible parameters and their settings are listed below by means of an exemplary list.

1. Overview of the definition and parameter files involved



2. Definition of a [Table_Template_nnn] section in the interface.ini file

A [Table_Template_nnn] section contains all parameters for the appearance and properties of a dynamic list. The extension "nnn" must be the same as the extension of the list the template is to belong to.

Example:

[Table_Template_002] for list <List name>.002.

If different lists must have the same appearance, they can be linked to the same section via the extension.

Example:

[Table_Template_002] for list <List name1>.002.

[Table_Template_002] for list <List name2>.002.

In the delivery state, one [Table_Template_999] section is provided in the interface.ini file. This section serves as default table definition for internal lists, e. g. selection list for official absence.

The [Table_Template_999] section can also be copied and renamed and thus used as template for customized definitions.

A customized table definition must be generated as a section named [Table_Template_001] to [Table_Template_020]. This allows a maximum of 20 different customized tables to be defined.

Section 7 of this chapter contains an example table with numbering. On the basis of the assigned numbers in the table, the parameterization of the table can be determined. Further information on parameters (e.g. colors, alignment and the like) can also be found in chapter 12 "Notes".

The wild cards "nnn" or "nn" in the parameter description must have been assigned the corresponding numeric extension, e.g. 001 or 03.

Parameterization takes place as follows.

File interface.ini, section [Table_Template_nnn]

Parameter	**Meaning**
[Table_Template_nnn]	Name of the template section. The numeric extension (nnn) must be the same as the extension of the list the template is to belong to.
Template_Name_de= Template_Name_en=	Name of the template, also language-dependent. The entry is for overview only and can also be omitted.
Input_Column=	Indication of the column for data transfer. Data entered in the defined column is transferred to the registration record. The possible value depends on the defined number of columns.
Number_Of_Rows=	Indicates how many rows are to be displayed at the same time (no. 6).
Selection_Type=	Indicates how many rows can be selected simultaneously (no. 8). Possible values: • none (no selection possible) • single (one row possible) • multi (several rows possible).
Auto_Confirmation_Timeout=	Indicates the time in seconds after which the list is hidden again.
Col_Lines=	Row width between the individual columns in pixels (no. 7).
Row_ColorBackground_A=	Background color of the even-numbered table rows (no. 5).
Row_ColorForeground_A=	Foreground color of the even-numbered table rows (no. 5a).
Row_ColorBackground_B=	Background color of the odd-numbered table rows (no. 4).
Row_ColorForeground_B=	Foreground color of the odd-numbered table rows (no. 4a).

Parameter	Meaning
Row_ColorBackgroundHeader=	Background color of the header (no. 3).
Row_ColorForegroundHeader=	Foreground color of the header (no. 3a).
Row_ColorCurrentSelection=	Color of the selection bar (no. 8).
Col_Width_nn=	Width of table column nn in pixels. If several table columns are to be displayed, the total width of all table columns must be adjusted to the display width.
Col_Visible_nn=	The parameter indicates whether the table column is to be visible or not (nos. 2a-2e). true= visible false = not visible
Col_Alignment_nn=	The parameter indicates how the contents of the table column are to be aligned. Possible values: left = left justified (no. 9) center = centered (no. 10) right = right justified.
Col_Type_nn=	The parameter indicates which content is displayed in the table column. Possible values: text = Text is displayed. Default value (e.g. no. 9) image = Graphics are displayed (no. 11) For text to be displayed, the parameter need not be defined (default setting).

3. Definition of the background and of the dialog text format in the [SurfaceDesign] section of the interface.ini file.

dormakaba Terminal 96 00-K7

For dialog with the operator, 2 info text lines and 1 line for error texts can be defined. If the contents of the info text line 1 are too long to be displayed, the info text line 2 containing the established parameter settings is displayed automatically.

The definition is carried out in the interface.ini file using several parameters. Further information on parameters (e. g. colors, alignment and the like) can also be found in chapter 12 "Notes".

The line for error texts is shown in the foreground above the info text line(s) for a brief period and then automatically hidden again.

When using this parameters in a dormakaba Terminal 98 00-K7, only the 1st info text lines will be used.

Parameterization takes place as follows.

File interface.ini, section [SurfaceDesign]

Parameter	Meaning
TableBackgroundColor=	The parameter defines the background color of the list display (no. 13).
TableInfoColor=	The parameter defines the color of the display box of the info text (no. 12).
TableInfoFontColor=	The parameter defines the color of the font of the info text (no. 1).
TableInfoFont=	The parameter defines the font of the info text (no. 1).
TableInfoAlignment=	The parameter defines the horizontal alignment of the display box.
TableInfoAlignmentV=	The parameter defines the vertical alignment of the display box.
TableErrorColor=	As above for error display box.
TableErrorFontColor=	As above for error text.
TableErrorFont=	As above for error text.
TableErrorAlignment=	As above for error display box.
TableErrorAlignmentV=	As above for error display box.

4. Definition of the column headings in the [Table_Template_nnn] section of the text.ini file.

The column headings are placed template-dependent in the /data/data/com.kaba.apps/hr/files/init/text.ini file. This means that for different templates it is also possible to define different column headings. The column headings can also be generated language-dependent by adding a language identification.

Parameterization takes place as follows.

File text.ini, section [Table_Template_nnn]

Parameter	Meaning
[Table_Template_nnn]	Name of the template section. The numeric extension (nnn) must be the same as the extension of the list the template is to belong to.
Col_Caption_nn_de= Col_Caption_nn_en= Col_Caption_nn_fr= ...	Heading for table column nn with language identification. The numeric extension (nn) makes reference to the table column the heading is to belong to (2a-2e).

5. References to graphics in the [Table_Images] section of the interface.ini file..

If graphics are also to be displayed in the list, they must be indicated in the [Table_Images] section of the interface.ini file.

Which graphic will be displayed, is controlled via the information in the list.

To display the graphics in a table column, the value for the parameter Col_Type_nn must be set to "image", e.g. Col_Type_05=image.

Parameterization takes place as follows.

File interface.ini, section [Table_Images]

Parameter	Meaning
[Table_Images]	Name of the graphics section.
Image<n>=<Dateiname.png>	Reference to graphics with filename (no.11). The numeric extension (n) must be same as the extension in the variable name in the list file. The name (Image) can be defined freely. The image name given here is just an example.

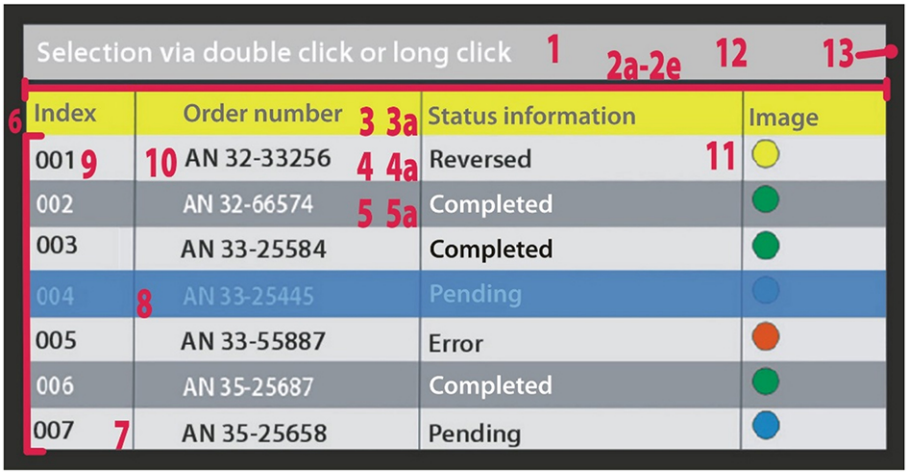
6. Placing graphics in the "images" folder in the data/data/com.kaba.apps/ba/files/images directory.

If graphics are also to be displayed in the list, they must be placed in the graphics folder in the format <filename>.png. The graphics displayed from this pool are those that are indicated by the definition in the list (see section 10). The size (visible dimension and file size) of the individual graphic file must have been adjusted to the display size or the disk space. For reasons of performance, the file size must be kept as small as possible.

7. Example configuration plus display example

The following list definition creates a list having the following properties.

The numbers (in red) refer to the explanations in the parameter descriptions.



Example definition for previous list.
a) File interface.ini, section [Table_Template_002]

```
[Table_Template_002]
Template_Name_de=Beispiel
Template_Name_en=Example
Input_Column=
Number_Of_Rows=7
Selection_Type=single
Auto_Confirmation_Timeout=
Col_Lines=1
Row_ColorBackground_A=gray
Row_ColorForeground_A=white
Row_ColorBackground_B=lightgray
Row_ColorForeground_B=black
Row_ColorBackgroundHeader=yellow
Row_ColorForegroundHeader=gray
Row_ColorCurrentSelection=blue
Col_Width_01=60
Col_Visible_01=true
Col_Alignment_01=left
Col_Width_02=156
Col_Visible_02=true
Col_Alignment_02=center
Col_Width_03=130
Col_Visible_03=false ###see note 1 below 1###
Col_Alignment_03=left
Col_Width_04=170
Col_Visible_04=true
Col_Alignment_04=left
Col_Width_05=96
Col_Visible_05=true
Col_Alignment_05=left
Col_Type_05=image ###see note 2 below###
```

Note 1: column 3 is not to be displayed.

Note 2: column 5 contains graphics.

Example definition for previous list. b) File interface.ini, section [SurfaceDesign]
[SurfaceDesign]
TableBackgroundColor=0 0 0 255
TableInfoColor=192 192 192 255
TableInfoFontColor=255 255 255 255
TableInfoFont=SansSerif PLAIN 22
TableInfoAlignment=left
TableInfoAlignmentV=center
TableErrorColor=255 0 0 255
TableErrorFontColor=255 255 255 255
TableErrorFont=SansSerif PLAIN 22
TableErrorAlignment=center
TableErrorAlignmentV=center

Example definition for previous list. c) File interface.ini, section [Table_Images]
[Table_Images]
Image1=bullet_rot.png
Image2=bullet_gruen.png
Image3=bullet_blau.png
Image4=bullet_gelb.png

Example definition for previous list. d) File text.ini, section [Table_Template_002]
[Table_Template_002]
Col_Caption_01_de=Lfd. Nr.
Col_Caption_01_en=Index
Col_Caption_02_de=Auftragsnummer
Col_Caption_02_en=Order number
Col_Caption_03_de=Schlüssel ###see note 1 below###
Col_Caption_03_en=Key ###see note 1 below ###
Col_Caption_04_de=Statusinformation
Col_Caption_04_en=Status information
Col_Caption_05_de=Grafik
Col_Caption_05_en=Image

Note 1: column 3 is not to be displayed.

Example definition for previous list. e) "images" directory in the path data/data/com.kaba.apps/ba/files/images.
bullet_blau.png
bullet_gelb.png
bullet_gruen.png
bullet_rot.png

8. Parameterization in the function key functions download data record

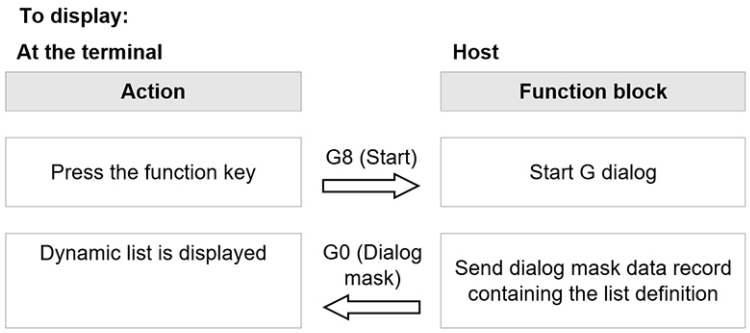
Parameter	Meaning
Block identifier	The block identification defines the function key that must start the G dialog.
Operating mode 6, 2. byte Activate dialog mode	The parameter must be set to 1, in order to activate the dialog mode for the indicated function key.

Info	Topic
3.10.34	Download record Function key functions

9. Checking whether the dialog has been enabled

Dialogs can be disabled and enabled via a download data record. To display a dynamic list via a function key, the "Dialog" function must have been enabled.

Info	Topic
5.2.1	Download record Dialog block/release



Example: G0 data record dialog window for displaying a dynamic list:

GID DID ON/OFF	Group address Device address Operating mode
G 0	Record type Record type modification
7	Dynamic list
0	No keyboard input
0	No displaying of badge content
0 1	Digit, from which on the selected list entry is to be inserted into the G1 record.
0 3	Number of digits of the selected list entry.
0 0	Position in the user information used for default value 00: no specification
0	Type of the input field, rolling
0	No acoustic signal
n .. n	User information (for the given example Selection via double click or long click
n .. n	HTTP address of the desired list. It is mandatory, that the http address starts at digit 54 of the data record. The address must be defined between the tags <LST> and </LST> **Example:** <LST>http://10.10.14.89/Lists/Example.002</LST> 160 digits max.

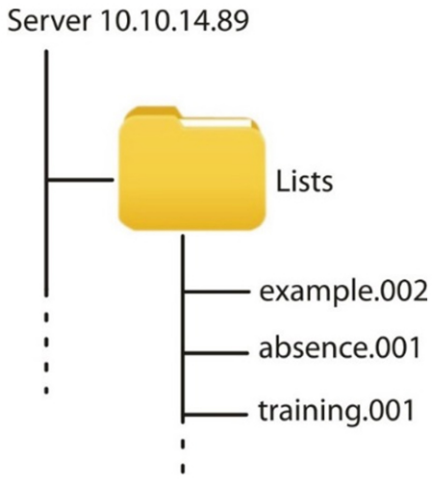
10. Filling the table with dynamic contents

The contents of the table are generated from a file the corresponding information is written to on the server side.

The filename and the path on the server the lists are to be retrieved from must be the list name and the path indicated in the HTTP call in the G0 data record.

Lists are defined by a numeric extension. This extension is used to determine the template to be used. Different lists can use the same template if they use the same extension, e.g. "absence.001", "training.001", a.s.o.

File naming for dynamic lists



Example.002

The example file example.002 contains the data for the table columns.

- Column 1 : Index number
- Column 2: Order number
- Column 3: Key code (this entry will not be displayed, the parameter Col_Visible_03=false).
- Column 4: Status information
- Column 5: Placeholder of the image

The individual contents of the file are separated by a semicolon, see example.

Example file example.002:

Column 1	Column 2	Column 3	Column 4	Column 5
001;	AN 32-33256;	AAA;	Reversed;	image4
002;	AN 32-66574;	BBB	Completed;	image2
003;	AN 33-25584;	CCC	Completed;	image2
004;	AN 33-25445;	DDD	Pending;	image3
005;	AN 33-55887;	EEE	Error;	image1
006;	AN 35-25687;	FFF	Completed;	image2
007;	AN 35-25658;	GGG	Pending;	image3

8 Display of HTML time sheets

8.1 Overview

The dormakaba devices of the NexT terminal series with B-Client HR40 offers the possibility to show time sheets of the person booking. This function requires a few configuration steps, which are summarized in this chapter.

To show a parameterization example, an HTML time sheet in conjunction with the EACM system under SAP has been selected.

The HTML time sheet is selected from the terminal via HTTP. For this purpose, a special key is configured on the terminal for each month to be shown, which generates a valid URL with the badge number and evaluation month after entering the badge.

After successful processing, SAP returns the HTML page in the response object. The display duration in seconds must be specified using the parameter <timeout=xxx>. With <timeout=0>, there will be no display. A reasonable maximum value for <timeout=> is 999, i.e. around 17 minutes. If the display is touched while the HTML time sheet is being displayed, the display time will be restarted.

EACM

Thanks to the interaction with the dormakaba software EACM, either the current month and (theoretically) any number of preceding months can be shown and selected if they are available in the superior system.

The selection of the month in conjunction with EACM is controlled via the special pidx parameter, see following chapter [▶ 8.2.2]. This parameter is used to inform the system by means of the generated URL about the month to be displayed.

Example for the current month

app.timesheet.MainActivity<url=http://host:port/kabaawm/htmlznw><pidx=0><timeout=30>

app.timesheet.TimesheetService<url=http://host:port/kabaawm/
htmlznw><pidx=0><timeout=30>

Example for the previous month

app.timesheet.MainActivity<url=http://host:port/kabaawm/htmlznw><pidx=1><timeout=30>

app.timesheet.TimesheetService<url=http://host:port/kabaawm/
htmlznw><pidx=1><timeout=30>

b-comm ERP 6 and B-COMM HCM

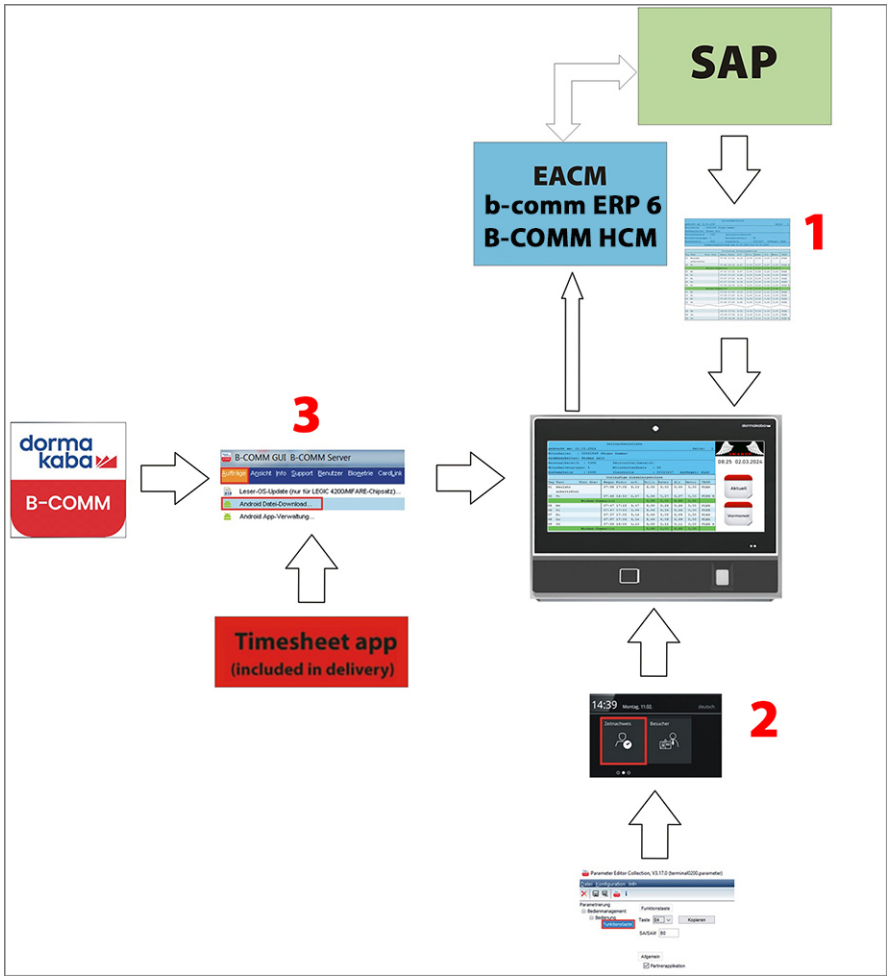
If dormakaba b-comm ERP 6 or B-COMM HCM software is used, only the current month can be shown. The pidx parameter is not used here.

Example for the current month

com.doka.app.timesheet.MainActivity<url=http://10.10.12.64:8080/timesheet/
zeitnachweis.jsp><timeout=30>

com.doka.app.timesheet.TimesheetService<url=http://10.10.12.64:8080/timesheet/
zeitnachweis.jsp><timeout=30>

The following diagram gives an initial overview of the interaction between the components concerned. This is only a simple functional overview without giving technical details.



The areas 1-3 of the diagram show the following tasks:

1 Request of the time sheet from the terminal to EACM, b-comm ERP 6 / B-COMM HCM, data transfer from SAP to EACM or b-comm ERP 6 / B-COMM HCM, configuration of the time sheet (appearance and contents) in SAP, data transmission to the terminal.

2 Configuration of the function key(s) and display timeout via PEC in B-COMM.

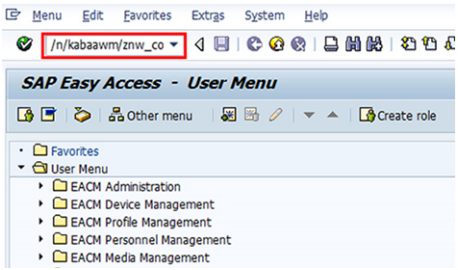
3 Download of the supplied Android app to the terminal via B-COMM. The Android app is not delivered together with the terminal software but with B-COMM or EACM.

8.2 Required configuration for the time sheet display

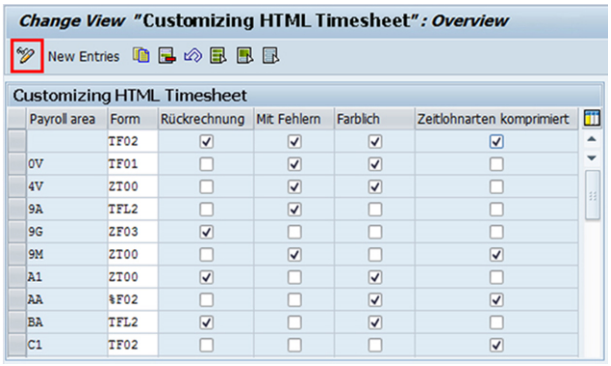
8.2.1 In the superior system. Example: EACM/SAP

To show a parameterization example, an HTML time sheet in conjunction with the EACM system under SAP has been selected. For parameterizing the time sheet in EACM/SAP, the /KABAAWM/ZNW_CONFIG transaction has been implemented.

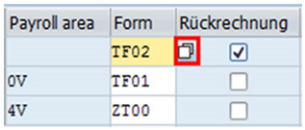
Open the maintenance screen in SAP via the /KABAAWM/ZNW_CONFIG transaction.



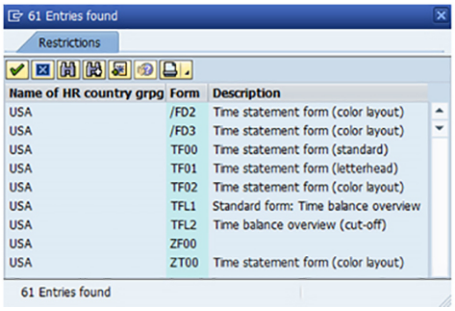
Select the Edit icon to configure the time sheet.



A list of the forms available in the system can be opened by clicking the Select icon.



You can now select a suitable form for the desired payroll area. Select the desired form from the list.



The check boxes in the maintenance screen can be used to define the characteristics of the time sheet.

Payroll area	Form	Rückrechnung	Mit Fehlern	Farblich	Zeitlohnarten komprimiert
	TF02	☑	☑	☑	☑
0V	TF01	☐	☑	☑	☐
4V	ZT00	☐	☑	☑	☐
9A	TFL2	☐	☑	☐	☐
9G	ZF03	☑	☐	☐	☐
9M	ZT00	☐	☑	☐	☑
A1	ZT00	☑	☐	☑	☐

The check boxes have the following meaning:

Check box	**Function**
Back calculation	Use the check box to define that back calculation periods are also displayed in the time evaluation.
With errors	Use the check box to define that a time sheet form is also displayed for employees for whom time evaluation has generated an error.
Color marking	Use the check box to define that the time sheet form is formatted in color according to SAP standard guidelines.
Time wage types, abbreviated	Use the check box to define that several time wage entries of the same time wage type are summarized in such a way that each time wage type appears only once per day.

Display example HTML time sheet

Zeitnachweisliste

gedruckt am: 22.01.2019 Seite: 1

Mitarbeiter : 00001049 Jürgen Sammer
Sachbearbeiter: Thomas Zeit

Personalbereich : 1300 Personalteilbereich:
Mitarbeitergruppe: 1 Mitarbeiterkreis : DU
Kostenstelle : 4400 Planstelle : 50010247 AZPRegel: FLEX

Auswertungszeitraum vom 01.01.2015 bis 31.01.2015

Vorläufige Einzelergebnisse

Tag	Text	Kter	Gter	Beguz	Enduz	erf.	Sollz	Rahmz	Glz	Mehrz	TAZP
01	Neujahr arbeitsfrei			07:58	17:05	9,13	8,00	8,00	0,00	0,00	FLEX
02	Fr			07:48	14:16	6,47	5,00	5,27	0,27	0,00	FLEX B
	Wochen-Summation						0,00	0,00	0,00	0,00	
05	Mo			07:47	17:15	9,47	8,00	8,26	0,26	0,00	FLEX
06	Di			07:47	17:15	9,48	8,00	8,26	0,26	0,00	FLEX
07	Mi			07:57	17:05	9,14	8,00	8,09	0,09	0,00	FLEX
08	Do			07:57	17:05	9,14	8,00	8,09	0,09	0,00	FLEX
09	Fr			07:59	14:06	6,13	5,00	5,11	0,11	0,00	FLEX B
	Wochen-Summation						0,00	0,00	0,00	0,00	
12	Mo			07:58	17:06	9,14	8,00	8,10	0,10	0,00	FLEX
13	Di			07:59	17:06	9,13	8,00	8,11	0,11	0,00	FLEX
14	Mi			07:49	17:16	9,45	8,00	8,28	0,28	0,00	FLEX
15	Do			07:46	17:15	9,48	8,00	8,25	0,25	0,00	FLEX
28	Mi			08:00	17:01	9,03	8,00	8,03	0,03	0,00	FLEX
29	Do			07:59	17:06	9,13	8,00	8,11	0,11	0,00	FLEX
30	Fr			07:59	14:06	6,13	5,00	5,11	0,11	0,00	FLEX B

8.2.2 Parameterization of one or several function keys

The time sheet can be selected on the terminal via function keys which have been parameterized exactly for this purpose.

For each month to be loaded, a specific function key has to be parameterized.

If b-comm ERP is used, only the current month can be shown.

Example

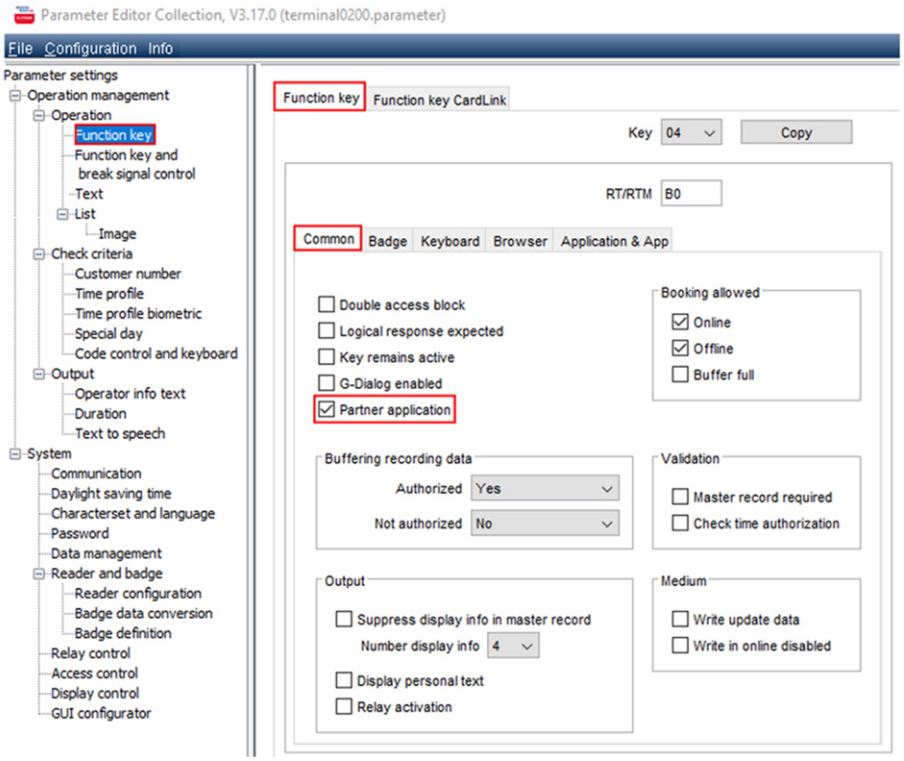
To be able to show the current month and the previous month, 2 parameterized function keys are necessary.

The parameterization of the function keys can be done in the PEC. The PEC (Parameter Editor Collection) is selected under B-COMM for the terminal 97 00 and allows a simple parameterization.

Step 1:

In order to be able to use the HTML time sheet, the check box “Partner application” for the corresponding app must be ticked (see the following chapter).

To do this, open the operation management in PEC, open the “Function key” tab in the “Operation – Function key” menu, and tick the “Partner application” check box in the “General” submenu.



Step 2:

For each month to be selected, a specific function key has to be parameterized. To do this, open the operation management in the PEC, open the “Function key” tab in the menu Operation - Function key and define the following parameters in the submenu “Application & App”:

1 Selection of a free function key

2 Record type and record type definition for this function key

3 Specification of the class app.timesheet.MainActivity<url=http://host:port/kabaawm/htmlznw><pidx=0>

4 Specification of the “<timeout=xxx>” parameter. with xxx representing the number of seconds.

The values for the URL must be adapted to the system (IP address, port).

The following values apply to the <pidx=> parameter:

0 = Display of the current month

1 = Display of the previous month

2 = Display of the month before last

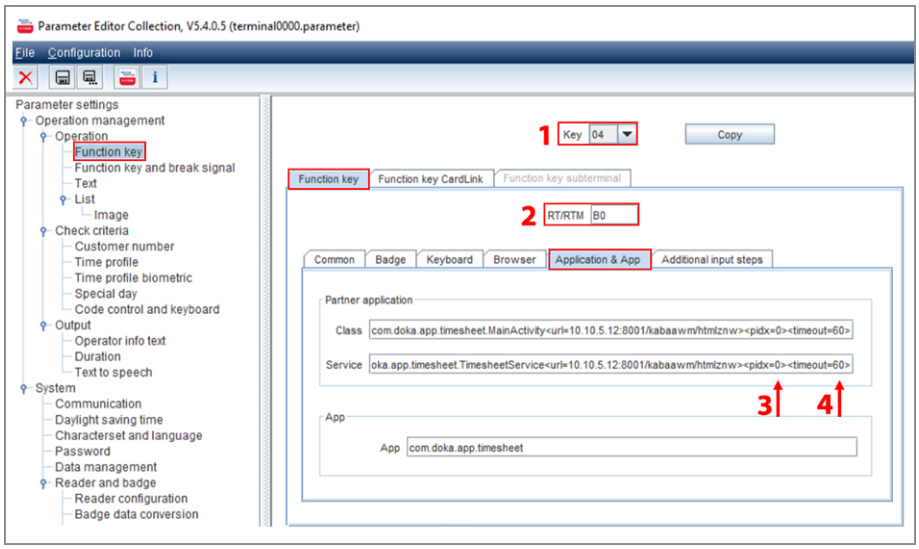
...= earlier months

The example shows the current month (< pidx=0>).

3. Specification of the service

oka.app.timesheet.TimesheetService<url=http://host:port/kabaawm/htmlznw><pidx=0><timeout=60>

The information given in point 3 applies to URL and <pidx>. Save the data.



Parameterization of function key 04 for displaying the current HTML time statement is thus complete.

To display the time sheet of the previous month, for example, a function key 05 can be parameterized with <pidx=1>.

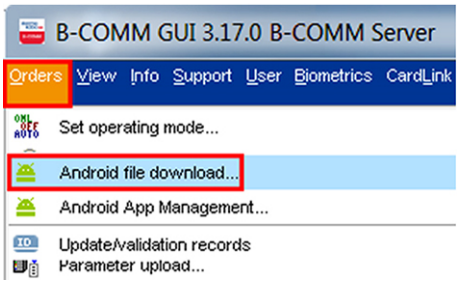
The appearance of the function key and the respective user texts can be adapted individually via the PEC.

8.2.3 Download and installation of the app

The function HTML time sheet on a terminal 97 00 requires an Android application. This application is included in the delivery, e.g. for EACM.

The application must be loaded onto the desired terminal and installed. For this, a special order in B-COMM is used.

Click the menu "Orders" in B-COMM and select the menu item "Android file download"



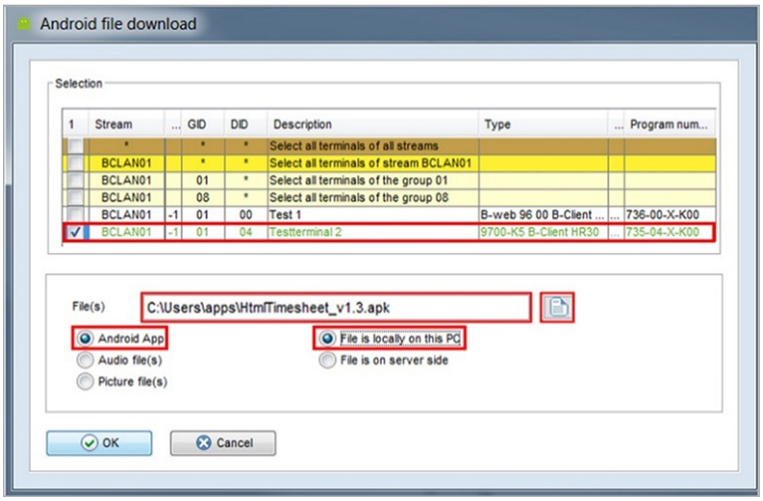
Enter the required parameters and specifications into the order menu.

1 Selection of one (or several) terminals to which the application is to be transmitted.

2 Selection of the file form "Android App".

3 Specification of the storage location of the app; use the directory browser for navigation, if necessary.

4 Confirm the specifications with "OK".



An order to download the Android app is written into the order file and processed afterwards according to the order parameters. Details regarding the orders, see reference manual B-COMM.

After the download, the Android app is automatically installed on the terminal.

8.3 Selection on the terminal

After successful parameterization and installation, the time sheet can be selected on the terminal. This is done by pressing the parameterized function key and making a booking.

The terminal generates a URL with the badge number used for booking and receives, as a response from the system, a time sheet for display.

In the example, 2 function keys have been parameterized: one for the current and one for the previous month.

Display of the time sheets ends automatically 1 minute after the last operation. The display can also be ended manually by pressing the “Back” key.

Display example on the terminal

Example for a time sheet

Zeitnachweisliste

gedruckt am: 22.01.2024 Seite: 1

Mitarbeiter : 00001049 Jürgen Sammer

Personalbereich : 1300 Personalteilbereich:
Mitarbeitergruppe: 1 Mitarbeiterkreis : DU
Kostenstelle : 4400 Planstelle : 50010247 AZPRegel: FLEX

Auswertungszeitraum

Vorläufige Einzelergebnisse

Tag	Text	Kter	Gter	Beguz	Enduz	erf.	Sollz	Rahmz	Glz	Mehrz	TAZP
01	Neujahr arbeitsfrei			07:58	17:05	9,13	8,00	8,00	0,00	0,00	FLEX
02	Fr			07:48	14:16	6,47	5,00	5,27	0,27	0,00	FLEX B
	Wochen-Summation						0,00	0,00	0,00	0,00	
05	Mo			07:47	17:15	9,47	8,00	8,26	0,26	0,00	FLEX
06	Di			07:47	17:15	9,48	8,00	8,26	0,26	0,00	FLEX
07	Mi			07:57	17:05	9,14	8,00	8,09	0,09	0,00	FLEX
08	Do			07:57	17:05	9,14	8,00	8,09	0,09	0,00	FLEX
09	Fr			07:59	14:06	6,13	5,00	5,11	0,11	0,00	FLEX B
	Wochen-Summation						0,00	0,00	0,00	0,00	
12	Mo			07:58	17:06	9,14	8,00	8,10	0,10	0,00	FLEX
13	Di			07:59	17:06	9,13	8,00	8,11	0,11	0,00	FLEX
14	Mi			07:49	17:16	9,45	8,00	8,28	0,28	0,00	FLEX
15	Do			07:46	17:15	9,48	8,00	8,25	0,25	0,00	FLEX
30	Fr			07:59	14:06	6,13	5,00	5,11	0,11	0,00	FLEX B

9 Bookings via cell phone (dormakaba mobile access)

9.1 Overview

Dormakaba mobile access is a simple and efficient solution which enables people to use a cell phone for fast and reliable access or time and attendance bookings.

The cell phone is here treated as a normal identification medium. Additional hardware and software options are required for access bookings.

The booking authorizations are transmitted via the air interface and the trusted service manager of a mobile network operator (MNO).

Rights can be changed at short notice, flexibly and fast.

Dormakaba mobile access transmits data via secure processes. All processes are protected by LEGIC Connect.

Several configuration steps in various environments are required to use mobile access in an access or time and attendance system.

The parameters which are required or possible within the B-Client HR40 device software used are described in the present manual.

All other required settings and activities are briefly described for a better understanding. The information required for this purpose can be found in the corresponding documentation.

In order to use mobile access in a terminal of the NexT series, this requires the relating function variant "mobile access".

Mobile Access in combination with CardLink

The booking variants CardLink and Mobile Access can also be can be used together under the same function key. In this case, the device software automatically detects during the booking process, whether a CardLink booking was made with an RFID medium or whether a smartphone was used for the booking (T/A or AC booking).

The settings required in the .ini files

- mobiledef.ini and mobileact.ini for mobile access and
- mediadef.ini and mediaact.ini for CardLink

are still necessary.

9.1.1 Features

- Central management of booking authorizations.
- Transfer of booking authorizations via the mobile network in a secure process.
- Combination with dormakaba online or standalone components.
- Booking authorizations can be issued:

  - on a cross-location basis, (for company employees from different locations)

  - outside office hours (e.g. for maintenance personnel)

  - at short notice, e.g. for substitutes in case of illness

  - for a limited period of time

  - even for isolated geographical locations (external warehouses, mining).
- No local media transfer is required.
- No local programming of components is necessary.
- No expenses due to forgotten or lost badges.

9.1.2 Further related documents

- Kaba exos mobile access planning guideline "PG Mobile Access" Nr. 04046727
- B-COMM Reference Manual No. 04037204

9.2 Glossary

The following glossary explains the most important technical terms and abbreviations in connection with dormakaba mobile access.

LEGIC Connect

LEGIC Connect is an online interface which was developed in cooperation with LEGIC to enable the use of Bluetooth Low Energy and NFC-enabled mobile phones as user media. The interface allows short-term adjustments of these media via the mobile network, so that even media currently in circulation can be updated on a timely basis.

MNO

Mobile **N**etwork **O**perator. Operator of a mobile network.

VCP Versatile Configuration Package

Configuration package containing information on the system (LEGIC Connect project ID) for distributing/sending digital keys etc. This configuration package requires the component so that it functions with mobile access.

Trusted Service Manager

A trusted service manager is a role in a near field communication ecosystem. It acts as a neutral broker that sets up business agreements and technical connections with mobile network operators, phone manufacturers or other entities controlling the secure element on mobile phones. The trusted service manager enables service providers to distribute and manage their contactless applications remotely by allowing access to the secure element in NFC-enabled handsets.

NFC Near Field Communication.

Wireless data exchange by means of RFID within a close range of up to 15 cm via an interface as per standard ISO/IEC 14443 or ISO/IEC 15693.

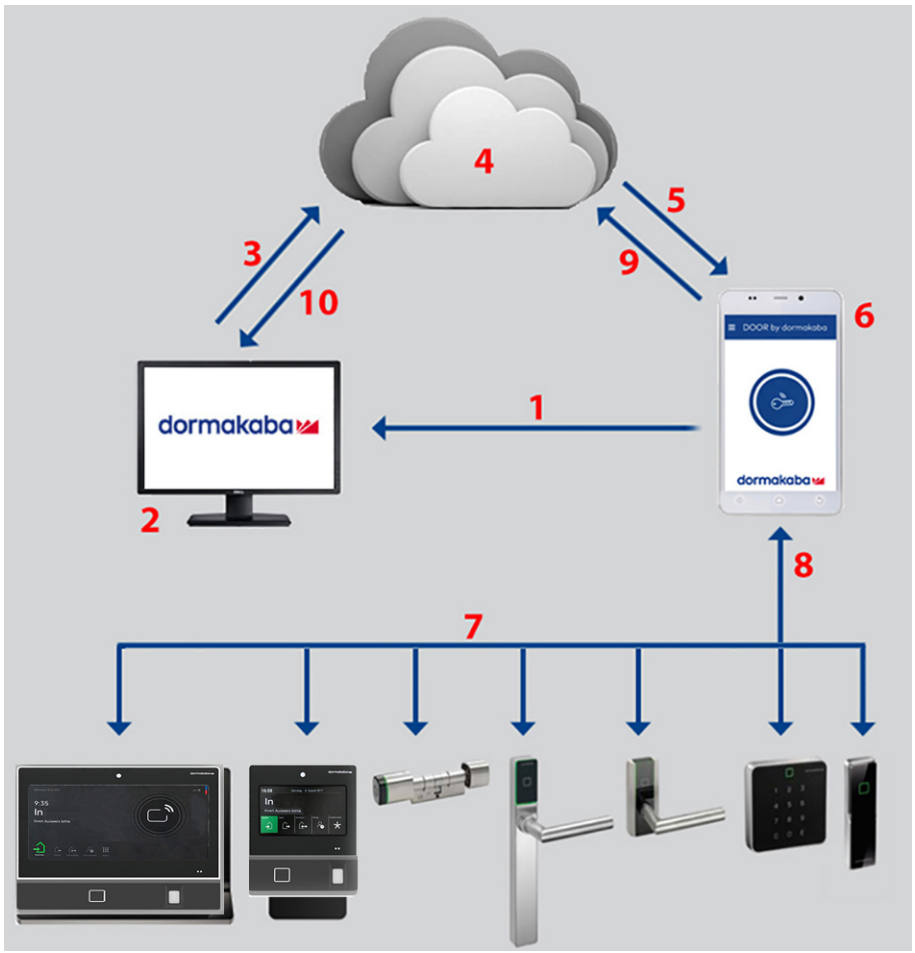
Bluetooth Low Energy

Bluetooth Low Energy is an energy-saving wireless technology which can be used to network devices in an environment of up to 10 meters.

API Key

The **A**pplication **P**rogramming **I**nterface key protects against unauthorized access via the LEGIC Connect interface.

9.3 Functional overview



1 Request access rights via app or phone call.

2 Access management software.

3 Assign rights.

4 LEGIC Connect enables secure data transfer.

5 Transfer access rights.

6 A smart phone serves as an access medium via NFC (Near Field Communication) or Bluetooth Low Energy.

7 Access via dormakaba NexT terminal or door components.

8 The door status and events are read back, if supported.

9 The door status and events are transmitted, if supported.

10 The door status and events are fed back into the system, if supported.

9.4 Requirements

For LEGIC Connect

The entire configuration is completed in the LEGIC Connect Control Center and the API key is present.

For the terminals

Definition of the .ini file parameters. They are explained in respective chapters, see below.

For the customer

- An Internet connection exists.
- A Bluetooth Low Energy or NFC enabled cell phone with an Android or iOS operating system is available.
- A SIM card is present in the cell phone.
- The 'DOOR' and 'VCP Installer' apps are installed on the cell phone.

9.5 Settings in the .ini-files

Some settings have to be made in the ini-files mobileact.ini and mobiledef.ini in the directory ../com.kaba.apps.ba.

A "Double access block" function for smartphone bookings is realized via the parameter "SerialGuardTime" in the "system.ini" file in the [Reader1] and [Reader2] sections.

Info	Topic
14.1.9	mobileact.ini
14.1.10	mobiledef.ini
14.1.4	system.ini

9.6 Data trecords for dormakaba mobile access

The terminal sends a data record concerning the result of VCP processing in the mobile access context.

Info	Topic
7.3.6	Report result of VCP processing

9.7 Configuration process

The following table illustrates an exemplary configuration process. Some of the steps or the persons responsible for them may vary in real installations.

	Configuration step	Responsible/Remark
1	First plan and administration Determine the number of doors and components, door spacing, evaluate sources of interference, cell phones used, purchase order release and processing.	Customer in cooperation with RMO.
2	Create a LEGIC Connect project in the LEGIC Cloud for this end customer.	LEGIC IT persons, COM-ID team
3	Configure the interface for LEGIC Connect in the host system. LEGIC Connect requires the following information for this purpose: - LEGIC Connect project ID - LEGIC Connect API key - LEGIC Connect URL - LEGIC Connect MobileApp ID - URL of the credential calculator	Host system administrator
4	Prepare the user media, create master records and integrate the cell phones via their phone numbers, master record download to terminals.	Customer Host system administrator
5	Download of VCP Installer App. Available in Google Play Store and Apple store. dorma kaba VCP	Customer's IT personnel The app is necessary to distribute project-specific key packages to actuators/terminals. This app is required for the initial installation of Bluetooth Low Energy and/or NFC enabled door components/terminals.
6	Download of dormakaba mobile access app and registration of LEGIC Connect. Available in Google Play Store and Apple store. dorma kaba	Customer, user Standard app for accessing Android or iOS cell phones that support Bluetooth Low Energy and NFC.

10 CardLink function using LEGIC and Mifare readers

10.1 General

CL

This chapter describes the use of the CardLink functionality as it is implemented in several derivates of the B-Client terminal software.

CardLink is an access control concept in which online components (terminals) are effectively connected to standalone components (door components). The main benefit of the CardLink concept is the fact that access authorizations no longer have to be available in the individual terminals, but are transmitted by a so-called update terminal to the media of the employees.

This allows integration of door components into an overall concept without the need for data lines and without requiring complicated connections and programming.

The CardLink function can work with different reader technologies. The reader technology is set in accordance with the reader used.

Depending on the technology used, different entries must be made in the parameter records. These differences have contrasting colors in the explanations of the data records.

The colors have the following meaning:

Parameters and settings for reader type LEGIC

Parameters and settings for reader type MIFARE

CardLink in combination with Mobile Access

The booking variants CardLink and Mobile Access can also be can be used together under the same function key. In this case, the device software automatically detects during the booking process, whether a CardLink booking was made with an RFID medium or whether a smartphone was used for the booking (T/A or AC booking).

The settings required in the .ini files

- mobiledef.ini and mobileact.ini for mobile access and
- mediadef.ini and mediaact.ini for CardLink

are still necessary.

10.2 Migration mode

For migrations, the new parameter CLAddressType has been incorporated in the mediadef.ini file.

For further explanations, please see the following reference.

Info	Topic
14.1.7	File mediadef.ini

10.3 Adressing in data records

10.3.1 Segment name in the data records <3fxx, V0, G2

The segment name specifies the segment of the medium in which the CardLink data or the application data are to be evaluated. The segment name can be specified in the following way:

Direct CardLink segment entry for reader type LEGIC

The segment name is specified in hex form. It must always comprise 14 digits. If the segment name of the LEGIC medium to which data are to be written is shorter than 14 digits, the missing digits must be filled with the wildcard character "hyphen" (–).

Example of direct segment entry:

2	3	F	7	D	C	5	6	D	—	—	—	—	—

Reference to mediadef.ini / mediaact.ini

The specification of the segment designator takes place by use of the keywords CARDLINK or APPLICATIONx in capital letters followed by placeholder characters "hyphen" (-). The identifier "x" at "APPLICATIONx" is a placeholder for a numbering, see example 2.

This notation causes the terminal to search for a

- CardLink findstring (parameter name: CLFindstring=) or an
- application findstring (parameter name: APPxFindstring=)

in the corresponding sections [LegicStructureX] of the mediadef.ini file and use it.

Reference mediadef.ini/mediaact.ini: Example 1:

C	A	R	D	L	I	N	K	—	—	—	—	—	—

Reference mediadef.ini/mediaact.ini: Example 2:

A	P	P	L	I	C	A	T	I	O	N	1	—	—

Info	Topic
14.1.7	File mediadef.ini
14.1.6	File mediaact.ini

10.3.2 Memory address in the data records <3fxx, V0, G2

This entry specifies the address of the CardLink data in the CardLink segment. The address can be specified in the following way:

Direct CardLink segment entry for reader type LEGIC

The address is specified in hex form. It must always comprise 4 digits. If the address is shorter than 4 digits, the missing digits must be filled with leading zeros.

Example of direct address entry:

0	0	0	0

Reference to mediadef.ini / mediaact.ini

The address is specified using 4 wildcard characters "hyphen" (–).

This notation causes the terminal to search for the associated CardLink address in the sections [LegicStructureX] of the mediadef.ini file (parameter name in the file: CLAddress=) and use it.

Example of reference to mediadef.ini/mediaact.ini:

–	–	–	–

Info	Topic
14.1.7	File mediadef.ini
14.1.6	File mediaact.ini

10.3.3 Application designation

Reader type MIFARE

The application designation specifies the file on the **Mifare** medium the CardLink data are written to.

For writing CardLink or application data, one of the following application designations must be entered in the parameter or data record:

- CARDLINK
- CARDLINK1....CARDLINK16
- MEDIATRACEBACK
- APPLICATION
- APPLICATION1....APPLICATION99

CARDLINK and CARDLINK1 are synonyms, i.e. they refer to the same file.

The application designation **may only be entered as a whole with capital letters and without spaces**. It must always comprise 14 digits. If the application designation is shorter than 14 digits, the missing digits must be filled with the wildcard character "hyphen" (–).

Information for CardLink online operation in G dialog

If more than one G2 dialog record is used, the details for the application designation are not required in the following dialog records.

Application designation: Example 1:

Designation for a CardLink section in the "system.ini" file.

C	A	R	D	L	I	N	K	1	6	—	—	—	—

Application designation: Example 2:

For the "Media Traceback" application containing G4/G6 dialog records, only the entry MEDIATRACEBACK is admissible.

M	E	D	I	A	T	R	A	C	E	B	A	C	K

Application designation: Example 3:

Designation for an application in the "system.ini" file.

A	P	P	L	I	C	A	T	I	O	N	2	5	—

Info	Topic
10.8.1	G4 record: Request data from media
10.7.1	G2 record: Dialog record Write data
10.8.4	G6 record: Delete MediaTraceback file contents

10.3.4 Offset

Reader type MIFARE

The offset specifies from which position of the application the data are entered.

The value must always be entered in 4 digits.

The default setting for the offset is 0000, i.e. from the beginning. This indication may be omitted; in this case, the default value is applied.

If more than one G2 dialog record is used, the details for the offset are not required in the following dialog records. However, the preset number of places must be present in the dialog record in any case.

10.4 CardLink validation via function key functions

This download record is used to define the CardLink function "Validation" for a certain function key.

Along with the validation, a time stamp will be written to the medium. For the data records of record type/record modification v0 described below, previous activation of a function key is required with this download record.

GID DID ON/OFF	Group address Device address Operation mode
> 3	Record type Record type modification
f 0 0	Block identification and number of the function key, see the following table.
A ?	Record type (A) and record type modification (?) of the CardLink data record, see chapter
n n	Administration area, see following description
n ... n	14-digit segment name (direct entry or reference to mediadef.ini) or 14 digits application designation
n n n n	4-digit memory address (direct entry or reference to mediadef.ini) or 4 digits offset
0	Mode byte, see following description.
0	Reserved 0

Block identifier

The block identification defines the function key for which this download record is valid.

Possible values: f00 to f05 and f31 to f40 according to the function keys F00 to F05 and F31 to F40.

Record type/Record type modification

The following table shows the standard settings for the RT/RTM for the respective function key.

Function key	RT/RTM
F00	A?
F01	B?
F02	B?
F03	B?
F04	B?
F05	F?
F31	B1
F32	B2
F33	B4
F34	B5
F35	BE
F36	FA
F37	FB
F38	FC
F39	FD
F40	FE

Administration area

The administration area is a classification criterion in the CardLink concept. The administration area stored with this download record is compared to the administration area read by the medium. A maximum of 256 administration areas is possible.

Possible hexadecimal values: 00 to FF

Standard factory setting

— —

The administration area is not checked with this setting.

Segment name/application designation

See explanations in chapter [▶ 10.3.1] and chapter [▶ 10.3.3].

Standard factory setting

— — — — — — — — — — — — — —

No validation is performed for the function key with this setting.

When using non-segmented LEGIC media, the digits of the segment name must be filled with zeros (no hyphens). However, the characters are not checked.

Memory address/Offset

See explanations in chapter [▶ 10.3.2] and chapter [▶ 10.3.4].

Standard factory setting

— — — —

No validation is performed for the function key with this setting.

Mode byte

Bits 1 and 2 affect the

- deletion and
- rewriting

of the time stamp to the medium.

Generally, a time stamp is changed only if the validation of the administration area corresponds to the value given in the CardLink function key functions parameter.

If bit 1 was not set (default), the time stamp will be reset (invalidation).

The revalidation, i.e., the application of a new time stamp, is controlled by means of bit 2. If bit 2 was not set (default), the validation will be carried out as defined in the validation record v0.

If there is no validation record, the time stamp will be treated as if the CardLink bit 1 was set to the value 0 (do not carry out validation).

Byte 1

8	4	2	1	Hex value

3	2	1	0	Bit
				Report the time stamp. 0 = Do not report time stamp (default) 1 = Time stamp will be reported
				Delete the time stamp 0 = Time stamp will be deleted (default) 1 = Time stamp will not be deleted
				Carry out the validation according to the validation record v0 0 = Validation carried out according to the validation record v0 (default) 1 = Validation will not be carried out
				Reserved 0

The links between the mode byte settings and the v0 record are shown in the following table.

Mode byte 1		CardLink byte from validation record v0	Time stamp on medium after booking
Bit 2	Bit 1	Bit 1	
0	0	0	deleted
		1	new
0	1	0	unchanged
		1	new
1	0	not relevant	deleted
1	1	not relevant	unchanged

Info	Topic
10.5.6	CardLink data record
10.5.1	CardLink validation record v0

10.5 Data records for CardLink validation

The validation is a CardLink-specific function during which a CardLink validation stamp is written to a LEGIC medium. In doing so, please observe the following:

Writing CardLink validation stamp	
•	If the CardLink validation is activated, first, it is processed during booking, i. e. before the time and attendance recording. The CardLink validation is independent of the authorizations during time and attendance booking.
•	The current date and current time rounded up to the next full hour are written onto the data carrier.
•	If the CardLink validation under a function key is active, and validation for the person booking was activated via a suitable v0 record, the validation stamp will be written.
•	If the CardLink validation was activated by means of a function key, but - there is no v0 validation record available for the medium number of the person or - the validation for this badge number is off the validation will be reset (devalidated).

Info	Topic
10.5.1	CardLink validation record v0

10.5.1 CardLink validation record download

The validation record contains all the employee-related CardLink information.

The number of validation records that can be loaded to the terminal depends on the licensed memory configuration. If the permitted number is exceeded in a validation record download, the terminal generates an O2 error record (command cannot be executed at the moment).

GID DID ON/OFF	Group address Device address Operation mode
v 0	Record type Record type modification
n ... n	ID no.; number of digits 5-20, depending on definition.
0 2	Validation record
n	CardLink byte 1, see following description
n	CardLink byte 2, see following description Evaluated 1 or 0 for reader type LEGIC or Evaluated 1 or 0 for reader type MIFARE

CardLink byte 1

The CardLink byte 1 contains CardLink-specific parameters for update and validation.

8	4	2	1	Hex value
3	2	1	0	Bit

Update required
0 = No
1 = Yes
Run validation
0 = No
1 = Yes
Reserved 0
Reserved 0

Update required

1: The symbol "Update required" is displayed.

0: The symbol "Update required" is not displayed.

Symbol "Update required"

The symbol, simultaneously with the booking response, is always displayed blinking. Its size and position can be changed in the file interface.ini in section [SurfaceDesign] by means of the parameter LegicWriteImageBounds.

Running validation

1: Validation is run for the corresponding medium.

0: Validation is not run.

Info	Topic
14.3.4	File interface.ini

CardLink byte 2

Reader type MIFARE or

Reader type LEGIC

CardLink byte 2 contains specific parameters for the media traceback function. If bit 0 is set in CardLink byte 2, it is signaled to the host application that the media traceback data is to be read from this medium by means of a G dialog.

8	4	2	1	Hex value
3	2	1	0	Bit
				Read out media traceback data 0 = No 1 = Yes
			Reserved 0	
		Reserved 0		
	Reserved 0			
Reserved 0				

10.5.2 Deleting CardLink validation record

A validation record is deleted from the terminal by means of the following data record. To this end, the medium ID whose associated validation record is to be deleted is entered.

If all validation records in the terminal are to be deleted, wildcards (characters *) can be entered instead of the medium ID.

The number of wildcards must correspond to the definition of the parameter IDLength in the [MasterRecordDefinition] section of the file application.ini.

NOTICE

Note that the wildcards must be used with caution, because all update records are lost after the delete download.

GID DID ON/OFF	Group address Device address Operation mode
v 1	Record type Record type modification
n .. n	ID no.; number of digits 5-20, depending on definition.

10.5.3 Request CardLink validation record

A validation record is requested by means of the following data record. To this end, the medium ID whose associated validation record is to be requested is entered.

If all validation records are to be requested, the corresponding number of wildcards (characters *) can be entered instead of the medium ID. The number of wildcards must correspond to the definition of the parameter IDLength in the [MasterRecordDefinition] section of the file application.ini.

After the successful upload of one or several validation records, the terminal will send a V9 record as identification of the end.

GID DID ON/OFF	Group address Device address Operation mode
v 8	Record type Record type modification
n .. n	ID no.; number of digits 5-20, depending on definition.

Info	**Topic**
10.5.5	End identification

10.5.4 Report CardLink validation record

The terminal transmits the following data record after requesting via a v8 record containing the ID number.

GID DID ON/OFF	Group address Device address Operation mode
v 0	Record type Record type modification
n ... n	ID no.; number of digits 5-20, depending on definition.
n .. n	Content of the validation record.

Info	Topic
10.5.1	Validation record
10.5.3	Request validation record

10.5.5 End of CardLink validation record upload/download

The following data record reports the end of an up- or download procedure. It must be transmitted to the terminal after a validation record download as identification of the end after the data record. The terminal sends this record automatically to the computer after the upload of the validation record has been completed.

GID DID ON/OFF	Group address Device address Operation mode
v 9	Record type Record type modification

10.5.6 CardLink data record validation

Reader type LEGIC with CardLink version 1.0

The validation process is logged by means of a data record. It contains the following information:

- Media data according to parameter setting
- Result of the writing operation
- Administration area
- Log data (battery and memory
- LEGIC UID (is always read and transferred)

The CardLink data record is transmitted prior to the standard time and attendance record.

The keyboard entry data are not entered in the data record.

GID DID ON/OFF	Group address Device address Operation mode
n n	Record type and record type modification see following description
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day.
h h m m (s) (s)	Time Seconds only if set as parameter.
n ... n	6 digits identification of the user language. Only if set as parameter.
0	Record format for LEGIC with CardLink version 1.0
n ... n	Media contents according to parameter setting; variable length.

n	Information byte, see following description.
n n	Status byte
n n	Administration area, defined for function key.
n n n n n n	6-digit log data, see the following description. 4-digit door no. 1 digit Battery state 1 digit Event memory
n ... n	LEGIC UID; variable length (max. 10 bytes = 20 characters).

Record type/Record type modification

The record type and record type modification of the registration record are specified by means of the function key parameter setting in the parameter record fxx.

Info	Topic
13.4	Table of clock identifiers
10.7.2	Status byte

Information byte

The information byte with its entry contains information on whether the data transmission to the medium has been successful.

Entry	Meaning
0	Validation data have been written successfully
4	No writing possible with connected reader
5	Validation data could not be written
7	Validation data deleted or invalidation performed, but not written again. : Reason: - no validation record available or - "Do not perform validation" programmed in validation record or - "Deactivate validation" was programmed.
8	Validation data could not be deleted
9	Administration area does not match
A	Type is not identical
B	CardLink segment could not be read

Administration area, defined for function key.

The administration area is defined by the function key parameter setting in the parameter record fxx.

Info	Topic
10.4	Function keys configuration for CardLink

Log data

The log data are data written from the door components onto the data carrier. The information on door number, battery state and status of the event memory are entered in the data record as follows:

Entry	Meaning	Possible values
Door number	Number of the door component from which the information is received.	0000 – 9999
Battery state	Charging state of the door component battery.	T =Battery (nearly) flat F = Battery fully charged
Event memory	Filling level of the door component event memory.	T = Event memory full F = Defined filling level of the event memory not reached yet.

10.5.7 CardLink data record validation

Reader type LEGIC with CardLink version 1.1

Reader type MIFARE

The validation process is logged by means of a data record. It contains the following information:

- Media contents according to parameters
- Result of the validation process
- Administration area
- CardLink byte 2 from the validation record
- UID (is always read out and transferred)
- Data of the actuator status file
- Time stamp

The descriptions of the CardLink validation record for reader type LEGIC apply.

The parameters

- Record format
- Contents of the actuator status file
- Previous validation stamp
- New validation stamp

are described in the following.

GID DID ON/OFF	Group address Device address Operation mode
n n	Record type and record type modification see following description
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day.
h h m m (s) (s)	Time Seconds only if set as parameter.
n ... n	6 digits identification of the user language Only if set as parameter.
1	Record format for LEGIC and MIFARE with CardLink version 1.1
n ... n	Media contents according to set parameters. Variable length.
n	Information byte, see following description.
n n	Status byte
n n	Administration area, defined for function key.
n	CardLink byte 2 from validation record Evaluated 1 or 0 for reader type LEGIC or Evaluated 1 or 0 for reader type MIFARE
n ... n	20 digits UID

n ... n	64 digits for contents of the actuator status file, see following description.
n ... n	6-digit previous validation stamp, see following description.
n ... n	6-digit new validation stamp, see following description.

Record type/Record type modification

The record type and the record type modification are defined by the function key parameter setting in the parameter record fxx.

Information byte

The information byte with its entry contains information on whether the data transmission to the medium has been successful.

Entry	Meaning
0	Validation data have been written successfully
4	No writing possible with connected reader
5	Validation data could not be written
7	Validation data deleted or invalidation performed, but not written again. Reason: - no validation record available or - "Do not perform validation" programmed in validation record or - "Deactivate validation" was programmed.
8	Validation data could not be deleted
9	Administration area does not match
A	Type is not identical
B	CardLink segment could not be read
C	CardLink application name entry invalid

Contents of the actuator status file

The data from the actuator status file are written to this position of the CardLink validation record. 4 blocks of 8 bytes each, i.e., 32 bytes, of user data are written, which corresponds to 64 digits.

Previous validation stamp, new validation stamp

The validation stamps contain the time at which a validation was performed in 3 bytes each in the CardLink-specific format.

- Previous validation stamp: Validation stamp read from the medium.
- New validation stamp: Current time stamp.

A validation stamp is only written to the "Validation" registration record if the "Report time stamp" bit (bit 0) in the download record of the CardLink function key functions was set.

Info	Topic
10.9.1	Actuator status
13.4	Table of clock identifiers
10.7.2	Status byte
10.4	Function key functions

10.5.8 CardLink error message for validation/update process

If an error occurs during the writing operation of the update or validation data, it can be displayed for the user at the terminal by means of an error symbol along with the data record.

This symbol is displayed if the parameter "DisplayLegicWriteErrorImage" is set to "true" in the file interface.ini in section [SurfaceDesign]. In the error state, the symbol is displayed blinking with alternating acoustic signals together with the booking response.

This parameter is **false** by default, so that the error symbol is not displayed.

Error symbol using the parameter "DisplayLegicWriteErrorImage=true"

The position and size of the symbol on the display is the same as for the symbol "Update required". The symbol is always displayed blinking. Its size and position can be changed in the file interface.ini in section [SurfaceDesign] by means of the parameter LegicWriteImageBounds.

If the "Update period" bit is additionally active when an error occurs during the writing process, the error message (if parameterized) has higher priority.

In this case the icon "Update required" is not displayed.

Info	Topic
10.5	Update required
14.3.4	File interface.ini

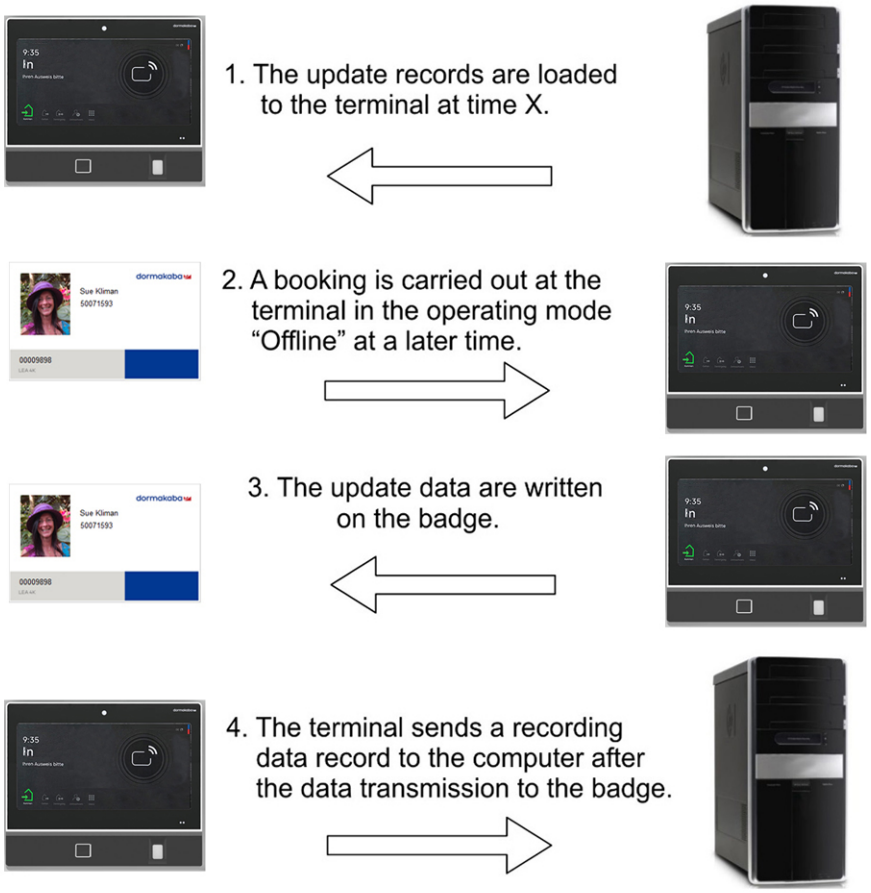
10.6 CardLink offline update records

General information

This chapter deals with the data records for the CardLink Offline Update option in both transmission directions (computer - terminal and terminal - computer).

The function allows data to be written to media by means of a terminal even if the terminal is in the "Offline" operating state. The data to be written are available as CardLink update data. They are loaded into the terminal by means of data records at any time and then saved in a special memory area. They are stored there in case of a possible power failure. If a booking is carried out later, these data are written to the medium if

- for the ID used for booking an update record is available.
- the active function key for the required data for CardLink Update was configured.
- the update data in the terminal are not marked as "written".



Schematic process of CardLink Update in the Offline operating state

By default the data is written to the medium irrespectively of the operating state of the terminal. However, the parameter setting of the function key functions (operating mode 7) can be used to specify that the write function is blocked in the "Online" operating state.

The data is transmitted to the terminal in the form of 7-bit ASCII characters. This means that 2 ASCII characters form a useful data byte. The values 0-9, A-F and a-f are allowed. After transmission the ASCII characters are converted into 8-bit data and saved in the terminal.

After successful completion of a writing operation to the medium, the data are marked as "written" in the terminal and no longer included in the next booking.

If a connection to the host is in place, it will be notified of the result of the writing operation via a registration record.

Launch

If the segments to be overwritten on the medium are read-only, the reader must be launched before overwriting this area. For launching, a so-called "launch card" RTM 63 must be introduced into the scanning field of the LEGIC reader **for at least 20 seconds**.

Launch data in the terminal are not deleted by a cold start. Existing launch data can be requested by sending a Ttnn record.

Info	Topic
3.10.30	Function key as dialog key
10.7	CardLink Update Online
5.2.10	Ttnn record: Request launch data
7.3.7	St record: Report launch data
3.10.34	Function key functions, operating mode 7

10.6.1 Update record download

Update data to be written to a medium are loaded to the terminal by means of one or several update records. The maximum size of the update data that can be written to a medium is 1024 bytes.

One update record can be used to transfer max. 100 bytes in the form of 2 ASCII characters per byte. Therefore, several update records may be necessary to write completely new update data to a medium. The order of the update records for a medium is defined by continuously numbering the record for this medium ID.

The minimum size of the update data in the data record is 1 byte (= 2 7-bit ASCII characters).

The number of update records that can be loaded to the terminal depends on the licensed memory configuration. Always the double number of update records than master records can be loaded into the terminal. If the permitted number is exceeded for an update record download, the terminal generates an O2 error record (command cannot be executed at the moment).

GID DID ON/OFF	Group address Device address Operation mode
V 0	Record type and record type modification see following description
n ... n	ID no.; number of digits 5-20, depending on definition.
n n	Record number, see the following description
n	Status byte, see following description
n	Active byte, see following description
n ... n	14-digit segment name (direct entry or reference to mediadef.ini) or 14 digits application designation
n n n n	4-digit memory address (direct entry or reference to mediadef.ini) or 4 digits offset
n ... n	Data min. 2 ASCII characters max. 200 ASCII characters Permitted characters: 0-9, A-F, a-f see following description

Record number

The record number must be assigned continuously to every medium ID if more than one update record is to be sent for this ID. Thus, the record number determines the order of the data to be written. The numbering starts at 01.

The fields

- Active byte
- Segment name/Application designation
- Memory address/Offset

are relevant only for the **first** transmitted data record. If several V0 records are transmitted one after another, these entries are no longer considered for the following records. In this case, the data are simply written consecutively into the segment of the file on the medium which has been defined in the first data record.

Status byte

The status byte defines whether further data records will follow or not. The entry "1" means that additional data records are being transmitted from the computer to the terminal. From the perspective of the terminal this means that in case of a "1" further data records are expected. If the terminal is to send the UID in the registration record, "2" is entered in the status byte of the last data record.

Entry	Meaning
0	Last data record
1	Further data records follow
2	Last data record, plus the request to the terminal also to send the UID.

Active byte

The active byte specifies whether the update records are to be written during the next booking. The active byte of a data record is automatically set to inactive after the writing operation has been completed. The data of the inactive data record are not written during the following booking.

Entry	Meaning
0	Data record is inactive since it has been already written.
1	Data record active, will be written with next booking.

Segment name/Application designation

See explanations in chapter [▶ 10.3.1] and chapter [▶ 10.3.3].

Memory address/Offset

See explanations in chapter [▶ 10.3.2] and chapter [▶ 10.3.4].

Data

The data is transmitted to the terminal in the form of 7-bit ASCII characters. This means that 2 ASCII characters form a useful data byte. The values 0-9, A-F and a-f are allowed. After transmission the ASCII characters are converted into 8-bit data and saved in the terminal. During a booking the data is written to the medium. The data must meet the following conditions:

1 The update record must contain **at least 2 ASCII characters** of data.

2 The update record may contain **a maximum of 200 ASCII characters** of data.

3 The number of ASCII characters in the area "Data" must be an **even number**.

If one of these conditions is not met, the dialog response "O1" (= incorrect record contents) is generated by the terminal.

10.6.2 Delete update record

An update record is deleted from the terminal by means of the following data record. To this end, the medium ID whose associated update record is to be deleted is entered.

If all update records in the terminal are to be deleted, wildcards (characters *) can be entered instead of the medium ID. The number of wildcards must correspond to the definition of the parameter IDLength in the application.ini file in section [MasterRecordDefinition].

NOTICE

Note that the wildcards must be used with caution, because all update records are lost after the delete download.

GID DID ON/OFF	Group address Device address Operation mode
V 1	Record type Record type modification
n .. n	ID no.; number of digits 5-20, depending on definition.

10.6.3 End of update record upload/download

The following data record reports the end of an up- or download procedure. It must be transmitted after an update record download at the terminal as identification of the end after the last data record. The terminal sends this record automatically to the computer after the upload of the update record has been completed.

GID DID ON/OFF	Group address Device address Operation mode
V 9	Record type Record type modification

10.6.4 Requesting update record

An update record is requested by means of the following data record. To this end, the medium ID whose associated update record is to be requested is entered.

If all update records are to be requested, the corresponding number of wildcards (characters *) can be entered instead of the medium ID. The number of wildcards must correspond to the definition of the parameter IDLength in the file application.ini in section [MasterRecordDefinition].

After the successful upload of one or several update records the terminal sends a V9 record as identification of the end.

GID DID ON/OFF	Group address Device address Operation mode
V 8	Record type Record type modification
n .. n	ID no.; number of digits 5-20, depending on definition.

Info	Topic
10.6.3	End of update record upload/download (V9 record)

10.6.5 Report CardLink update record

The terminal transmits the following data record after requesting via a V8 record containing the ID number.

GID DID ON/OFF	Group address Device address Operation mode
V 0	Record type Record type modification
n ... n	ID no., variable length 5 to 20 digits. Default: 7 digits
n .. n	Content of the update record

Info	Topic
10.6.1	Update record
10.6.4	Request update record

10.6.6 Data record update

The write operation to the medium is logged in a data record. The following information is saved in it:

- Media contents according to parameters
- Result of the writing operation
- UID if requested with status byte.

GID DID ON/OFF	Group address Device address Operation mode
n %	Record type and record type modification see following description
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day.
h h m m (s) (s)	Time Seconds only if set as parameter.
n ... n	6 digits identification of the user language Only if set as parameter.
n	Reserved
n ... n	Media contents according to parameters, variable length
n	Information byte, see following description.
n n	Status byte
n ... n	UID; variable length (max. 10 bytes = 20 characters)

Record type/Record type modification

The record type of data record is defined by means of the function key configuration. The record type modification is "%".

Information byte

The information byte with its entry contains information on whether the data transmission to the medium has been successful. The entries "B" and"C" are valid for MIFARE readers only.

Entry	Meaning
0	Update data have been written successfully
1	Update data could not be written
2	Incomplete update data
4	No writing possible with connected reader
5	Validation data could not be written
6	The medium was removed before writing took place
B	CardLink application could not be read
C	CardLink application name entry invalid

Note:

1. The entry 2 is output if the last data record of the update data has not been received by the terminal. In this case the data are incomplete and cannot be written to the medium.

2. The entry 6 is also output if the medium was removed from the scanning field just for a short period of time.

Info	Topic
13.4	Table of clock identifiers
10.7.2	Status byte and UID

10.7 CardLink update records in online mode

General information

This chapter deals with the data records for the CardLink Online Update option in both transmission directions (computer - terminal and terminal - computer). The TA2 option is required in addition to the CardLink Online Update option.

This function allows data to be written to media by means of an online terminal, for example for CardLink. An extended version of the G dialog is used for this purpose. The dialog is started by means of any booking at the terminal. The computer transmits the G2 dialog record "Write data" to the terminal. Several G2 records can be transmitted one after another. The completion of the writing operation is acknowledged by the terminal by sending a G3 record to the computer. After that the computer can transmit an acknowledgment response, for example R1.

The data are transmitted in the form of ASCII characters.

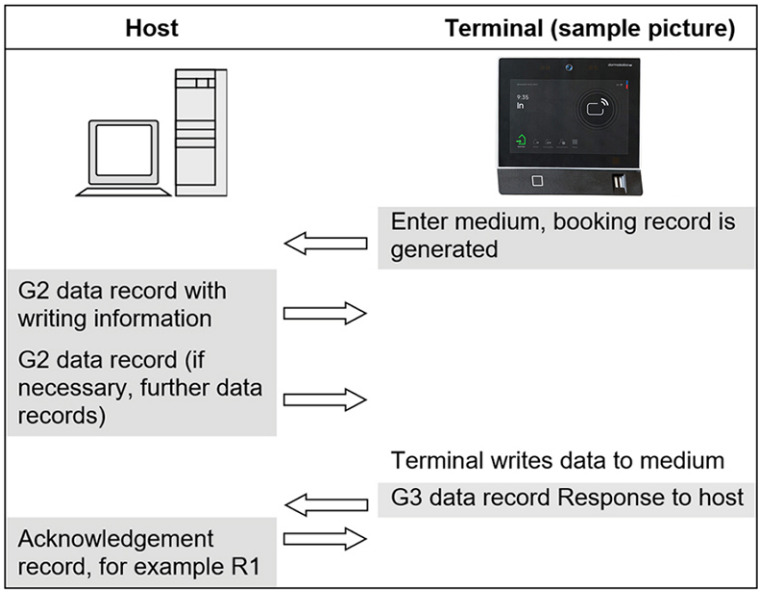
Launch

If the segments to be overwritten on the medium are read-only, the reader must be launched before overwriting this area. For launching, a so-called "launch card" RTM 63 must be introduced into the scanning field of the LEGIC reader for **at least 20 seconds.**

Launch data in the terminal are not deleted by a cold start.

NOTICE

The medium must have a segment defined for the write process of the file (Mifare). It must be kept in the reader field during the entire writing operation. If it is removed too soon, not all data can be transferred to the medium. In addition, an error message is generated at the computer (G3 record).



Schematic diagram of a CardLink Online Update procedure.

10.7.1 Dialog record Writing data

The G2 dialog record for writing data to media has the following structure:

GID DID ON/OFF	Group address Device address Operation mode
G 2	Record type and record type modification see following description
n n	Dialog text see following description
n	Status byte, see following description
n ... n	14-digit segment name (direct entry or reference to mediadef.ini) or 14 digits application designation
n n n n	4-digit memory address (direct entry or reference to mediadef.ini) or 4 digits offset
n ... n	Data min. 2 ASCII characters max. 200 ASCII characters Permitted characters: 0-9, A-F, a-f see following description

Note:

The fields

- Dialog text
- Segment name/Application designation
- Memory address/Offset

are relevant only for the first transmitted data record. If several G2 records are transmitted one after another, these entries are no longer considered. In this case the data are simply written one after the other into the defined segment on the medium.

If the G2 record is transmitted without previous booking at the terminal, the terminal generates the dialog response "O2" (= command cannot be executed at the moment).

Dialog text (default texts)

The dialog texts guide the users at the terminal. As the write operation requires the medium to be positioned the entire time in the reader field, a corresponding dialog text should be defined. The definition of the dialog texts is carried out by means of a corresponding download record.
Note: BI is the abbreviation for "Block identification".

Dialog text no.	BI (compatible)	BI (multilingual)	Dialog text
00	D00	d00	Present badge
01	D01	d01	Enter code
02	D02	d02	Present 2nd badge
03	D03	d03	Enter password
04	D04	d04	Enter personal no.
05	D05	d05	Enter reason
06	D06	d06	Enter position
07	D07	d07	Enter cost center
08	D08	d08	Enter project
09	D09	d09	Enter internal order
10	D10	d10	Enter object type
11	D11	d11	Enter object no.
12	D12	d12	Dialog text 12
13	D13	d13	Dialog text 13
14	D14	d14	Dialog text 14
15	D15	d15	Dialog text 15

Info	Topic
3.19.2	Dialog texts

Status byte

This entry defines whether additional data records follow or not and whether the data can be written on the medium. The entry "1" means that additional data records are being transmitted from the computer to the terminal. Only the last data record with the status byte "0" or "2" is acknowledged with a G3 record by the terminal after the writing operation.

Entry	Meaning
0	Last data record
1	Additional data records follow
2	Last data record, plus the request to the terminal also to send the UID.

Segment name/Application designation

See explanations in chapter [▶ 10.3.1] and chapter [▶ 10.3.3].

Memory address/Offset

See explanations in chapter [▶ 10.3.2] and chapter [▶ 10.3.4].

Data

The data is transmitted to the terminal in the form of 7-bit ASCII characters. This means that 2 ASCII characters form a useful data byte. The values 0-9, A-F and a-f are allowed. After transmission the ASCII characters are converted into 8-bit data and saved in the terminal. During an authorized booking the data is written to the medium.
The data must meet the following conditions:

1 The update record must contain **at least 2 ASCII characters** of data.

2 The update record may contain **a maximum of 200 ASCII characters** of data.

3 The number of ASCII characters in the area "Data" must be an **even number**.

If one of these conditions is not fulfilled, the dialog response "O1" (= incorrect record contents) is generated by the terminal.

Info	Topic
7.2	Error records

10.7.2 Dialog record Dialog response

The terminal sends the following G3 dialog record to the computer after writing data to the medium. The last G2 record of the computer with the status "0" (last data record) or "2" (last data record with UID request) must have been transmitted beforehand.

After having received the G3 record, the computer can send a positive or negative booking response (e.g. R1, also with text) to the terminal. Depending on the configuration, it is possible, for example, to open a door.

The G3 record has the following structure:

GID DID ON/OFF	Group address Device address Operating mode
G 3	Record type and record type modification see following description
n	Information byte, see following description
n n	Status byte, see following description
n ... n	UID; variable length, (max. 10 bytes = 20 characters), if requested in the status byte of the G2 record

Information byte

The information byte informs on the result of the writing operation.

Entry	Meaning
0	Data have been written successfully.
1	Data could not be written
2	Time-out elapsed (no G2 record with status "0" or "2" received).
3	Overflow, i.e. the number of transmitted characters has exceeded the number that can be written. The terminal application can process max. 1024 bytes irrespective of the medium size.
4	No writing possible with the connected reader.
6	The medium was removed from the reader field before terminating the write operation.
C	CardLink application name entry invalid

Status byte reader type LEGIC

01-06: General error.

Status byte reader type MIFARE

In case of error (information byte not = 0), the reader-specific error code can be found at this position; otherwise, "00". The reader-specific data are not for evaluation.

Status byte	**Meaning or possible cause**	**Measures or possible solution**
01	Communication lost: The media was removed from the RF field	Leave the media in the RF field until the process has been finished.
02	Authentification failed: Application was defined using the fabrication key Application does not exist on the medium (Mifare Classic) Error while creating the actuator status file (incorrect sitekey).	 Enable Fabkey Replacement Create the application. Correct the site key search order. The first entry of the search order is used for creating the file.
04	File not found on media.	Create the file/application.
08	This media is not selected: The media was removed from the RF field	Leave the media in the RF field until the process has been finished.

Unique number (UID)

The hexadecimal UID is entered in the data record after an error-free writing operation on the medium. The UID is variable and may be maximum 10 bytes long (= 20 characters).

10.8 Media booking history (MediaTraceback)

The media booking history allows you to reproduce which bookings have been made with which medium, when and where within the installation. This requires a "Media traceback" file (MIFARE DESFire or LEGIC advant) on the medium; e.g. generated by the dormakaba media manager.

Information about the location and time of the booking are written to this application when booking on a standalone component. During the following booking at a terminal, this information can be read out in a G dialog.

Hereby, the following applies:

- The media traceback file is only read out if the host is active.
- Data from the medium are not buffered in the terminal.
- The medium must be positioned in the RF field of the reader during the entire process.

Transparent reading of data segments

The data records described in the following chapters (G dialog records) allow to read data from various storage areas of a medium.

To do so, the relating key words (MEDIATRACEBACK or APPLICATION) must be entered in the dialog record.

For LEGIC media it is also possible to enter the designation of the segment directly.

10.8.1 Dialog record Request media data

This dialog record is used to read data from

- the media traceback file or
- an application file

of the medium.

The entire process is structured as G dialog.

GID DID ON/OFF	Group address Device address Operation mode
G 4	Record type Record type modification
n n	Dialog text no. see following description
0	Reserved
n ... n	Segment name: MEDIATRACEBACK or APPLICATION or direct segment addressing admissble. or Application designator: MEDIATRACEBACK or APPLICATION admissible.
n n n n	4-digit memory address (direct entry only) or 4 digits offset
n n n n	4 digits maximum length of data to be read out

Dialog text no.

The dialog texts guide the users at the terminal. As the read operation requires the medium to be positioned the entire time in the RF field during reading, a corresponding dialog text should be defined. With 2 hyphens "- -", a dialog text from the previous booking remains unchanged.

If more than one G4 dialog record is used, the details for the dialog text no. are not required in the following dialog records.

Segment name

Only the designators

- MEDIATRACEBACK or
- APPLICATION or
- direct segment designator

are admissible.

Other keywords are **not** admissible.

See explanation in chapter [▶ 10.3.1].

Application designation

Only the designators

- MEDIATRACEBACK or
- APPLICATION

are admissible. If more than one G4 dialog record is used, the details for the application designation are not required in the following dialog records.

See explanation in chapter [▶ 10.3.3].

Memory address

Only direct entry admissible.

See explanation in chapter [▶ 10.3.2].

Offset

The offset specifies from which position of the application the data to be read can be found. The default setting for the offset is 0, i.e. from the beginning. This indication may be omitted; in this case, the default value is applied.

See explanation in chapter [▶ 10.3.4] .

Length

The length specifies the maximum number of bytes of the data to be read. The maximum value is 1024. If fewer bytes than defined in the length are read, only the existing number of bytes is transferred. The default setting is "no value" so that the complete data volume is always read. This indication may be omitted; in this case, the default value is applied.

10.8.2 Dialog response Report media data

After requesting the media from the traceback file or the application file by means of the G4 record, the terminal sends a G5 record in response. A G5 record is transferred in response to each G4 record. Each G5 record contains a maximum number of 100 bytes of useful data.

If you want to read the entire data volume of 1024 bytes out the media traceback file, 11 G5 records are required as response, which must be requested one by one by means of a G4 record.

If there are no more data in the media traceback file, this is signaled by the status identification "0" (last data record) in the status byte of the G5 record.

After having received the last G5 record, the host can send a positive or negative booking response to the terminal to terminate the booking process.

In case of error (result in information byte not = 0), the UID and the useful data are not transferred.

The G5 record for transferring the readout data is structured as follows:

GID DID ON/OFF	Group address Device address Operation mode
G 5	Record type Record type modification
n	Information byte, see following description
n n	Status byte, see following description
n ... n	20 digits UID
n ... n	Max. 200 digits for ASCII characters (100 bytes of useful data).

Information byte

The information byte informs about the result of the read operation.

Entry	Meaning
0	Data could be read
1	Data could not be read
3	Length in G4 record > 1024.
4	No reading possible with connected reader.
6	Medium not in reader field.
C	Entry is not MEDIATRACEBACK or APPLICATION.

Status byte

This entry is used to specify whether when no error has occurred further data records follow or not.

Entry	Meaning
0	Last data record
1	Further data records follow

UID

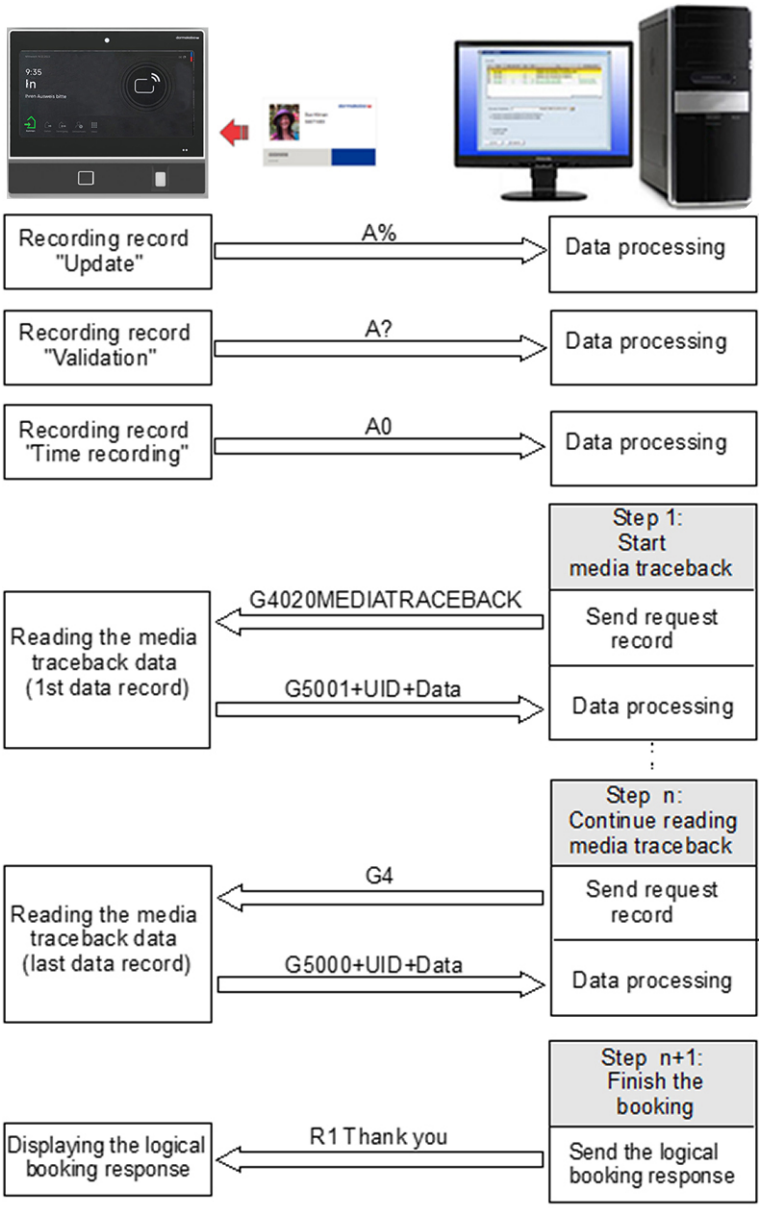
Identification of the medium used.

Useful data

Media data; max. 100 bytes in each data record.

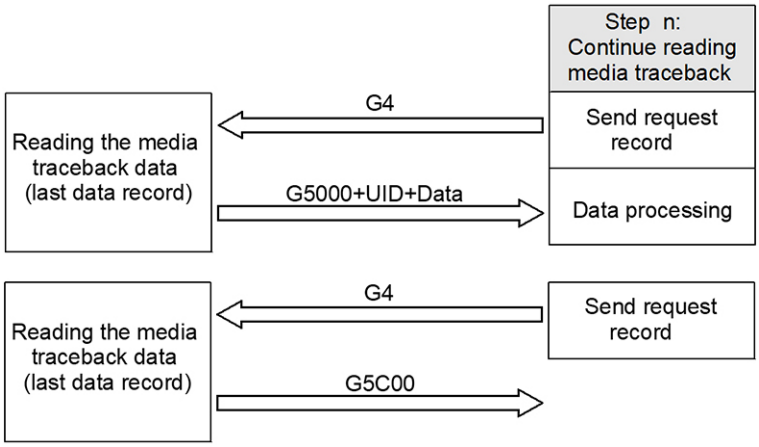
10.8.3 Example of a G dialog

The following graphic shows an exemplary sequence of a G dialog started by pressing the function key F00 with update and validation and subsequent readout of the media traceback file.



The following graphic shows a variation of the previous example. After reading out the last data record, a query is sent again; this query, however, only contains SA/SAM G4. It is considered by the terminal as new query dialog but without having specified the application designation MEDIATRACEBACK.

The terminal responds to this query record with a G5 record and the entry "C" in the information byte (= entry is not MEDIATRACEBACK).



10.8.4 Dialog record Deleting the MediaTraceback file contents

With the following data record, the contents of the media traceback file is deleted from a medium.

GID DID ON/OFF	Group address Device address Operation mode
G 6	Record type Record type modification
n n	Dialog text no.
0	Reserved
n .. n	Segment name Only MEDIATRACEBACK admissible or Application designation Only MEDIATRACEBACK admissible

Dialog text no.

The dialog texts guide the users at the terminal. As the delete operation requires the medium to be positioned the entire time in the reader field, a corresponding dialog text should be defined.

Segment name

Only MEDIATRACEBACK admissible.

Application designation

The key word MEDIATRACEBACK must be entered for deleting media traceback data. The key word **may only be entered as a whole word with capital letters**. Other key words are not permitted.

10.8.5 Dialog response Report result of deleting process

The terminal sends the following G7 record in response to the G6 instructions for deleting the contents of the media traceback file.

After having received the G7 record, the host can send a positive or negative booking response to the terminal to terminate the booking process.

In case of error (result in information byte not = 0), the UID and the useful data are not transferred.

The G7 record for transmitting the result of the deletion is structured as follows:

GID DID ON/OFF	Group address Device address Operation mode
G 7	Record type Record type modification
n	Information byte, see following description
n n	Status byte
n .. n	20 digits UID

Information byte

The information byte informs about the result of the delete operation.

Entry	Meaning
0	Data could be deleted.
1	Data could not be deleted.
4	No deletion possible with connected reader.
6	Medium not in reader´s RF field.
C	Entry is not MEDIATRACEBACK.

Status Byte

In case of error (information byte = 1), the reader-specific error code can be found at this position; otherwise, "00".

UID

Identification of the medium used.

10.9 Actuator status file

Reader type MIFARE

The actuator status file is an application on a medium in which the standalone component can store its status information. This application can be created on the medium via a terminal.

The information required for creating the actuator status file must be provided by the host system, e.g. dormakaba Media Manager.

If the "Create Actuator Status file" is active, it will be executed in connection with the "Write CardLink data" (G2 record) or "Update record download" (V0 record) function if

The application designation used in the records is "CARDLINK" or "CARDLINK1" to "CARDLINK16".

In a simple validation process, the actuator status file is **not** created.

The "Create actuator status file" function is **disabled** after a cold start. An actuator status file is **not** created. If you want to use this function, **enable it** by means of a data record.

If an error occurs when creating the actuator status file, this is reported in the status byte of the G3 data record or the update data record. The actuator status file is only created after a successful write operation. (Information byte of the record contains the value "0" = data successfully written).

Info	Topic
10.7.2	G3 data record
10.6.6	Update data record
10.7.1	G2 data record
10.6.1	Update record download

10.9.1 Data record Enabling/disabling creation of actuator status file

Reader type MIFARE

The following data record enables/disables the "Create actuator status file" function in the addressed terminal.

GID	Group address
DID	Device address
ON/OFF	Operation mode
T	Record type
t	Record type modification
A	Identifier
n	see following table.

Identifier

Entry	Meaning
8	Disable creation of the actuator status file.
9	Enable creation of the actuator status file.

11 Biometrics

11.1 General information

BIO

A biometric reader (CBM = Compact Biometric Module) can also be used as a hardware option in the B-Client HR40 application. The biometric data (templates) are normally saved both centrally on an FTCS server and locally in the database of the CBM. When the number of employees is very small, the data storage can be limited to the CBM. This can be specified by means of a "Standalone" parameter in the ../system.ini file.

The local data storage includes up to 1,000 templates (= 500 persons, each with 2 templates) in the standard version. CBM modules up to a max. of 10,000 templates are available.

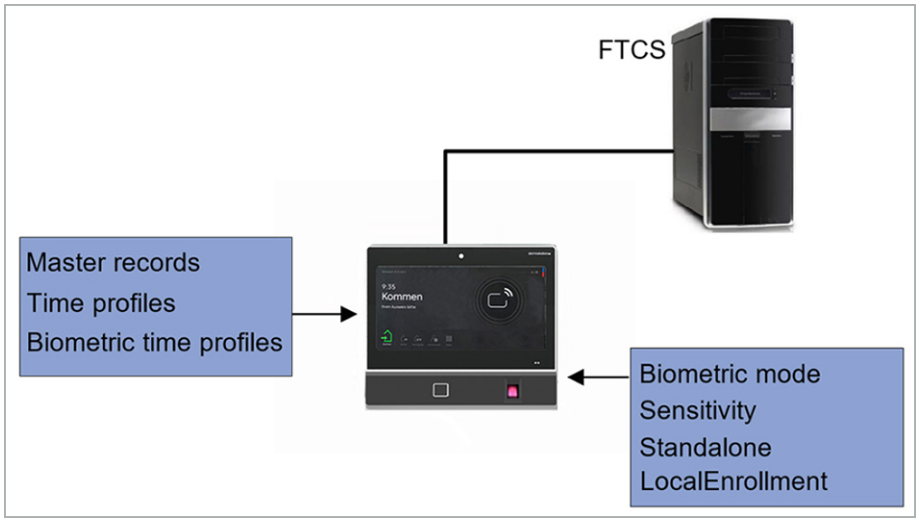
The user guidance is carried out on the display. Small graphics are displayed for visualization for the functions "Enroll" and "Identify" in case of an error. The "Enroll" function serves as a tool to register fingerprints directly at the terminal.

Persons with indistinctive biometric features can use the function "Alternative ID entry". The alternative ID entry must be released via the biometrics byte in the master record. This can be done via

- the badge (RFID) or
- a keypad (if available).

11.2 Biometrics overview

The following overview shows the most import download records and parameters in connection with biometrics. The individual parameters are described in the following chapters. Master records and time profiles are described in the chapters stated below.



Info	Topic
5.5	Master records
11.4.7	Time profiles

11.3 Biometric parameters in the system.ini file

The configuration parameters of the biometric reader are summarized in the "system.ini" file. The "Default" data of a certain parameter in the system.ini indicate that the setting in question is active in the terminal if no other setting was made. If parameters mentioned here are not contained in the "system.ini" file in the [Section] mentioned, they must be created and assigned the desired parameter value, if necessary. To do so, open the file with a text editor, edit or create the desired parameters and save. It can be useful to make a backup copy of the original "system.ini" file.

Info	Topic
14.1.4	Sections in the "system.ini" file

11.3.1 "Standalone" biometric parameter

The "Standalone" parameter defines whether the terminal provides the templates only locally in the CBM or whether a connection to the FTCS is required. For systems with FTCS, the templates and their ID numbers are administered centrally on the FTCS server and sent to the individual biometric terminals. The templates or their ID are always checked directly in the CBM module of the terminal.

The parameter setting is described in the [Reader1CBM] section of the "system.ini" file.

11.3.2 "ProximityScale" biometric parameter (sensitivity)

In idle state, the sensors of the biometric reader are off. The "ProximityScale" parameter defines the switch-on threshold at which the sensors are activated upon approaching a finger.

The parameter setting is described in the [Reader1CBM] section of the "system.ini" file.

11.3.3 "BiometricMode" biometric parameter

The BiometricMode parameter defines the biometric operating mode of the terminal. One of 5 operating modes can be activated. The following table shows an overview of all 5 biometric operating modes. Each operating mode will be described below in detail.

The parameter setting is described in the [Reader1CBM] section of the "system.ini" file.

11.4 Use of biometric readers

Terminals of the dormakaba terminal 97 xx series in connection with the B-Client HR30 software can operate CBM biometric reader modules. Devices with biometric reader can be operated in the following authentication modes. The modes 2 to 5 require an additional RFID reader.

- Mode 1: Identification
- Mode 2: Verification
- Mode 3: Verification by ID comparison
- Mode 4: Combination of modes 1 and 2
- Mode 5: Combination of modes 2 and 3

The authentication mode is defined by the BiometricMode parameter in the [Reader1CBM] section of the system.ini file.

The authentication mode cannot be changed during operation.

To be able to handle data as normal badge booking, predefined strings (PresetBooking) can be used to format reader data.

Info	Topic
14.1.4	Sections in the "system.ini" file

11.4.1 PresetBooking

The template ID or the badge number entry via keypad can be extended using a default value. In doing so, the template ID or the keypad input data are superimposed on the default value starting from the right.

Allowed values are the characters "0" to "9", "A" to "F", and the special characters ":", ";", "<", "=", ">" and "?". The default value length is unlimited.

Due to the combination of the ID length in the CBM module, the ID length of the master record, and the complete badge definition, particular attention has to be paid to a consistent and conclusive definition.

The preset booking is defined by generating a parameter named PresetBooking=xx in the related [Reader1CBM] section of the system.ini file.

Example:

[Reader1CBM]

.....

PresetBooking=000000000200000000000

Data delivered:

000012345

PresetBooking=000000000200000000000

This corresponds to the subterminal standard booking.

The badge definition requires a 5-digit ID. The read ID is aligned to the right side. The default value is also entered right-justified.

This gives the following booking data:

000012345	Reader data
12345	Adapted length
000000000200000000000	PresetBooking
000000000200000012345	Booking record

The created data record is compatible to the standard badge definition (22 digits) and can be processed. The standard badge definition is:

- 8-digit customer number (02000000)
- 1-digit version (0)
- 5-digit ID number (12345)

Info	Topic
14.1.4	Sections in the "system.ini" file

11.4.2 Biometric mode 1

Biometric identification (default setting)

The authentication mode 1 stands for a mere identification that is a comparison between a template which is generated from a read-in fingerprint and all other templates which are stored in the CBM module.

In the authentication mode 1, the CBM module provides the control unit only with the ID number related to the template. Therefore, the ID number must be integrated in the badge definition for further processing in the host system. Integration is done by extending the ID number using the parameter "PresetBooking".

Booking procedure

1 The person activates the biometric sensor by pressing a key or approaching a finger to the sensor.

2 The person books with a finger.

3 Identification (1:n comparison) of the ID number belonging to the template of this person using the ID numbers saved in the CBM takes place.

4 A data record is generated from the previously mentioned parameter and the ID number.

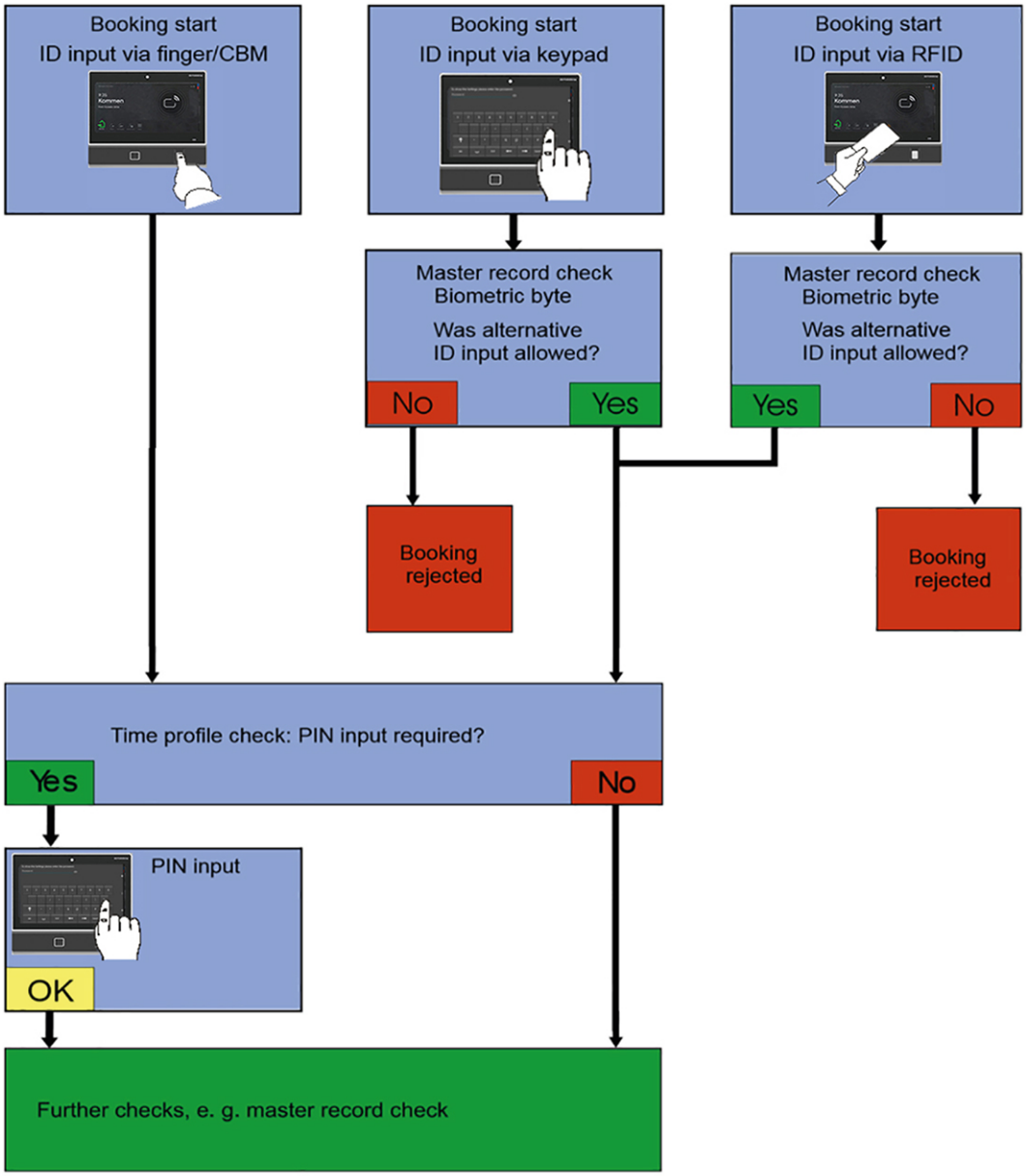
5 The generated data record is transmitted to the host.

With less pronounced minutia, a person may have the right to perform an alternative booking via the biometric byte of the master record.

This can be done by

- entry of the ID number via the keypad.
- retrieval of the ID number from the RFID badge.
- reading of the RFID badge or (alternatively) keypad input.

Graphic overview biometric mode 1



11.4.3 Biometric mode 2

Verification dependent on the time profile. A template is contained on the RFID badge.

Booking procedure

1 The person books using his RFID badge. Template and ID are entered.

2 The biometric sensor is activated (depending on the biometric time profile). If the biometric time profile is not active or is outside a defined time span, a simple RFID booking without biometrics is allowed.

3 The person books with a finger. A template is formed.

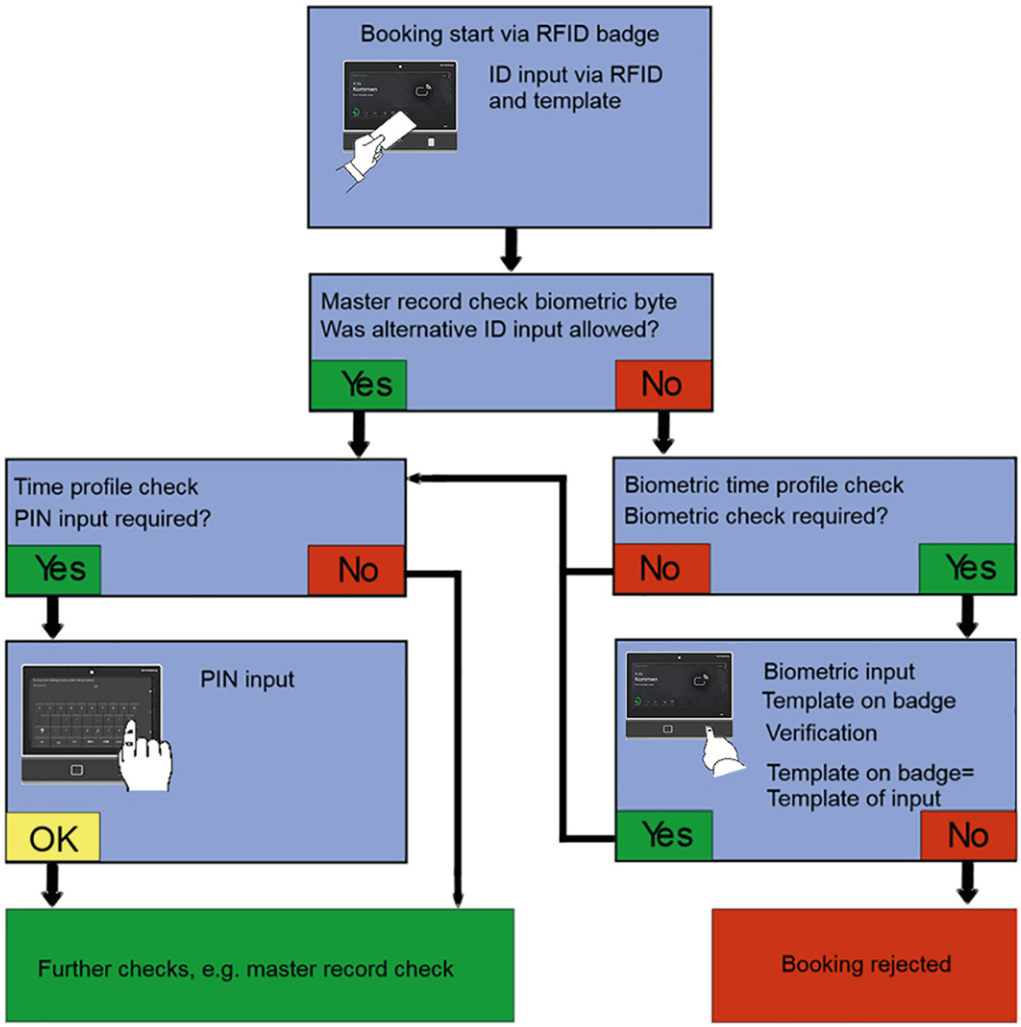
4 Verification (1:1 comparison) between the template of the RFID badge and the template of the registered finger takes place in the CBM.

A failed verification is rejected with the message "Not authorized" (terminal text T19). The error identification F "Verification unsuccessful" is entered in the data record.

Alternative booking:

- Retrieval of the ID number from the RFID badge.

Graphic overview biometric mode 2



11.4.4 Biometric mode 3

Verification dependent on the time profile using 2 features (RFID ID and biometrics).

This mode can be used in particular when existing badges are not capable of saving a biometric segment.

Booking procedure

1 The person books using his RFID badge. The ID is entered.

2 The biometric sensor is activated (depending on the biometric time profile). If the biometric time profile is not active or is outside a defined time span, a simple RFID booking without biometrics is allowed.

3 The person books with a finger. A template is formed.

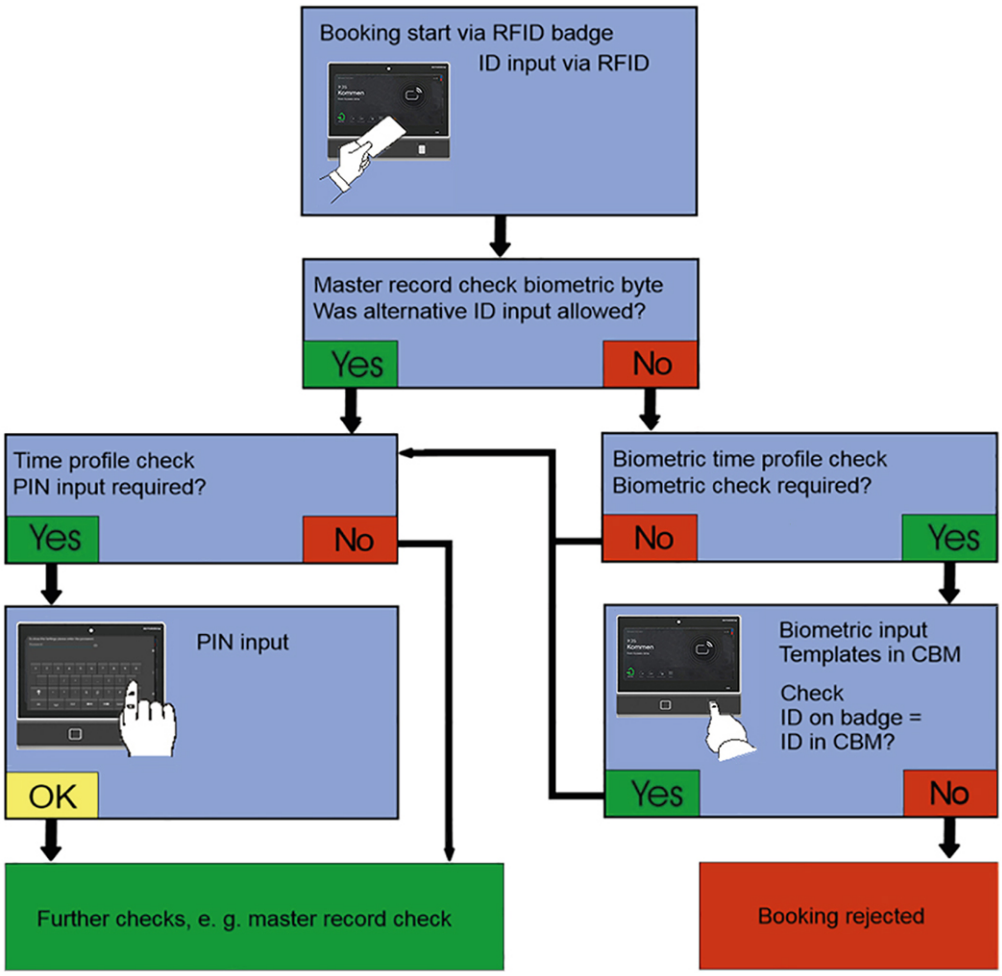
4 Verification (1:1 comparison) between the ID of the RFID badge and the ID of the template produced takes place in the CBM.

A failed verification is rejected with the message "Not authorized" (terminal text T19). The error identification F "Verification unsuccessful" is entered in the data record.

Alternative booking:

- Retrieval of the ID number from the RFID badge.

Graphic overview biometric mode 3



11.4.5 Biometric mode 4

Verification dependent on time profile or biometric identification in mixed operation..

Biometric mode 4 summarizes the properties of biometric modes 1 and 2. The **beginning of the booking** decides the further booking sequence.

Booking procedure

- Finger is placed on the biometric sensor

  - Biometric mode 1, biometric verification

- RFID badge containing the biometric segment is presented

  - Biometric mode 2, biometric verification

Version 1

1 The person activates the biometric sensor by pressing a key or approaching a finger to the sensor.

2 The person books with a finger.

3 Identification (1:n comparison) of the ID number belonging to the template of this person using the ID numbers saved in the CBM takes place.

Version 2

1 The person books using his RFID badge. Template and ID are entered.

2 The biometric sensor is activated (depending on the biometric time profile). If the biometric time profile is not active or is outside a defined time span, a simple RFID booking without biometrics is allowed.

3 The person books with a finger. A template is formed.

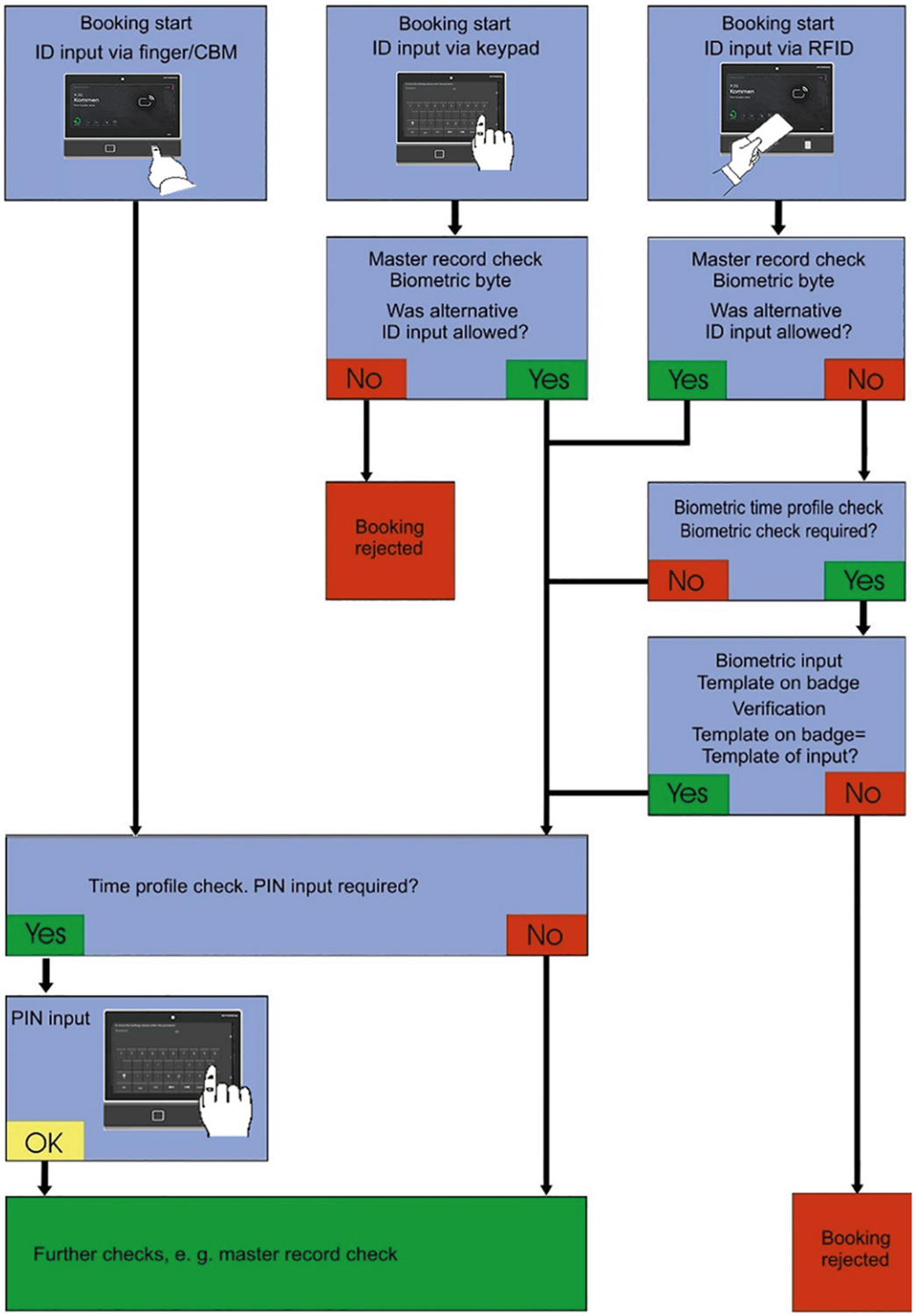
4 Verification (1:1 comparison) between the template of the RFID badge and the template of the registered finger takes place in the CBM.

A failed verification is rejected with the message "Not authorized" (terminal text T19). The error identification F "Verification unsuccessful" is entered in the data record.

Alternative booking:

- Entry of the ID number via the keypad.
- Retrieval of the ID number from the RFID badge.
- Reading of the RFID badge or (alternatively) keypad input.

Graphic overview biometric mode 4

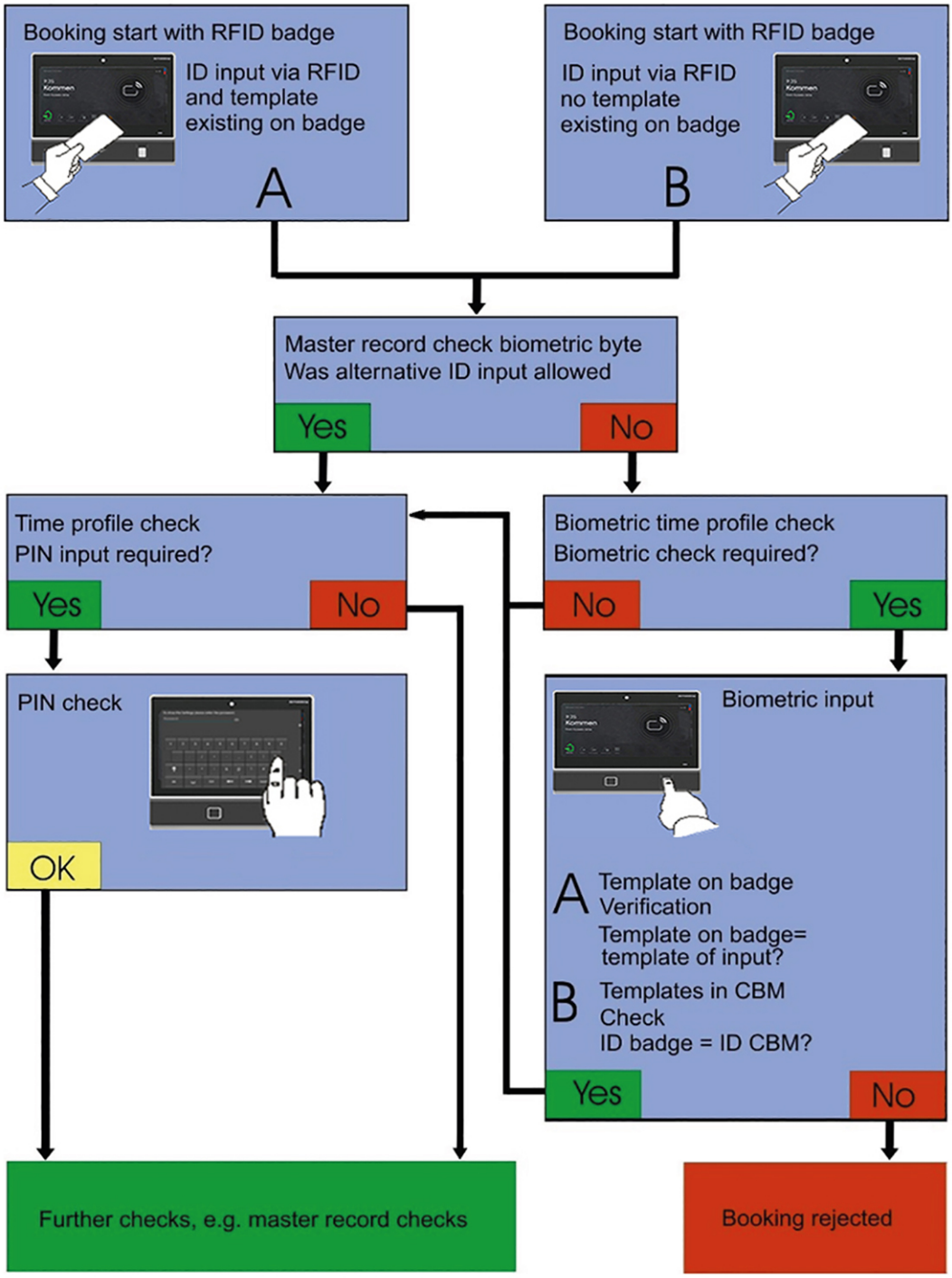


11.4.6 Biometric mode 5

Biometric mode 5 summarizes the properties of biometric modes 2 and 3. This allows mixed operation of badges with or without biometric templates at a reader.

In accordance with the presence of a biometric template, the booking sequence is as described in the chapters for Biometric modes 2 or 3, see chapter and chapter.

Graphic overview biometric mode 5



11.4.7 Biometric time profiles

The biometric time profile is evaluated in biometric modes 2, 3, 4 and 5. The biometric time profile is used to define at which intervals verification via the finger print is required in addition to reading the RFID badge. In contrast to the general time profile, the biometric time profile is not person-related. It applies equally to all persons. Validation via the biometric time profile is activated upon selecting the biometric operating mode.

1 biometric time profile can be stored in the terminal. A time profile can contain up to seven time pairs.

A time pair always consists of the parameters

- Time span
- Validity days and
- Identification byte

Profile name

Only the profile name 01hex is allowed.

The entry 00 means:

- The time profile is invalid or inactive
- Verification via the finger print is not required.

Standard factory setting

One time profile is stored in the terminal. It contains the following entries:

- 1 time pair 00.00 Uhr to 11:59 p.m.
- Validity days: Entry 7Fhex for all weekdays.

At these settings, verification in biometric modes 2, 3 and 4 is **always** required.

Info	Topic
11.4.7.4	Necessary download record

11.4.7.1 Time span

A time pair is always valid for a certain time span, for example 8:00 a.m. to 5:00 p.m., until the set minute expires. This means that the time span "8:00 a.m. to 5:00 p.m." ends at 5:00:59 p.m. This must be observed when defining time spans. If you want the time span to finish by definition, for example, at 5:00 p.m., the end time must be entered as "4:59 p.m.".

The required time span is defined by entering hours/minutes for "Start" and hours/minutes for "End". For undefined time spans, verification via the finger print is not required.

Possible values: 00:00 to 11:59 p.m. The entry "**24:00**" is **not** permitted.

If a time span is to be active for several days (e.g. from 10:00 p.m. until 08:00 a.m.), two time pairs must be defined.

Example:

10:00 p.m. to 11:59 p.m. for one day and 00:00 to 7:59 a.m. for the next day.

The times must not overlap. If a time span ends, for example, at 5:00 p.m., the next one can start at 5:01 p.m. at the earliest and not at 5:00 p.m.

Standard factory setting

There is a time span from 00:00 to 11:59 p.m.

Info	Topic
11.4.7.4	Necessary download record

11.4.7.2 Validity days

The parameter specifies the weekdays for which the defined time pair shall be valid. For a booking, all time pairs are checked until the time span and date match the current time and date.

Standard factory setting

Entry 7F. All weekdays have been defined as validity days.

Info	Topic
11.4.7.4	Necessary download record

11.4.7.3 Identification byte

The identification byte includes the special day identification.

Features:

- The special day identification assigns the biometric time profile to a special day.
- If the current day of a booking is identified as a special day, the time pairs are only checked for the special day identification.
- The check for validity days is skipped.
- All time pairs are checked until time span and special day identification match the current time and day.
- The definition of the special days and special day identifications is performed in the special day table.

Standard factory setting

No special day identification has been stored.

Info	Topic
3.18	Special day table
11.4.7.4	Necessary download record

11.4.7.4 Download record: Biometric time profile

The parameters described in the previous sections are modified in the terminal using the following download record. The entries correspond to the default values after a cold start.

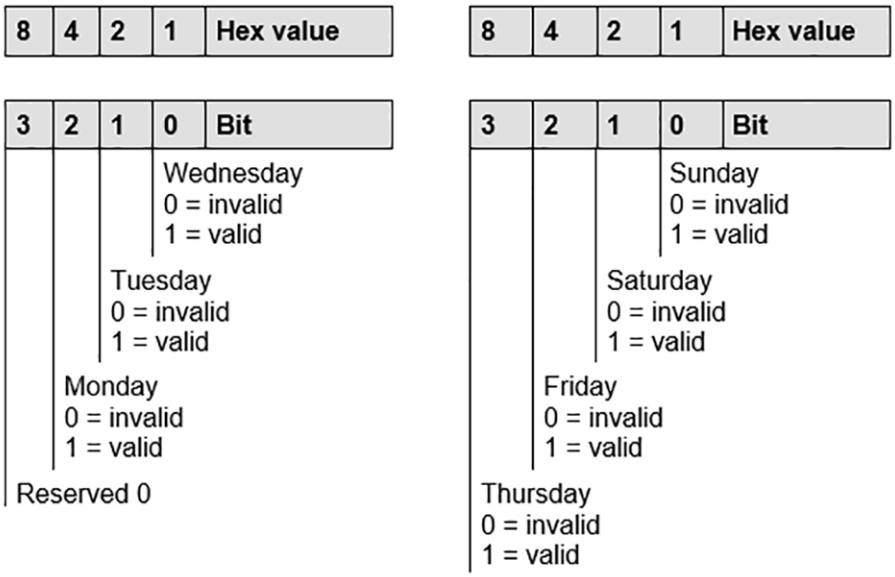
GID DID ON/OFF	Group identification Device address Operating state	
> 3	Record type Record type modification	
P Q 1	Block identification PQ1	
0 1	Profile name 01_{hex} 00 = no time profile	
0 0 0 0	Hour "Start" Minute "Start" — Time span	1st time pair
2 3 5 9	Hour "End" Minute "End"	
7 F	Validity days	
0	Identification byte	
0	Reserved 0	
. . .	A maximum of 6 further time pairs is possible	

Block identification

The biometric time profile has the block identification PQ1.

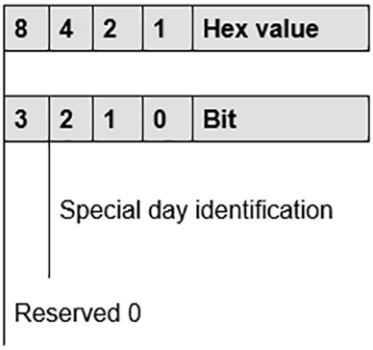
Validity days

The validity days are transmitted in two bytes. The terminal interprets the characters entered in the download record as a hex code.



Identification bytes

The terminal interprets the identification byte as a hex code.



Info	Topic
3.5.1	Hexadecimal coding

12 Mifare reader

12.1 General information

This chapter describes the integration and configuration of a dormakaba Mifare proximity reader in dormakaba terminals.

For a better understanding, a bullet point list gives you an overview of the underlying application concept.

dormakaba Mifare proximity readers are a part of an application concept developed by dormakaba which combines the use of different applications on a medium with a new ARIOS security concept. The maximum installation protection and, at the same time, an easy handling and high flexibility are guaranteed.

A central element is the security card. This security card generates the installation key and is highly encrypted. The installation key is never obvious. By means of the security card, the system operators have control over their system and their badges. This security card is protected additionally by a PIN/PUK system. The system operators keep their security card in-house even if they order badges.

The basic points of this application concept are:

- Each installation can be operated independently of other installations
- The installation key is automatically generated and always remains invisible
- An oral or written forwarding of the installation key is impossible
- Does not require any central point for managing the installation key
- Distribution of the installation key via programming master card (standalone components) or communication software (online components)
- Each application on a medium has its own, specifically calculated application key
- Use of a temporary intermediate key (production key) for third party suppliers to produce media
- Detailed logging of all important processes.

The present information is subdivided into the following topics:

- Overview of the possible components of an installation
- Inheritance hierarchy of the different cards
- Term definition
- Presentation of all possible data records.

12.2 Possible system configuration

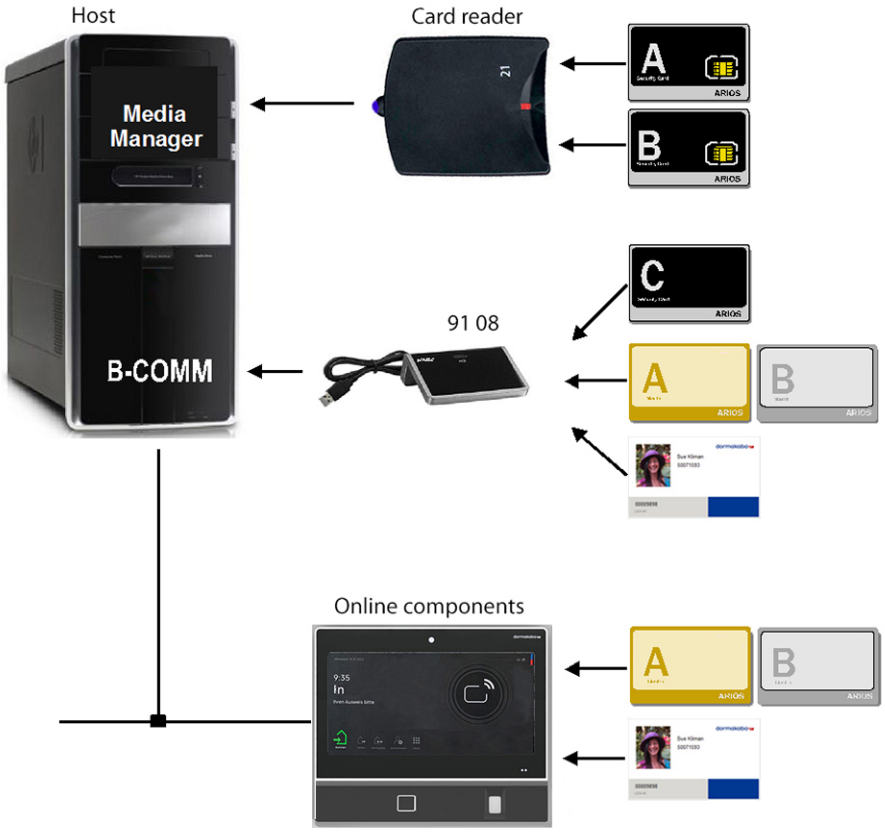
The authorization media allow the system operator to start up systems or to make changes. In this concept, three types of authorization media are used.

- Security cards of type A/B as smart card with chip
- Security cards of type C as RFID smart card
- Programming master cards of type A/B as RFID card

The following chart shows possible configuration options. Two possible application cases are shown together. Terminal: sample picture.

1 Creating cards over the card reader and the dormakaba Media Manager

2 Distributing the installation key via the dormakaba desktop reader 91 08. The installation key can be distributed over the programming master A/B to the terminals or the standalone components.

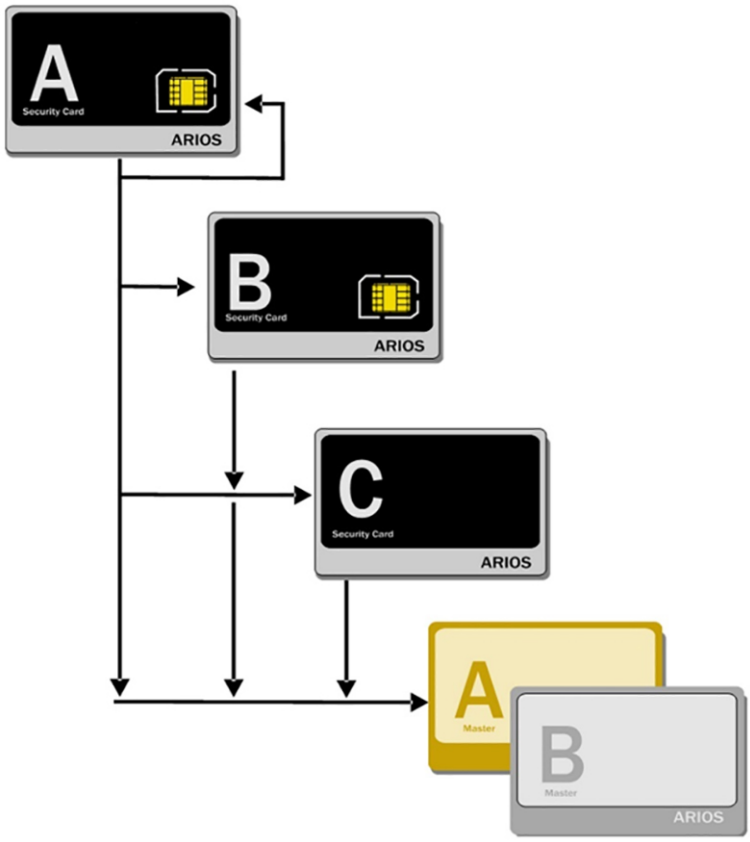


12.3 Card hierarchy and inheritance

The following graph shows the cards that can be used in a system and the possibilities for inheriting the card contents.

Hereby, the following applies:

- With a security card A, all other cards, including a copy of A, may be created.
- With a security card B, a security card C and the programming master cards A and B can be created, but no copy of B.
- With a security card C, only the programming master cards A and B can be created.



12.4 Term definitions

Installation key

The main key of the installation. It is generated using security card A or B, and it is never transparently available (private). It is used to generate all other keys in the security concept. It is stored in all components with a security chip. It is encoded by means of the security chip to distribute it in an installation.

Installation key check sum

The check sum of the installation key is a generated check sum that is unique for the installation key. It has a length of 8 bytes for each installation key.

Application key

The key to identify the application on the user medium.

Production key

The production key is independent of the unique number and only temporarily stored on the medium (between medium production and first use in the installation). The production keys are transferred to the card producer for producing cards. These keys are calculated within the security chip (in the security card) based on the installation key. When first being used in the installation, the security chip (in the desk reader or in an online terminal) changes back the production key to an individual application key of this medium.

dormakaba Media Manager

Software for Windows-based operating systems for a simple and secure generation and administration of authorization and user media.

Media

General term for badges, key tags or keys.

Parent info

Information about the inheritance of rights with programming master cards.

Programming master

Umbrella term for the programming masters A and B. Programming masters are RFID badges by means of which the standalone components can be initialized and configured. Programming masters can be configured specifically for the installation using the security card. They are used for distributing the installation key to all components of the installation.

Security card A

Contact smart card according to ISO 7816. It contains a security chip for generating the installation key and other safety-relevant functions. Each installation needs minimum one security card for initializing. Up to 8 installation keys/external application keys can be generated and stored together with the application configurations in the security chip. Copies of this data can be stored on further security cards A, B, and C (backup, multiplication). The security card A is provided by dormakaba.

Security card B

Description see security card A.

The security card B is created from a security card A; it contains the same data. Other copies cannot be generated using security card B.

Security card C

Contact-free Mifare DESFire RFID card. It is created from a security card A or B; it contains the same data. Other security cards C cannot be generated using security card C. In a standard configuration, the security card C is provided by dormakaba.

UID

Unique identification of a medium with a length of up to 24 bytes.

12.5 Application cases

The following table contains a short reference of the application cases described in this document. You will find data records and in-depth information about the different topics in the belonging sub-chapters.

Info	Field of application
12.6	Managing the installation keys
12.7	Distributing installation keys by means of programming masters
12.7.5	Exchanging production key against application key
12.8	Reporting the terminal status

12.6 Managing the installation keys

12.6.1 Downloading the installation key

With this data record, an installation key is sent to a terminal. There may be maximum 8 installation keys in a system. A check sum of the installation key can be transferred additionally. If a check sum is also transferred, an existing installation key is overwritten if the transferred check sum does not match the one of the set installation key.

If a check sum is not specified, the installation key is always overwritten.

If the designation of the installation key is outside the permitted range from 0 to 7, an O3 record (command not executable) is sent.

The response to the "TtSn" data record is given with the "StSn" record.

GID DID ON/OFF	Group address Device address Operating mode
T t	Record type Record type modification
S n	Identifier of the installation key, see following table
n .. n	192 digits installation key
n ... n	16 digits check sum

Identifier

Entry	Meaning
S0	Set sitekey 0
S1	Set sitekey 1
...	...
S6	Set sitekey 6
S7	Set sitekey 7

Info	Topic
12.6.2	Data record StSn

12.6.2 Result of installation key download

The terminal sends the following data record in response to a TtSn record. The StSn record indicates whether the download of the installation key was successful or not.

GID DID ON/OFF	Group address Device address Operating mode
S t	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
S n	Identifier of the installation key (S0 to S7)
n	Result, see following table.

Result

Entry	Meaning
0	Writing the installation key was successful
1	Writing the installation key was not successful

Info	Topic
12.6.1	Data record TtSn

12.6.3 Delete individual installation key

With this data record, an installation key is deleted in a terminal. The response to the "TtDn" data record is given with the "StDn" record.

GID DID ON/OFF	Group address Device address Operating mode
T t	Record type Record type modification
D n	Identifier of the installation key

The installation key is selected based on the identification in the data record according to the following table:

Identifier

Entry	Meaning
D0	Delete sitekey 0
D1	Delete sitekey 1
...	...
D6	Delete sitekey 6
D7	Delete sitekey 7

Info	Topic
12.6.4	Data record StDn

12.6.4 Result of individual installation key deletion

The terminal sends the following data record in response to a TtDn record. The StDn record indicates whether the deletion of the installation key was successful or not.

GID DID ON/OFF	Group address Device address Operating mode
S t	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
D n	Identifier of the installation key (D0 to D7)
n	Result, see following table.

Result

Entry	Meaning
0	Deleting the installation key was successful
1	The installation key could not be deleted, or it does not exist

Info	Topic
12.6.3	Data record TtDn

12.6.5 Delete all installation keys

With this data record, all installation keys are deleted in a terminal.

GID DID ON/OFF	Group address Device address Operating mode
T t	Record type Record type modification
A 1	Delete all installation keys

12.6.6 Result of deleting all installation keys

The terminal sends the following data record in response to a TtA1 record. The StA1 record indicates whether the deletion of all installation keys was successful or not.

When performing a cold start of the terminal, the installation keys are also deleted, and this data record is sent.

GID DID ON/OFF	Group address Device address Operating mode
S t	Record type Record type modification, see following description.
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
A 1	Identifier for "Delete all installation keys"
n	Result, see following table.

Result

Entry	Meaning
0	Deleting all installation keys was successful
1	Deleting all installation keys was not successful

Info	Topic
12.6.5	TtA1 record.

12.7 Distributing installation keys via programming master A/B

This function is required if there are standalone components in an installation in addition to online terminals. The installation keys cannot be sent to them over an online connection.

On a programming master A or B, all installation keys presently being used in an installation are stored. By booking on a terminal or standalone component, the installation keys are transferred to the terminal or the standalone component.

If a programming master A or B is detected during booking, the installation keys of the programming master and those of the standalone component/terminal are synchronized. Hereby, the following applies:

- All installation keys of the programming master that do not exist or have been modified are transferred.
- Installation keys in the terminal/standalone component that are not stored in the programming master are deleted.
- Individual installation keys cannot be transferred later on to the terminal/standalone component by means of the programming master.

This process is logged in 2 data records of the terminal. This function can be enabled and disabled by means of data records (for online terminals). After a cold start of the terminal, this function is disabled, i.e. the installation keys **cannot** be transferred to the terminal by means of a programming master.

12.7.1 Disabling/enabling installation key transfer

The transfer of installation keys from a programming master to the terminal can be enabled or disabled by means of a data record. After a cold start of the terminal, all installation keys of the terminal are deleted. The function "Transfer installation key from programming master" is disabled after a cold start. If you want to use it, enable it previously by means of a data record.

12.7.2 Data record "Disabling/enabling installation key transfer"

The following data record enables/disables the function "Transfer installation key from programming master" in the addressed terminal.

GID	Group address
DID	Device address
ON/OFF	Operating mode
T	Record type
t	Record type modification
A	
n	Identifier, see following table

Identifier

Entry	Meaning
2	Disabling the installation key transfer from the programming master
3	Enabling the installation key transfer from the programming master

12.7.3 Distributing the installation key and report check sum

The terminal sends the following data record if you have tried to transfer one or several installation keys via the programming masters A or B to the terminal. The StB1 record indicates whether the transfer of the installation key(s) was successful or not.

With a positive result (result = 0), the following is transferred together with the data record:

- 20 digits UID of the programming master
- 128 digits check sum of the transferred installation key(s). For each installation key, there are 16 available digits (= 8 bytes) for the check sum.

If the transfer of the installation key(s) failed (results= 1, 2, or 3), the UID and the check sum are not transferred.

Neither is the data record B2, which follows in case of success, transferred in case of failure.

GID DID ON/OFF	Group address Device address Operating mode
S t	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
B 1	Identifier "Report check sum of installation key"
n	Result, see following table.
n ... n	20 digits UID
n ... n	128 digits check sum of the installation key

Result

Entry	Meaning
0	Transfer of installation key successful
1	Transfer of installation key not successful
2	Transfer of installation key disabled
3	No transfer of installation key because a valid check sum has not been found.

12.7.4 Distributing the installation key and report parent info

The terminal sends the following data record after the B1 record if you have transferred one or several installation keys via the programming masters A or B to the terminal. With a positive result (result = 0), the following is transferred together with the data record:

- 20 digits UID of the programming master
- 128 digits parent info sum of the transferred installation key(s). For each installation key, there are 16 available digits (= 8 bytes) for the parent info.

If the transfer of the installation key(s) failed (results= 1, 2, or 3), this data record is not transferred.

GID DID ON/OFF	Group address Device address Operating mode
S t	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
B 2	Identifier "Report parent info of installation key"
n	Result, 0 = Transfer of installation key successful
n ... n	20 digits UID
n ... n	128 digits parent Info

12.7.5 Exchanging production key against application key

External companies receive a so-called production key for writing on media. It is used for programming the media in the first place. For the daily use at the end customer's, the production key is then exchanged against the application key.

Each application on a medium has its own production key that is replaced by its individual application key during the exchange. The exchange can also be made on a terminal.

The function "Exchange production key against installation key" is disabled after a cold start. If you want to use it, enable it by means of a data record.

GID	Group address
DID	Device address
ON/OFF	Operating mode
T	Record type
t	Record type modification
A	
n	Identifier, see following table

Identifier

Entry	Meaning
4	Disable exchanging production key against application key.
5	Enable exchanging production key against application key.

12.7.6 Report production key exchange

The following data record is sent after exchanging the production key against the application key on a dormakaba Mifare medium for an application. It contains:

- Contents of the medium according to the parameterization of the A02 record.
- UID of the dormakaba Mifare medium.
- Application designation for which the production key has been exchanged. Possible value only: IDENTIFICATION

GID DID ON/OFF	Group address Device address Operating mode
f $	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
0	Reserved 0
n ... n	Media content according to the parameter setting variable length.
n ... n	20 digits UID
n ... n	14 digits application description Possible value only: IDENTIFICATION

12.8 Report terminal status

The system statuses 1-8 are explained in chapter. In this chapter, the specific dormakaba Mifare system status 9 is explained.

GID DID ON/OFF	Group address Device address Operating mode
S 7	Record type Record type modification
CI	Clock identifier
Y Y M M D D	Date indicator; Year-year/month-month/day-day
h h m m (s) (s)	Time Seconds only if set as parameter.
n	SYSTAT9

8	4	2	1	Hex value

3	2	1	0	Bit

Key transfer from programming master permitted
0 = No
1 = Yes
Production key against application key exchange enabled
0 = No
1 = Yes
Reserved
Reserved

13 Tables

13.1 Tables of record types/record type modifications (RT/RTM)

13.1.1 RT/RTM for download data records

Functional group	RT/RTM	Meaning	Chapter
Display records	D0	Static display	[▶ 5.1]
	D1	Dynamic display authorized	
	D2	Dynamic display not authorized	
Dialog records	G0	Dialog mask	[▶ 7.7.5]
	G9	End of dialog	[▶ 7.7.7]
Dialog records CardLink	G2	Writing CardLink data	[▶ 10.7.1]
	G4	MediaTraceback request data	[▶ 10.8.1]
	G6	MediaTraceback delete data	[▶ 10.8.4]
Data record CardLink	>3	+identifier fnn: CardLink validation via function key	[▶ 10.4]
	Tt	+identifier A8: Disable creation of actuator status file	[▶ 10.9.1]
	Tt	+identifier A9: Enable creation of actuator status file	[▶ 10.9.1]
	v0	CardLink validation record	[▶ 10.5.1]
	v1	CardLink validation deletion record	[▶ 10.5.2]
	v8	CardLink validation request record	[▶ 10.5.3]
	v9	CardLink validation end record	[▶ 10.5.5]
	V0	CardLink update record	[▶ 10.6.1]
	V1	CardLink update deletion record	[▶ 10.6.2]
	V8	CardLink update request record	[▶ 10.6.4]
	V9	CardLink update end record	[▶ 10.6.3]
Controlling the terminal	I0	Release bookings	[▶ 5.2.1]
	I1	Block bookings	[▶ 5.2.1]
	I2	Enable local parameter setting	[▶ 5.2.1]
	I3	Block local parameter setting	[▶ 5.2.1]
	I6	Enable dialog	[▶ 5.2.1]
	I7	Block dialog	[▶ 5.2.1]
	J0	Activate relay statically	[▶ 5.2.2]
	J1	Activate relay dynamically	[▶ 5.2.2]
	J2	Reset relay	[▶ 5.2.2]

Functional group	RT/RTM	Meaning	Chapter
Controlling the terminal	T0	Switch terminal to "Online" mode	[▶ 5.2.3]
	T1	Switch terminal to "Offline" mode	[▶ 5.2.3]
	T2	Switch terminal to "Autonomous" mode	[▶ 5.2.3]
	T3	Request recorded data in autonomous mode	[▶ 5.2.3]
	T4	Request terminal status	[▶ 5.2.3]
	T8	Request program number	[▶ 5.2.3]
	TI	Request terminal info	[▶ 5.2.4]
	TV	+ identifier s: Set terminal volume	[▶ 5.2.7]
	TV	+ identifier g: Request terminal volume	[▶ 5.2.8]
	Tt	+ identifier B0: Request backup status	[▶ 5.2.5]
	Tt	+ identifier B1: Delete Backup status	[▶ 5.2.6]
	Tt	+ identifier Tn: Request reader information	[▶ 5.2.9]
	Tt	+ identifier nn: Request launch data	[▶ 5.2.10]
Data records for use with ARIOS	Tt	+ identifier A1: deleting all installation keys	[▶ 12.6.5]
	Tt	+ identifier A2: disabling installation key transfer	[▶ 12.7.2]
	Tt	+ identifier A3: enabling installation key transfer	[▶ 12.7.2]
	Tt	+ identifier A4: disabling installation key exchange	[▶ 12.7.5]
	Tt	+ identifier A5: enabling installation key exchange	[▶ 12.7.5]
	Tt	+ identifier Sn: installation key download	[▶ 12.6.1]
	Tt	+ identifier Dn: deleting single installation key	[▶ 12.6.3]
Control buffer memory	M0 M1 M2	Read buffer memory Delete buffer memory Initialize buffer memory	[▶ 5.2.11]

Functional group	RT/RTM	Meaning	Chapter
Logical booking response	R0	Silent confirmation	[▶ 5.3]
	R1	Booking authorized	[▶ 5.3]
	R2	Booking not authorized	[▶ 5.3]
	R3	Repeat input	[▶ 5.3]
	R4	Booking authorized with 8 characters display info	[▶ 5.3.1]
	R5	Booking authorized with 13 characters display info	[▶ 5.3.1]
	R6	Booking authorized with 20 characters display info	[▶ 5.3.1]
	R9	Prolongation of response timeout	[▶ 5.3.2]
	RU	Booking authorized with individual user information	[▶ 5.3]
	RV	Booking not authorized with individual user information	[▶ 5.3]
Date/time	U0	Set date/time without seconds	[▶ 5.4]
	U1	Set date/time with seconds	[▶ 5.4]
Master records	Y0	Individual master record download	[▶ 5.5.1]
	Y0	Pseudo master record download	[▶ 5.5.4]
	Y8	Request individual master record	[▶ 5.5.2]
	Y1	Delete individual master record	[▶ 5.5.2]
	Y8	Request pseudo master record	[▶ 5.5.5]
	Y1	Delete pseudo master record	[▶ 5.5.5]
	Y9	End of individual master record download	[▶ 5.5.3]
	Y9	End of pseudo master record download	[▶ 5.5.6]
Parameter download/ upload	>3	Parameter download record	[▶ 3.20]
	>8	Parameter upload request	[▶ 6.1]
	>9	End of parameter download	[▶ 6.3]

13.1.2 RT/RTM for upload data records

Functional group	RT/RTM	Meaning	Chapter
Recording data	A0	Access	[▶ 7.1]
	A1	Access (in)	
	A2	Access (out)	
	B0	Interrogation	
	B1	IN or IN with reason	
	B2	OUT or OUT with reason	
	B3	Business errand	
	B4	End of business errand	
	B5	Business errand OUT	
	B6	Correction	
	BE	Intermission end	
	BS	Intermission start	
	F0	Special function	
Dialog records	G1	Dialog input	[▶ 7.7.6]
	G8	Dialog request	[▶ 7.7.4]
	G9	End of dialog	[▶ 7.7.7]
Dialog records CardLink	G3	Dialog response	[▶ 10.7.2]
	G5	Report MediaTraceback data	[▶ 10.8.2]
	G7	Report result of deleting operation	[▶ 10.8.5]
Data records CardLink	v0	CardLink validation record upload	[▶ 10.5.4]
	v9	CardLink validation record upload end	[▶ 10.5.5]
	V0	CardLink update record upload	[▶ 10.6.5]
	V9	CardLink update record upload end	[▶ 10.6.3]
	nn	CardLink V1.0 data record validation LEGIC	[▶ 10.5.6]
	nn	CardLink V1.1 data record validation LEGIC	[▶ 10.5.7]
	n%	CardLink data record update	[▶ 10.6.6]
Error records	O1	Record content false	[▶ 7.2]
	O2	Command not executable at the moment	[▶ 7.2]
	O3	Command not executable	[▶ 7.2]
	O5	Missing license	[▶ 7.2]

Functional group	RT/RTM	Meaning	Chapter
Status records	S0	Buffer memory empty	[▶ 7.3.2]
	S2	Buffer memory full	[▶ 7.3.2]
	S7	Report terminal status	[▶ 7.3.3]
	SV	Report program number	[▶ 7.3.1]
	St	+ identifier B0: Report backup status	[▶ 7.3.4]
	St	+ identifier Tn: Report reader information	[▶ 7.3.5]
	St	+ identifier VR: Report result of VCP processing	[▶ 7.3.6]
	St	+ identifier nn: Report launch data	[▶ 7.3.7]
	TI	Report terminal info	[▶ 7.3.8]
	TV	Report terminal volume	[▶ 7.3.9]
Status records for ARIOS	S7	Report terminal status SYSTAT 9	[▶ 12.8]
	St	+ identifier A1: Result of deleting all installation keys	[▶ 12.6.6]
	St	+ identifier B1: Distributing the installation key and reporting the check sum	[▶ 12.7.3]
	St	+ identifier B2: Distributing the installation key and reporting the parent info	[▶ 12.7.4]
	St	+ identifier Dn: Result of single installation key deletion	[▶ 12.6.4]
	St	+ identifier Sn: Result of installation key download	[▶ 12.6.2]
	f$	Report exchange of production key	[▶ 12.7.6]
Alarm records	X1	Door open time too long	[▶ 7.4]
	X2	Door break open	[▶ 7.4]
	X3	Door closed again	[▶ 7.4]
	X4	Sabotage contact loop opened	[▶ 7.4]
	X5	Opened sabotage contact loop closed again	[▶ 7.4]
Service alarm records	XS	Service alarm records: Temperature alarm	[▶ 7.4.1]
Master records	Y0	Individual master record upload	[▶ 7.5.1]
	Y0	Pseudo master record upload	[▶ 7.5.2]
	Y9	End of individual master record upload	[▶ 7.5.3]
	Y9	End of pseudo master record upload	[▶ 7.5.3]
Parameter Up-/Download	>3	Parameter upload record	[▶ 7.6.1]
	>9	End of parameter upload	[▶ 7.6.2]

13.2 Table of parameter settings errors

Possible parameter setting errors during operation:

Error	Meaning
4	Less digits were read from the badge than set with the parameter group "Badge definitions".
5	A master record refers to a time profile which is not stored in the terminal.
7	The parameter "Individual master record required" defines that a master record must be assigned to each ID number read from the badge, but the parameter "Badge input required" does not define a badge input and/or the parameters of the parameter group "Badge definitions" do not define an ID number.
8	Fix text is larger than 8 digits.
9	Via the parameters of the parameter group "Badge definitions" a badge block with more than 10 digits and with the identification 7 or 8 has been defined (end/start of validity).
A	The number of characters in the recording data records exceeds 249 characters when using "Additional input steps".

Info	Topic
3.14	Badge definitions
3.10.4	Individual master record required
3.10.12	Badge input required

13.3 Table of operating modes

ON/OFF/AUTO (hex)	Meaning
Space character (20)	Operating mode Online
0 (30)	Operating mode Offline
@ (40)	Operating mode Autonomous

13.4 Table of clock identifiers

Identifier in data record	Time valid	Data record with seconds	Language identifier	Reason
0	No	No	No	Time not set
1	Yes	No	No	Time set without seconds
2	No	Yes	No	Time not set
3	Yes	Yes	No	Time set with seconds
4	No	No	Yes	Time not set
5	Yes	No	Yes	Time set without seconds
6	No	Yes	Yes	Time not set
7	Yes	Yes	Yes	Time set with seconds

13.5 Table of error identifiers

Error identifier	Meaning
0	No error
2	Customer no. error
3	Status error
4	No master record found
5	Master record blocked
6	No correction authorization
7	No official absence authorization
8	In/out error (plausibility error)
9	No time authorization (profile)
E	No badge number entry authorization
F	Verification not successful
:	PIN wrong or missing
;	Versions no. false
>	Time exceeded
@	Attempted double access

13.6 Table of block identifiers

Block identifier	Meaning	Chapter
A02	Badge definitions	[▶ 3.14.7]
A12	Conversion of badge data	[▶ 3.15.5]
B01 to B15	Mailbox texts	[▶ 3.19.4]
b01 to b15	Mailbox texts with language identification	[▶ 3.19.4]
D00 to D15	Dialog texts	[▶ 3.19.2]
d00 to d15	Dialog texts with language identification	[▶ 3.19.2]
E01	Standard/daylight saving time switchover and seconds in data record	[▶ 3.9.8.3]
F00 to F05 and F31 to F40	Function key functions	[▶ 3.10.34]
I01 to I10	Display info texts	[▶ 3.19.5]
i01 to i10	Display info with language identification	[▶ 3.19.5]
K01	Max. allowed door opening time and relay	[▶ 3.16.4]
L01 to L04	Special day table	[▶ 3.18.3]
M00 to M05 and M31 to M40	Function key texts	[▶ 3.19.1]
m00 to m05 and m31 to m40	Function key texts with language identification	[▶ 3.19.1]
N00 to N99	List texts	[▶ 3.19.6]
n00 to n99	List texts with language identification	[▶ 3.19.6]
P01 to PFE	Time profiles	[▶ 3.17.5]
PFF	Reserved	
PQ1	Biometric time profiles	[▶ 11.4.7]
S01	Data backup, fix record length for badge data/ keyboard/list data, logical group/device address, response timeout	[▶ 3.9.7]
T01 to T28	General texts	[▶ 3.19.3]
t01 to t28	General texts with language identification	[▶ 3.19.3]
W01 to W20	Lists	[▶ 3.11.5]
X01	Display time "authorized/not authorized", relay operation time, operator timeout and display time "display info"	[▶ 3.9.14]
X02	Customer number	[▶ 3.9.17]
X04	Code control byte	[▶ 3.9.20]
X07	Character set identification	[▶ 3.9.22]
X08	Additional customer number	[▶ 3.9.24]
Z01	Function key control	[▶ 3.12.3]
Z11 to Z13	Relay control	[▶ 3.13.4]
z00	.ini file settings	[▶ 3.20]

13.7 Table of graphics for user guidance

The supplied graphics for user guidance with the file extension "_Theme0x" in the [SurfaceDesign] section of the "interface.ini" file can be customized. One set of each of the graphics labeled

- <filename>_Theme01.png and
- <filename>_Theme02.png

in the filename (e.g. Ausweis_Theme01.png, Ausweis_Theme02.png) is already included in the delivery.

This means that graphics

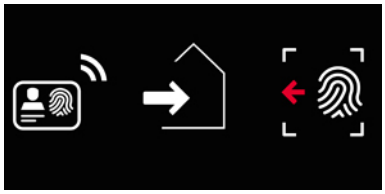
- in white lines for dark backgrounds (<filename>_Theme01.png) or
- in black lines for light backgrounds (<filename>_Theme02.png)

are ready for use.

The user can also define his/her own colorations and generate the graphics accordingly in the /data/data/com.kaba.apps.ba/files/init/images directory.
The graphics receive a theme extension of their own, e.g. <filename>_Theme3.png.

For display, the "Theme" parameter must be indicated accordingly in the [SurfaceDesign] section, e.g. Theme=3. If self-defined graphics can not be found by the system, the default graphic will be loaded automatically.

Example graphics for user guidance

By defining the parameter "Theme=01" in the [SurfaceDesign] section, the graphics are displayed in white lines. This is the default setting.	By defining the parameter "Theme=02" in the [SurfaceDesign] section, the graphics are displayed with dark lines.
	

13.8 Tables of language identifications

The language identification consists of 6 digits:

- digit 1 and 2 represent the language
- digit 3 and 4 represent the country
- digit 5 and 6 are not used at the moment and must be filled with blanks (20hex).

If no country identification is needed, the digits 3 and 4 may also be filled up with blanks.

For displaying texts in a certain language the use of the appropriate character set is necessary. The stated character set defines the format for date/time as well as the display format for default texts (if present). If there are no default texts for a language, these texts are output in English.

The tables below show the way the user information and texts are displayed.

If no character set has been chosen or character set 000 USA ASCII has been set (see chapter [▶ 13.9]), the 8-bit characters can be used. To use multilingualism, it is absolutely necessary.

Digit	Meaning
1 and 2	Language identification
3 and 4	Country identification
5 and 6	Not used, must be filled with blanks (20hex).

13.8.1 Country identifications

Country identification	Meaning
GB	Great Britain
US	USA

The differentiation between countries offers the benefit, for example that "en" can be separated for the countries "GB" and "USA". This allows to define different texts for the UK and USA, these texts can contain country-specific terms.

13.8.2 Language identification & character set ISO 8859-1

The language identifications listed here allow to display texts with the character set ISO 8859-1 (Latin 1). There are pre-defined default texts for languages marked with *.
If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
da	Danish*
de	German*
en	English*
es	Spanish*
fi	Finnish**
fr	French*
it	Italian*
nl	Dutch*
no	Norwegian*
sv	Swedish*

Character set Latin 1

CODE	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	not used															
1...																
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8...	not used															
9...																
A...	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
B...	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C...	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D...	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E...	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F...	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

13.8.3 Language identification & character set ISO 8859-2

The language identifications listed here allow to display texts with the character set ISO 8859-2 (Latin 2). There are pre-defined default texts for languages marked with *.
If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
cs	Czech*
hr	Croatian*
hu	Hungarian*
pl	Polish*
ro	Romanian
sk	Slovak*
sl	Slovenian

Character set Latin 2

CODE	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8...	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9...	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A...	NBSP	Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź	SHY	Ž	Ż
B...	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
C...	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
D...	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
E...	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
F...	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	˙

13.8.4 Language identification & character set ISO 8859-5

The language identifications listed here allow to display texts with the character set ISO 8859-5 (Cyrillic). There are pre-defined default texts for languages marked with *.
If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
bg	Bulgarian
be	Bielorussian
ru	Russian*
sr	Serbian
uk	Ukrainian

Character set Cyrillic

CODE	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8...	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9...	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A...	NBSP	Ё	Ђ	Ѓ	Є	S	I	Ї	J	Љ	Њ	Ћ	Ќ	SHY	Ў	Џ
B...	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C...	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
D...	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E...	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F...	№	ё	ђ	ѓ	є	s	i	ї	j	љ	њ	ћ	ќ	§	ў	џ

13.8.5 Language identification & character set ISO 8859-6

The language identifications listed here allow to display texts with the character set ISO 8859-6 (Arabic). There are pre-defined default texts for languages marked with *.
If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
ar	Arabic

Character set Arabic

Code	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8...	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9...	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A...	NBSP				¤								،	SHY		
B...												؛				؟
C...		ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د
D...	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
E...	-	ف	ق	ك	ل	م	ن	ه	و	ى	ي	ً	ٌ	ٍ	َ	ُ
F...	ِ	ّ	ْ													

13.8.6 Language identification & character set ISO 8859-7

The language identifications listed here allow to display texts with the character set ISO 8859-7 (Modern Greek). There are pre-defined default texts for languages marked with *. If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
el	Greek

Character set Modern Greek

CODE	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8...	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9...	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A...	NBSP	'	'	£	€		¦	§	¨	©	ͺ	«	¬	SHY		—
B...	°	±	²	³	΄	΅	Ά	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
C...	ΐ	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O
D...	π	P		Σ	T	Y	Φ	X	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
E...	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F...	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

13.8.7 Language identification & character set ISO 8859-9

The language identifications listed here allow to display texts with the character set ISO 8859-9 (Turkish). There are pre-defined default texts for languages marked with *.
If there are no default texts for a language, these texts are output in English.

Language identification	Meaning
tr	Turkish

Character set Turkish

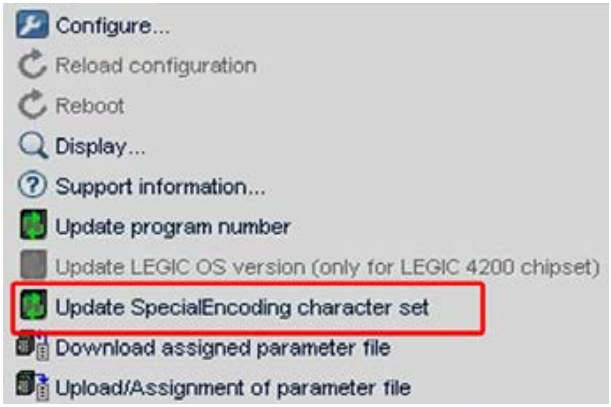
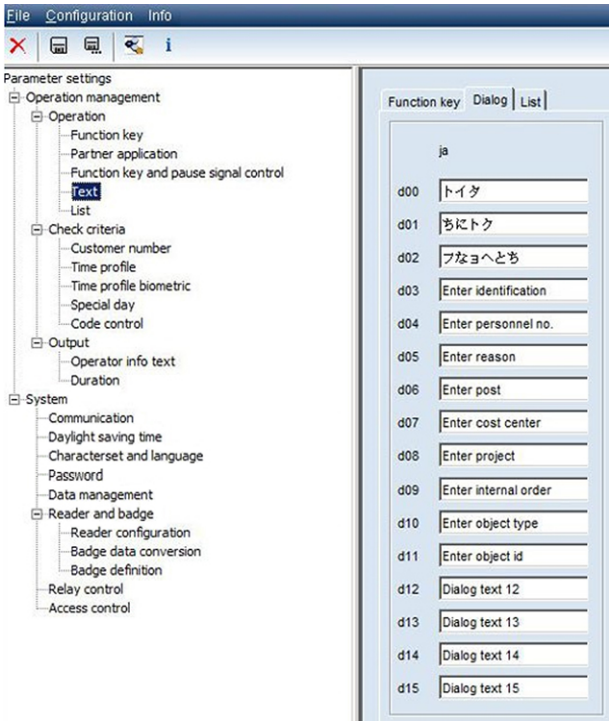
Code	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8...	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9...	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
A...	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
B...	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C...	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D...	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E...	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F...	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

13.8.8 Character set coding Japanese with SpecialEncoding

Creating a Japanese character set encoding (1)	
1.	Create a list of all the texts to be translated. • Make sure to translate only those texts that are actually needed. • A maximum of 128 characters may be used in the 8-bit range. • Only use characters that are supported by the font of the terminal (CJK characters). Note: The Japanese characters used in the example are no real translations, but only sample characters. Text — Characters used in the example with their Unicode Example text 1 — 30c8, 30a4, 30bf トイタ Example text 2 — 3061, 306b, 30c8, 30af ちにトク Example text 3 — 30d5, 306a, 30e7, 3078, 3068, 3061, 30d5 フなョへとちフ ….

Creating a Japanese character set encoding (2)	
	Character set table used. The marks indicate the characters used. ● Red: Characters for example text 1 ● Green: Characters for example text 2 ● Blue: Characters for example text 3 
2.	Use a Unicode editor (e.g. Notepad++) to create a table containing the character map location and the corresponding Unicode character. The character map location comprises the values of 0x80-0xFF. To determine the Unicode for a character in Windows, you can use, for example, the charmap.exe program (Start-Run-charmap.exe).

Creating a Japanese character set encoding (3)	
	Sample table with character map location and defined Unicode character. The character map table must start with the section name [Special...]. A declaration for Japanese could be as follows: [SpecialJapan] Char80=3061 Char81=3068 Char82=306a Char83=306b Char84=3078 Char85=30a4 Char86=30af Char87=30bf Char88=30c8 Char89=30d5 Char8A=30e7 Char8B= Char8C= ..... • The Char80-CharFF codes define the character map location. • The hexadecimal value defines the Unicode character.
3.	The file with the character map table must be saved in the terminal as "**encoding.ini**" in the path /data/data/com.kaba.apps.hr/files/init/encoding.ini.
4.	In the [RegionalSettings] section of the "system.ini" file (path: /data/data/com.kaba.apps.ba/files/init/system.ini), the name of this section must be specified as the parameter for CharsetEncoding =. In this example: [RegionalSettings] CharsetEncoding=SpecialJapan
5.	In the [Languages] section of the "interface.ini" file (path: /data/data/com.kaba.apps.hr/files/init/interface.ini), the corresponding language ID and country flag image must be specified. Example: [Languages] Language0=ja (for Japan) Language0Image=Japan.png
6.	If the order 3 to 5 has been kept, the application will restart automatically. Otherwise, the terminal will need to be restarted.

Creating a Japanese character set encoding (4)	
7.	Download the texts to be displayed with the language ID. • Start B-COMM (**from Version 3.10**) . • Select "Update SpecialEncoding character set" from the terminal's context menu (terminal - right mouse button). The command loads the SpecialEncoding character set from the "encoding.ini" file of the terminal and enters it into the channel configuration file. The character set stored in the terminal is thus made available in B-COMM. If the command is not available (grayed out), check whether the right character set (KabaSpecialEncoding) has been selected in the terminal settings ("Configure terminal" menu). 
8.	Select the PEC parameter editor in B-COMM. 
9.	Enter the desired text (d00-d02 in the example) and download the parameters.

13.8.8.1 Language identification & SpecialEncoding

The SpecialEncoding allows representing special characters using a custom Unicode reference table if these characters are supported by the language(s) installed on the terminal. These (8-bit) special characters can be used in addition to the 7-bit characters provided by ISO 8859-1.

The reference table is stored in a separate file named "encoding.ini", in the path /data/data/com.kaba.apps.hr/files/init/encoding.ini. A section named [Special<Name>] needs to be created here to allow assignment of the corresponding character set encoding.

The total number of characters (128) in this section must not be exceeded. The Unicode characters are defined for the 0x80-0xFF range; see example below.

The font (CJK characters) limits the scope of display in the terminal to the following ranges

- Simplified Chinese (GB2312)
- Traditional Chinese (Big 5)
- Japanese (JIS 0208)
- Korean (KSC 5601)

Not all Unicode characters can be displayed.

- The section created needs to be enabled in the /data/data/com.kaba.apps.ba/files/init/system.ini file. This is done using the CharsetEncoding=Special<Name> parameter in the [RegionalSettings] section.
- The corresponding language ID needs to be specified in the /data/data/com.kaba.apps.hr/files/init/interface.ini file. This is done using the LanguageX=<LanguageID> parameter in the [Languages] section. ("Language0=" to "Language4=" parameters).
- The corresponding country flag image needs to be specified in the /data/data/com.kaba.apps.hr/files/init/interface.ini file. This is done using the LanguageXImage=<FileName.png> parameter in the [Languages] section ("Language0Image=" to "Language4Image=" parameters).

Offline texts and user texts have to be displayed using the SpecialEncoding reference table and the 7-bit ISO 8859-nn character code or the 8-bit character code.

The following example illustrates the steps needed for creating a coded character set for the Japanese language.

13.8.8.2 Quick guide for character set encoding via SpecialEncoding

1. Create a list of the translations needed.

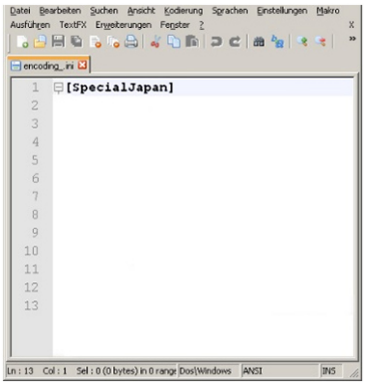
Example text1 トイタ

Example text2 ちにトク

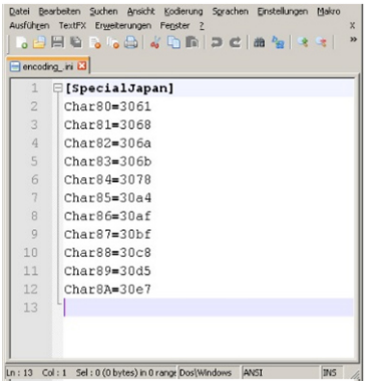
Example text3 フなョへとち

2. Collect the Unicode values for the characters used.

3. Create an "encoding.ini" file using an editor. The file must start with the section name [Special<Name>].



4. Enter the character map locations and the corresponding characters in the "encoding.ini" file. The value range for the character map locations covers the ranges Char80 to CharFF.



5. Save the "encoding.ini" file in the terminal in the path /data/data/com.kaba.apps.hr/files/init/encoding.ini .

6. Edit the following files in the terminal and save them again (see examples).

/data/data/com.kaba.apps.ba/files/init/system.ini

In the [RegionalSettings] section, assign the <SectionName> value from point 3 to the CharsetEncoding parameter.

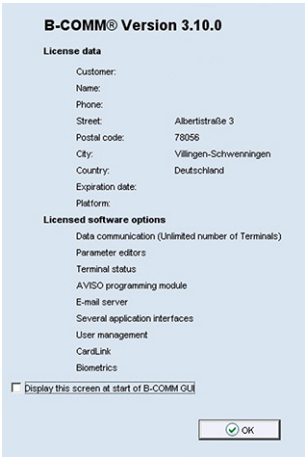
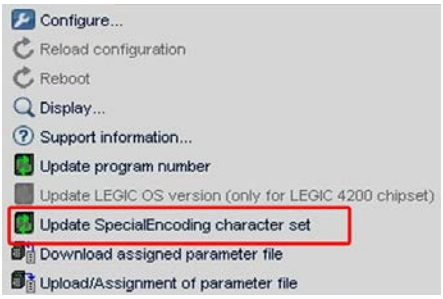
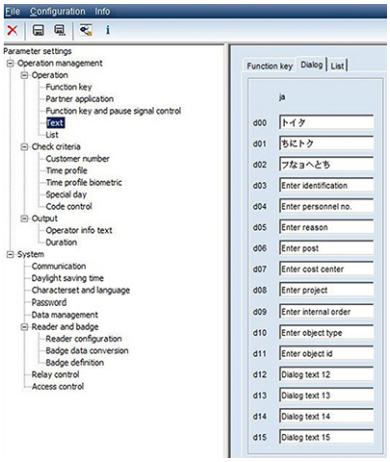
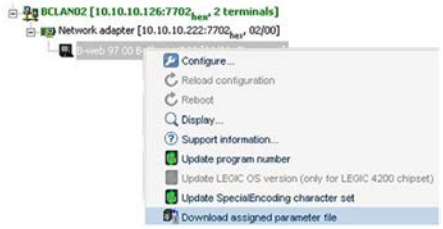
b) /data/data/com.kaba.apps.hr/files/init/interface.ini

In the [Languages] section, specify the language ID and the country flag image.

Consider the notation for the file name!

```
system.ini
[RegionalSettings]
CharsetEncoding=Special<Name>

interface.ini
[Languages]
Language0=ja
Language0Image=Japan.png
```

7. If the order 5 to 6b has been kept, the application will restart automatically. Otherwise, the terminal will need to be restarted. 		8. Start B-COMM (from Version 3.10). 
9. Select the "Update SpecialEncoding character set" to load the character set from the terminal into B-COMM. 		10. Select the PEC parameter editor for the terminal with the corresponding parameter file. 
11. Enter the desired texts. 		12. Load the parameter file into the terminal. 

13.9 Table of ISO 646 character sets

The character set identifier can be set as transmission of 7-bit characters. The conversion of characters during multilingual operation is performed by means of parameters listed in the "system.ini" file in the [RegionalSettings] section, see chapter [▶ 14.1.4] ff.

The tables below show the way the user information and texts are displayed in accordance with the selected character set.

If, for example, the ASCII character **#** is transmitted to the terminal and the character set **003** has been set, the character **£** is displayed.

Character set identifier	Meaning
000	Character set USA (ASCII)
001	Character set France
002	Character set Germany
003	Character set Great Britain
004	Character set Denmark I
005	Character set Sweden
006	Character set Italy
007	Character set Spain I
008	Character set Japan
009	Character set Norway
010	Character set Denmark II
011	Character set Spain II
012	Character set Latin America
099	Kaba Benzing character set (BT 930/Rockwell)

ASCII character (hex)	Character set identifier													
	000	001	002	003	004	005	006	007	008	009	010	011	012	099
Blank (20)														
! (21)	!	!	!	!	!	!	!	!	!	!	!	!	!	!
„ (22)	„	„	„	„	„	„	„	„	„	„	„	„	„	„
# (23)	#	#	#	£	#	#	#	Pt	#	#	#	#	#	#
$ (24)	$	$	$	$	$	¤	$	$	$	¤	$	$	$	$
% (25)	%	%	%	%	%	%	%	%	%	%	%	%	%	%
& (26)	&	&	&	&	&	&	&	&	&	&	&	&	&	&
' (27)	'	'	'	'	'	'	'	'	'	'	'	'	'	'
((28)	(	(	(	(	(	(	(	(	(	(	(	(	(	(
) (29)	)	)	)	)	)	)	)	)	)	)	)	)	)	)
* (2A)	*	*	*	*	*	*	*	*	*	*	*	*	*	*
+ (2B)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
, (2C)	,	,	,	,	,	,	,	,	,	,	,	,	,	,
- (2D)	-	-	-	-	-	-	-	-	-	-	-	-	-	-
. (2E)	.	.	.	.	.	.	.	.	.	.	.	.	.	.
/ (2F)	/	/	/	/	/	/	/	/	/	/	/	/	/	/
0 (30)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1 (31)	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2 (32)	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3 (33)	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4 (34)	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5 (35)	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6 (36)	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7 (37)	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8 (38)	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9 (39)	9	9	9	9	9	9	9	9	9	9	9	9	9	9
: (3A)	:	:	:	:	:	:	:	:	:	:	:	:	:	:
; (3B)	;	;	;	;	;	;	;	;	;	;	;	;	;	;
< (3C)	<	<	<	<	<	<	<	<	<	<	<	<	<	<
= (3D)	=	=	=	=	=	=	=	=	=	=	=	=	=	=
> (3E)	>	>	>	>	>	>	>	>	>	>	>	>	>	>
? (3F)	?	?	?	?	?	?	?	?	?	?	?	?	?	?

ASCII character (hex)	Character set identifier													
	000	001	002	003	004	005	006	0007	008	009	010	011	012	099
@ (40)	@	à	§	@	@	É	@	@	@	É	É	á	á	@
A (41)	A	A	A	A	A	A	A	A	A	A	A	A	A	A
B (42)	B	B	B	B	B	B	B	B	B	B	B	B	B	B
C (43)	C	C	C	C	C	C	C	C	C	C	C	C	C	C
D (44)	D	D	D	D	D	D	D	D	D	D	D	D	D	D
E (45)	E	E	E	E	E	E	E	E	E	E	E	E	E	E
F (46)	F	F	F	F	F	F	F	F	F	F	F	F	F	F
G (47)	G	G	G	G	G	G	G	G	G	G	G	G	G	G
H (48)	H	H	H	H	H	H	H	H	H	H	H	H	H	H
I (49)	I	I	I	I	I	I	I	I	I	I	I	I	I	I
J (4A)	J	J	J	J	J	J	J	J	J	J	J	J	J	J
K (4B)	K	K	K	K	K	K	K	K	K	K	K	K	K	K
L (4C)	L	L	L	L	L	L	L	L	L	L	L	L	L	L
M (4D)	M	M	M	M	M	M	M	M	M	M	M	M	M	M
N (4E)	N	N	N	N	N	N	N	N	N	N	N	N	N	N
O (4F)	O	O	O	O	O	O	O	O	O	O	O	O	O	O
P (50)	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Q (51)	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R (52)	R	R	R	R	R	R	R	R	R	R	R	R	R	R
S (53)	S	S	S	S	S	S	S	S	S	S	S	S	S	S
T (54)	T	T	T	T	T	T	T	T	T	T	T	T	T	T
U (55)	U	U	U	U	U	U	U	U	U	U	U	U	U	U
V (56)	V	V	V	V	V	V	V	V	V	V	V	V	V	V
W (57)	W	W	W	W	W	W	W	W	W	W	W	W	W	W
X (58)	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Y (59)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z (5A)	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
[(5B)	[	°	Ä	[	Æ	Ä	°	¡	[	Æ	Æ	¡	¡	[
\ (5C)	\	ç	Ö	\	0	Ö	\	Ñ	¥	0	0	Ñ	Ñ	\
] (5D)	]	§	Ü	]	Å	Å	é	¿	]	Å	Å	¿	¿	]
^ (5E)	^	^	^	^	^	Ü	^	^	^	Ü	Ü	é	é	↑
_ (5F)	_	_	_	_	_	_	_	_	_	_	_	_	_	↓

ASCII character (hex)	Character set identifier													
	000	001	002	003	004	005	006	007	008	009	010	011	012	099
‘ (60)	‘	‘	‘	‘	‘	é	ù	‘	‘	é	é	‘	ü	‘
a (61)	a	a	a	a	a	a	a	a	a	a	a	a	a	a
b (62)	b	b	b	b	b	b	b	b	b	b	b	b	b	b
c (63)	c	c	c	c	c	c	c	c	c	c	c	c	c	c
d (64)	d	d	d	d	d	d	d	d	d	d	d	d	d	d
e (65)	e	e	e	e	e	e	e	e	e	e	e	e	e	e
f (66)	f	f	f	f	f	f	f	f	f	f	f	f	f	f
g (67)	g	g	g	g	g	g	g	g	g	g	g	g	g	g
h (68)	h	h	h	h	h	h	h	h	h	h	h	h	h	h
i (69)	i	i	i	i	i	i	i	i	i	i	i	i	i	i
j (6A)	j	j	j	j	j	j	j	j	j	j	j	j	j	j
k (6B)	k	k	k	k	k	k	k	k	k	k	k	k	k	k
l (6C)	l	l	l	l	l	l	l	l	l	l	l	l	l	l
m (6D)	m	m	m	m	m	m	m	m	m	m	m	m	m	m
n (6E)	n	n	n	n	n	n	n	n	n	n	n	n	n	n
o (6F)	o	o	o	o	o	o	o	o	o	o	o	o	o	o
p (70)	p	p	p	p	p	p	p	p	p	p	p	p	p	p
q (71)	q	q	q	q	q	q	q	q	q	q	q	q	q	q
r (72)	r	r	r	r	r	r	r	r	r	r	r	r	r	r
s (73)	s	s	s	s	s	s	s	s	s	s	s	s	s	s
t (74)	t	t	t	t	t	t	t	t	t	t	t	t	t	t
u (75)	u	u	u	u	u	u	u	u	u	u	u	u	u	u
v (76)	v	v	v	v	v	v	v	v	v	v	v	v	v	v
w (77)	w	w	w	w	w	w	w	w	w	w	w	w	w	w
x (78)	x	x	x	x	x	x	x	x	x	x	x	x	x	x
y (79)	y	y	y	y	y	y	y	y	y	y	y	y	y	y
z (7A)	z	z	z	z	z	z	z	z	z	z	z	z	z	z
{ (7B)	{	é	ä	{	æ	ä	à	¨	{	æ	æ	í	í	Ä
| (7C)	|	ù	ö	|	0	ö	ò	ñ	|	0	0	ñ	ñ	Ö
} (7D)	}	è	ü	}	å	å	è	}	}	å	å	ó	ó	Ü
~ (7E)	~	¨	ß	~	~	ü	ì	~	~	ü	ü	ú	ú	Å

13.10 ASCII table

	Control		ASCII						Control		Latin 1					
	000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
0	NUL	DLE	Space	0	@	P	`	p	CTRL	CTRL	Space	°	À	Ð	à	ð
1	SOH	DC1	!	1	A	Q	a	q	CTRL	CTRL	¡	±	Á	Ñ	á	ñ
2	STX	DC2	“	2	B	R	b	r	CTRL	CTRL	¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s	CTRL	CTRL	£	³	Ã	Ó	ã	ó
4	EQT	DC4	$	4	D	T	d	t	CTRL	CTRL	¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u	CTRL	CTRL	¥	µ	Å	Õ	å	õ
6	ACX	SYN	&	6	F	V	f	v	CTRL	CTRL	¦	¶	0	Ö	0	ö
7	BEL	ETB	’	7	G	W	g	w	CTRL	CTRL	§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x	CTRL	CTRL	¨	¸	È	0	è	0
9	HT	EM	)	9	I	Y	i	y	CTRL	CTRL	©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z	CTRL	CTRL	ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{	CTRL	CTRL	«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|	CTRL	CTRL	¬	¼	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}	CTRL	CTRL		½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~	CTRL	CTRL	®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL	CTRL	CTRL	¯	¿	Ï	ß	ï	ÿ

14 Configuration (.ini) –files

NOTICE

Total breakdown of the device due to misconfiguration of the .ini files.
The device is no longer bootable, usable or shows incorrect or instable behavior.

- Experience on how to use .ini files is mandatory.
- Customizing is only permitted in the .ini files intended for this purpose.
- dormakaba recommends to create backup files of the .ini files before making any customizations.
- Define only parameter values that are logical and reasonable.

The .ini-files in the terminal are all structured according to the same scheme. An .ini-file contains several sections representing a parameter group. These sections contain the actual parameters.

A section is characterized by square brackets, e.g. [Programs].

After the section name, the parameters are listed, e.g. MainClass

The parameter name is terminated with the character "=".

Then the parameter value follows, e. g "1" or "true".

Example:

[MasterRecordDefinition]

IDLength=7 (sets the length of the badge number to 7 digits).

Comments are marked with "//" at the beginning of a line.

If the content of an .ini file changes, the file will be read again after approx. 20 seconds and the new values of the parameters are taken into account.

Exception

The parameters of the application.ini file are considered after the next program start.

The parameters of the .ini files of the ../data/data/com.kaba.apps.ba/files/init directory are considered after the next program start.

The following chapters show the contents of the .ini files and their meaning.

In the chapter [▶ 15] you find more detailed notes about several parameters and their setting possibilities.

14.1 Files in the path /data/data/com.kaba.apps.ba/files/init

14.1.1 communication.ini

[Host]

The [Host] section contains parameters necessary for the communication with the host.

Parameter	Possible values	Meaning
IPAddress=	Valid IP address. Default: 123.0.0.1	The parameters **IPAddress** (a correct IP address) and **PortNumber** (BETA System ID) describe the host communication with the terminal.
PortNumber=	Valid port number. Default: 0x7700	For the port number the following values are allowed: 0x7700 to 0x77EF (hex) or 30464 to 30703 (dec).
EncryptionEnabled=	true false (default)	This parameter allows encryption of the communication.
Type=	UDP XML	The parameter defines the type of host communication.
UsingProxy=	true false	The parameter defines, whether a proxy shall be used for the HTTP communication or not. Basically, the proxy settings of the system are used but it is also possible to define alternative proxy settings in the [Proxy] section. These settings will be overtaken in the system settings when starting the B-Client HR.
httpServer=	hostIP:port/directory or hostname:port/directory **Example 1:** httpServer=123.0.0.1:8080 **Example 2:** httpServer=kbss200.dom3.kaba.grp:8500:Hostsimulation	The parameter configures the HTTP/HTTPS server. The definition of the directory is optional. If no definitions are made for the directory, the requested files are searched for in the root directory of the server.

[Proxy]

The [Proxy] section describes the parameter necessary for the communication with a proxy server.

Parameter	Possible values	Meaning
IPAddress=	Valid IP address Default: no address	The settings for the communication with a proxy server are to be requested at your network administrator.
PortNumber=	Valid port number. Default: 0	

[Client]

The [Client] section is only important if a device is equipped with more than one network adapter.

Parameter	Possible values	Meaning
IPAddress=	Valid IP address.	The **IPAddress** parameter defines the IP-address of the network adapter used for the communication of the device.

[FTCS]

The [FTCS] section defines the connection parameters to the FTCS (finger template control service) that is used with biometrics.

Parameter	Possible values	Meaning
IPAddress=	Valid IP address.	The **IPAddress** parameter defines the IP address of the FTCS.
PortNumber=	Valid port number.	The **PortNumber** parameter defines the port number of the FTCS.

14.1.2 terminal.ini

[Proximity]

The [Proximity] section describes parameters for controlling the brightness of the display, in conjunction with the proximity sensor.

Parameter	Possible values	Meaning
Brightness=	30 – 100 Default: 70	This parameter defines the basic brightness of the display. Details are quoted in %. 100% signifies maximum brightness.
BrightnessReducedTimeout=	0 (switch off) 30 – 3,600 Default: 120	The parameter defines the time in seconds after which the display is switched to reduced brightness. This time sequence is initiated by the proximity sensor. The value "0" switches off this function, i.e. the display **never** switches into reduced brightness mode.
BrightnessReduced=	10 – 100 Default: 20	This parameter defines the reduced brightness of the display. Details are quoted in %.
StandbyTimeout=	0 (switch off) 60 – 14,400 Default: 0	This parameter defines the time in seconds after which the display is switched into standby mode. This time sequence is initiated by the proximity sensor. The value "0" switches off this function, i.e. the system **never** switches into standby mode. Whenever the values for **BrightnessReducedTimeout** overlap with **StandbyTimeout** (StandbyTimeout less than or equal to BrightnessReducedTimeout), a transition time of 30 seconds is specified by the system. See sketch below.
ReaderIlluminationFrequency=	1,000 – 5,000 Default: 1,500	The parameter defines the flashing frequency of the illuminated ring. Here, the set time in milliseconds is the "On" time of the illuminated ring. The "Off" time of the illuminated ring is always double the "On" length of time.

Parametrization example with terminal 98 00-K7

- The basic brightness of the display is 70%
- 120 seconds after the last activity, brightness is reduced to 20%
- 150 seconds after the last activity, the standby mode is activated.
- The illuminated ring flashes with a default setting of 1,500 ms "On", 3,000 ms "Off".

14.1.3 debug.ini

The file debug.ini is for diagnosis use only. Do not change any settings file without contacting dormakaba. The diagnosis results will be logged in a file. An additional output on the console is possible.

[File]

The parameters of the [File] section define the output file.

Parameter	Possible values	Meaning
Length=	Default: 100,000	The **Length** parameter defines the number of characters that can be stored in the file with the name defined with the **Name** parameter. If the parameter maximum size is reached, the file is renamed to <**BackupName**>, see following parameter. If the file length is set to 0, the diagnosis is deactivated.
Name=	diagnosis/debug.log	
BackupName=	diagnosis/debug.bak	

[Special]

The parameters in the [Special] section describe all classes and objects to be logged.

Parameter	Possible values	Meaning
SaveInFile=	0 (= no) 1 (= yes)	The **SaveInFile** parameter defines whether diagnosis data shall be stored in the defined file (0: no; 1: yes).
ShowOnConsole=	0 (= no) 1 (= yes)	The **ShowOnConsole** parameter defines whether diagnosis data shall be shown on console (0: no; 1: yes).
ClassWithLevel0=	Example: Class1 Default: no class	The **ClassWithLevelX** (**ObjectWithLevelX**) parameters define single classes (objects) that shall be tested. The first parameter defines the class (without package identification), the second one the level of diagnosis: Value 9: all data Value 0: only important data are logged. Diagnosis data with level 0 are always logged, independently from the settings.
ClassWithLevel1=		
ObjectWithLevel0=	Example: Object1 Default: no object	
ObjectWithLevel1=		

[Global]

The parameters of the [Global] section are valid for all classes not named in the Special section.

Parameter	Possible values	Meaning
SaveInFile=	0 (= no, default) 1 (= yes)	The **SaveInFile** parameter defines whether diagnosis data shall be stored in the defined file (0: no; 1: yes). The ShowOnConsole parameter defines whether diagnosis data shall be shown on console (0: no; 1: yes). The **ShowOnConsole** parameter defines whether diagnosis data shall be shown on console (0: no; 1: yes). The **Level** parameter defines the volume of the diagnosis data: Value 9: all data Value 0: only important data are logged. Diagnosis data with level 0 are always logged, independently from the settings.
ShowOnConsole=	0 (= no, default) 1 (= yes)	
Level=	0 to 9 Default: 0	

14.1.4 system.ini

The system.ini file contains basic system parameters.

[Device]

The parameters in the [Device] section describe the device.

Parameter	Possible values	Meaning
Type=	ONE-K7 9600-K7 9700-K7 9800-K7	The **Type** parameter indicates the device type.

[Maintenance]

The [Maintenance] section describes all properties relevant for service personal.

Parameter	Possible values	Meaning
Language=	en (English, default) de (German)	The parameter **Language** defines the service language (menus, dialogs etc.).

[RegionalSettings]

The [RegionalSettings] section defines the language in the compatibility mode.

Parameter	Possible values	Meaning
Language=	en (default) Possible identifications are listed in the table of language identifications, see chapter.	The parameter **Language** defines the language in the compatibility mode.
CharsetEncoding=	ISO-8859-1 (default) ISO-8859-2 ISO-8859-5 ISO-8859-6 ISO-8859-7 ISO-8859-9 Special<Section name>	**CharsetEncoding** defines the preset ISO character set. The entry Special<Section name> refers to a freely definable character set encoding. A detailed description of the SpecialEncoding can be found in chapter [▶ 13.8.8].

[Reader1] und [Reader2]

The [Reader1] or [Reader2] section contains all parameters used to adapt the readers to the terminal.

Parameter	Possible values	Meaning
SerialType=	NONE LEGIC MIFARE BIOMETRIC_CBM BARCODE	The parameter **SerialType** defines the type of a serial reader. The type must match the installed hardware.
SerialInterface=	Depending on reader type	The **SerialInterface** parameters are used to adjust the reader interface. Allowed interface values: Baud rate 1,200 to 115,200 Baud Number of data bits 7, 8 Parity 0 = None 1 = Odd 2 = Even Number of stop bits 1, 2 The parameters are divided by blanks; missed parameters are filled up with default values.
SerialGuardTime=	2,000 (default)	**SerialGuardTime** defines the time (in milliseconds) that must be passed between two reader inputs with the same data. For mobile access, this parameter is used to realize a double access blocking.
SerialTrailer=	0x0D 0x0A (default)	With the **SerialTrailer** parameter, the trailer of the data can be defined (for serial reader type "Transparent" only). No more than 5 characters, divided by blanks, are possible. The characters can be defined by hex-values (0x value) or decimal values.
SerialHeaderLength=	0 (default)	The parameter **SerialHeaderLength** defines the number of leading characters that shall be cut (for serial reader type "Transparent" only).

[Reader1CBM]

The [Reader1CBM] section describes all settings valid for the biometric CBM reader module.

Parameter	Possible values	Meaning
PresetEnroll=	String of 15 valid characters max. Default: empty string	The **PresetEnroll** parameter defines a character string which is displayed during the enrollment process in the "Template ID" field. When typing in the template ID, the character string will be overwritten, beginning from the right. **Example:** Template ID input: 12345 Character string: 0000000 Template ID after input: 0012345 The max length of the character string is 15 digits . Valid characters: 1-9, A-F, and the special characters: ; < = > ?
PresetBooking=	Character string with valid characters. Default: empty string Factory setting: 000000000020000000000000	The **PresetBooking** parameter extends the template ID or the template ID entered via the keypad by a preset value. The preset value will be overwritten with the template ID, beginning from the right. When entering the template ID manually, the entry of leading zeros can be omitted. The factory settings generate a LEGIC compliant booking record. **Example:** Template-ID: 12345 Preset value: 000000000020000000000000 Badge data: 000000000020000000012345 The length of the preset value is not limited. Valid characters: 1-9, A-F, and the special characters: ; < = > ?
Standalone=	0 (default) 1	The **Standalone** parameter defines, whether standalone operation without connection to the FTCS is possible (=1) or not (=0). This parameter may be defined if necessary. If an activated standalone operation is deactivated again, all local enrolled finger prints will be deleted.
BiometricMode=	 1 2 3 4 5	Sets the biometric operating mode 1 = Identification 2 = Verification via template comparison 3 = Verification via ID comparison 4 = Mixed operation (mode 1 and 2) 5 = Mixed operation (mode 2 and 3)

Parameter	Possible values	Meaning
ProximityScale=	0 1 2 (default) 3	The **ProximityScale** parameter sets the trigger level of the proximity sensor. The sensor will activate the finger print reader module CBM when approaching a finger to the device. This parameter may be created if necessary to compensate interferences or disorders caused by environmental conditions. 0 = Deactivated. (CBM activation possible by pressing any function key). 1 = Low sensitivity 2 = Medium sensitivity (default) 3 = High sensitivity.
TemplateFormatEnrollment=	0 (default) 2	Defining the template format for local enrollment. 0 = PK_COMP V2 2 = PK_MAT
NoCheckOnTemplate	0 (default) 1	The parameter controls the check for duplicate templates in the template database of the CBM reader when enrolling new templates. It should be noted that the parameter has a negative logic, i.e. **no check** for duplicate templates is performed if the parameter is **1.** The check is **carried out** if the parameter is **0** (= default). The parameter may only be changed after consultation with dormakaba.
RFIDReader=	Reader2 (default) None	The **RFIDReader** parameter defines whether an RFID reader is to be used as an alternative reader for ID input, this is default setting or if it is used as a second reader (parameter value =None). This parameter must be defined if necessary.

The following sections

- [Reader1Special2]
- [Reader1Special2DataDefinition1] to [Reader1Special2DataDefinition9]

are used by dormakaba for customizing purposes.

Settings made here may not be changed without requesting dormakaba.

[Reader1Special2]

The [Reader1Special2] section is used for customizing an FLI-2 alien reader interface.

[Reader1Special2DataDefinition1] to [Reader1Special2DataDefinition9]

The [Reader1Special2DataDefinition1] to [Reader1Special2DataDefinition9] sections are used for customizing an FLI-2 alien reader interface.

14.1.5 images

The "images" directory contains all graphics to be used for a dynamic table.

More detailed information on the design of dynamic lists can be found in chapter [▶ 7.7.8].

The graphics that are stored here in this .ini file are referred to in item 6 of the named chapter.

Example:

```
Button_1.png

Button_2.png

Button_3.png
```

14.1.6 mediaact.ini

In the mediaact.ini file, the priority parameter defines the media definition(s) to be used. The media definition(s) is/are defined in the file mediadef.ini. Additionally, the priority parameter determines the succession, the media definition(s) shall be applied.

A detailed description of these files is given in the "Kaba LEGIC advant 4200" manual".

[LegicStructureRelease]

This section contains parameters relating the use and succession of the media definitions.

Parameter	Possible values	Meaning
Priority=	1 to 8 Default: 1 Example: Priority=1 3 4	This parameter establishes which media definitions from the mediadef.ini file are to be applied. Several media definitions can be established in any desired sequence. The parameter values are the number of the created sections [LegicStructure1] to [LegicStructure8] from the mediadef.ini file. The parameter values are separated by blanks.

14.1.7 mediadef.ini

[LegicStructure1] to [LegicStructure8]

These sections contain the parameters of the media definition. Max. 8 media definitions may be created in the relating [LegicStructure1] to [LegicStructure8] sections.

For ease of use, the mediadef.ini file already contains 10 ready-made media definitions ([LegicStructure1] to [LegicStructure8], [LegicStructureX] and [LegicStructureY]), see also the following examples for the default media definitions [LegicStructure1] to [LegicStructure3].

The default media definitions are designed for use with the default badge definition (A02 data record) and can read the data carriers encoded in this way. Please note that all media definitions supplied may have to be adapted to customer-specific badge definitions (e.g. with regard to the IDDataLength, CLFindstring, CLAddress and other parameters).

If the media definition [LegicStructureX] and/or [LegicStructureY] is to be used, the X/Y in the sections name must be replaced by a number1-8. The existing media definition(s) with this number(s) must be renamed.

A detailed description of these files is given in the "Kaba LEGIC advant 4200" manual".

Parameter	Possible values	Meaning
HFMode=	0 1 2 3	This parameter defines which media technology is to be used. The following parameter values can be set: 0 = LEGIC RF Standard (LEGIC Prime) 1 = ISO 15693 2 = 14443 A 3 = 14443 B
IDType=	UID UID_S ID CID ACC	This parameter defines the data that is to be used for the identification process. The identification data read by the medium are always preceded by the Structure Pattern Index (SPI, 1 byte). The SPI indicates the media definition that was used for the successful identification process. **Example:** If the identification process was successful with the media definition 3, the raw data are preceded by the SPI 3.

Parameter	Possible values	Meaning
IDType=	UID UID_S ID CID ACC	**UID** This setting is used to read the UID of a medium. The read UID is always provided with a length of 10 bytes. For a UID that is really shorter, leading bytes (0x00) precede it instead of the missing places. **UID_S** Like UID. In addition, the LEGIC chip of the user medium is also selected. The UID will only be accepted after successful selection (so-called safe UID). **ID** This setting is used to read a LEGIC data segment (stamp and contents of the data segment). This additionally requires the following parameters: • IDFindstring and StartSegment for the stamp • IDDataAddress to CRCAddress for the data. **CID** This setting is used to read - the card ID (10 bytes, shorter CIDs are filled with leading 0x00) - 4 bytes of the customer number from the data segment (If not available, 0x00 0x00 0x00 0x00 is entered.) - Data segment stamp The search criteria are the parameters • IDFindstring and • IDStartSegment. **ACC** This setting is used to read - the stamp and the - contents of a LEGIC access segment. This additionally requires the following parameters: • IDFindstring and IDStartSegment for the access segment stamp • IDDataAddress to IDCRCAddress for the access segment contents. The IDFindstring for LEGIC access always starts with 0x24 0x02

Parameter	Possible values	Meaning
IDFindstring= IDStartSegment=	depends on badge definition depends on badge definition	The following parameters IDFindstring to IDCRCAddress are required to select data from a data segment. The parameters must be adapted to the reader technology and to the data structure on the medium. IDFindstring The IDFindstring is the find stamp of the segment. It always begins with the SSC. IDStartSegment The IDStartSegment indicates the number of the segment in which the search following the IDFindstring should start.
IDDataAddress=	depends on badge definition	The **IDDataAddress** indicates at which address of the data segment the data readout should start.
IDDataLength=	depends on badge definition	The **IDDataLength** indicates how many places should be read out, starting with the address of the data segment.
IDCRCMode=	0 1 2	The **IDCRCMode** parameter specifies the CRC check method. The parameter can have the following values: 0 = no CRC check 1 = 8-Bit CRC check 2 = 16-Bit CRC check
IDCRCAddress=	depends on badge definition	The **IDCRCAddress** indicates at which address the checksum has been stored.
Key=	0	This parameter is used for extensions and not used at the moment.
IDPointer=	n n n n ...	The **IDPointer** parameter is used for flexible assembly of an ID from the read-out media data. To this end, up to 10 values can be given the parameter, on the basis of which the desired ID is assembled. The values indicate the position of the byte to be used. The individual values must be separated by blanks. **Example:** IDPointer=5 4 3 2 1 The counting of the reading position starts with 1 No definition means: no IDPointer to be used.
BlockXConvertionMode= BlockXConvertionFromDigit= BlockXConvertionLengthBefore= BlockXConvertionLengthAfter= BlockXConvertionMask=	depends on badge definition	The **BlockXConvertionMode** to **BlockXConvertionMask** parameters are described in detail in chapter [▶ 14.1.8].

Parameter	Possible values	Meaning
CLFindstring=	depends on badge definition	Findstring for CardLink
CLAddress=	depends on badge definition	Start address for CardLink
APPxFindstring=	depends on badge definition	Findstring for the application in cases the keyword in the segment designator is set to "APPLICATIONx", see chapter chapter. The identifier "x" at "APPLICATION" is a placeholder for a numbering.
ASFindstring=	depends on badge definition	Findstring for Actuator Status File
MTBFindstring=	depends on badge definition	Findstring for Media Traceback
BIOFindstring=	depends on badge definition	Findstring for Biometric (B-Client HR30 only).

Parameter to be created upon request

The following parameter is intended for mixed operation of LEGIC prime media using the CardLink function. In this case, a distinction must be made between addressing with the ProtocolHeaderAddress and addressing without the ProtocolHeaderAddress. The parameter can be set via a media definition.

Parameter	Possible values	Meaning
CLAddressType=	0 1	The parameter is only included for LEGIC prime media. It defines whether a CardLink address is interpreted from a data record (G2 record, V0 record or >3fxx record) or interpreted (=1) or not interpreted (=0) from a media definition as ProtocolHeaderAddress. For mixed operation, the parameter must have been added to the mediadef.ini file and set to 1.

14.1.7.1 Default media definition [LegicStructure1]

Copyable excerpt from the mediadef.ini file: [LegicStructure1]

Please note that the supplied media definition may have to be adapted to customer-specific ID card definitions (e.g. with regard to the IDDataLength, CLFindstring, CLAddress and other parameters).

```
//Legic prime KGH with SSC 02

[LegicStructure1]
HFMode = 0
IDType = ID
IDFindstring = 0x02
IDStartSegment = 1
IDDataAddress = 0
IDDataLength = 7
IDCRCMode = 1
IDCRCAddress = 7

Block0ConvertionMode = HexToASCII
Block0ConvertionFromDigit = 2
Block0ConvertionLengthBefore = 7
Block0ConvertionLengthAfter = 22
Block0ConvertionMask =

CLFindstring =
CLAddress =
CLAddressType=1
ASFindstring =
MTBFindstring =
BIOFindstring =
```

14.1.7.2 Default media definition [LegicStructure2]

Copyable excerpt from the mediadef.ini file: [LegicStructure2]

Please note that the supplied media definition may have to be adapted to customer-specific ID card definitions (e.g. with regard to the IDDataLength, CLFindstring, CLAddress and other parameters).

```
//ISO 14443A CID coding with SSC 03

[LegicStructure2]
HFMode = 2
IDType = CID
IDFindstring = 0x03
IDStartSegment = 1

//5 byte customer number, beginning with SSC
Block0ConvertionMode = HexToASCII
Block0ConvertionFromDigit = 16
Block0ConvertionLengthBefore = 5
Block0ConvertionLengthAfter = 16
Block0ConvertionMask =

//3 byte hex coded ID number
Block1ConvertionMode = HexToDecASCII
Block1ConvertionFromDigit = 9
Block1ConvertionLengthBefore = 3
Block1ConvertionLengthAfter = 6
Block1ConvertionMask =

CLFindstring =
CLAddress =
ASFindstring =
MTBFindstring =
BIOFindstring =
```

14.1.7.3 Default media definition [LegicStructure3]

Copyable excerpt from the mediadef.ini file: [LegicStructure3]

Please note that the supplied media definition may have to be adapted to customer-specific ID card definitions (e.g. with regard to the IDDataLength, CLFindstring, CLAddress and other parameters).

```
//ISO 14443A CID coding with SSC 06

[LegicStructure3]
HFMode = 2
IDType = CID
IDFindstring = 0x06
IDStartSegment = 1

//5 byte customer no. beginning with SSC
Block0ConvertionMode = HexToASCII
Block0ConvertionFromDigit = 16
Block0ConvertionLengthBefore = 5
Block0ConvertionLengthAfter = 16
Block0ConvertionMask =

//3 byte hex coded ID no.
Block1ConvertionMode = HexToDecASCII
Block1ConvertionFromDigit = 9
Block1ConvertionLengthBefore = 3
Block1ConvertionLengthAfter = 6
Block1ConvertionMask =

CLFindstring =
ASFindstring =
CLAddress =
MTBFindstring =
BIOFindstring =
```

14.1.8 Definition of conversion parameters

BlockXConvertionMode

This parameter defines the conversion mode. The following conversion modes are possible:

BlockXConvertionMode	Meaning
None	No conversion. **Example:** 0x367A > 0x36 0x7A >"6z" Range of values: 0x00 - 0xFF > 0x00 – 0xFF
HexToDecASCII	Converts hexadecimal coded bytes into decimal ASCII characters. **Example:** 0x125A > 4698(Dez) > 0x34 0x36 0x39 0x38 > "4698" Range of values: 0x0 - 0xF > 0x30 - 0x39 **The conversation of hex values is limited to 63 bit (7 byte + 7 bit).**
HexToASCII	Converts hexadecimal coded bytes into hexadecimal ASCII characters. **Example:** 0x125A > 0x31 0x32 0x35 0x41 >"125A" Range of values: 0x0 - 0xF > 0x30 - 0x39, 0x41 - 0x46
HexToSpecASCII	Converts hexadecimal coded bytes into special hexadecimal ASCII characters. **Example:** 0x125A > 0x31 0x32 0x35 -0x3A > "125:" Range of values: 0x0 – 0xF > 0x30- 0x3F
BCDToASCII	Converts BDC coded bytes into ASCII characters. **Example:** 0x3469 > 0x33 0x34 0x36 0x39 > "3469" Range of values: 0x0 - 0x9 > 0x30 - 0x39
LowNibbleHexToASCII	Converts the lower nibble of the badge raw data into hexadecimal ASCII characters. **Example:** 0x01>0x31>“1“ 0x02>0x32>“2“ 0x0A>0x41>“A“ Range of values: 0x0-0xF0 > x30-0x39, 0x41-0x46

BlockXConvertionFromDigit

The parameter defines the stating position of the conversion, first digit is position 1. The value "0" indicates an unused block. Negative values are not permitted.

BlockXConvertionLengthBefore

The parameter defines the number of bytes to be converted. Negative values are not permitted.

BlockXConvertionLengthAfter

The parameter defines how many bytes are existing after the conversion. If by conversion the data (number of bytes) becomes longer than specified here, converted data will be cut off beginning from the left (high byte). If the length of the converted data is less than indicated, the digits on the left hand side are filled up with ASCII characters "0" (30 hex).

BlockXConvertionMask

This parameter allows a bit manipulation of the bytes to be converted. To do so, a bit mask will be laid over these bytes. This way, a bitwise conversion can be realized.

Example:

Block0ConvertionMask = 0x0F 0xFF

If there are no bytes defined in the BlockXConvertionMask, the unmasked data bytes will be masked with 0xFF and then converted as specified in the BlockXConvertionMode parameter

14.1.9 mobileact.ini

This file defines which of the mobile structure definitions in the mobiledef.ini file are to be used. At the same time, the priority with which and order in which these mobile structure definitions are to be used also is defined.

[MobileStructureRelease]

This section contains parameters for the use and order of the mobile structure definitions.

Parameter	Possible values	Meaning
Priority=	10 or 11 Default: no value Example: Priority=10 11	The parameter indicates • which mobile structure definitions are to be used • and their order of use. The parameter values are separated by a space character.

14.1.10 mobiledef.ini

The mobile structure definitions are specified in this file. These include general information on the project and file IDs used, as well as the parameters for the technical implementation of transmission and any data conversion which may have to be performed.

[MobileGeneral]

This section contains general parameters for the use of MobileAccess.

Parameter	Possible values	Meaning
ProjectID=	4 bytes **Example:** ProjectID= 0x02 0x5D 0xB4 0xBA Default: no value	Hex-encoded customized LEGIC Connect project ID. This ID must match the ID stored in the app in order to perform a booking. This parameter has no default value assignment. It must not be possible to start any mobile communication without a **ProjectID**.
FileID=	Currently not used	--
InteractionMode=	0 (default) 1	The parameter determines whether the use of TapGo in connection with Mobile Access is enabled (1) or not (0). TapGo enabled means that the smartphone is treatened like an ID card in an access system. In the booking process, this means specifically, that the MobileApp for activation of access on the smartphone must be started once when entering the building. The app then remains active in the background. For bookings it is necessary only presenting the smartphone in front of the reader. In addition to the "InteractionMode" parameter, the parameter "BleOptionFilters=3" must be set in the [MobileStructure10] section, see below.

[MobileStructure10]

This section contains general parameters for the transmission technology Bluetooth Low Energy.

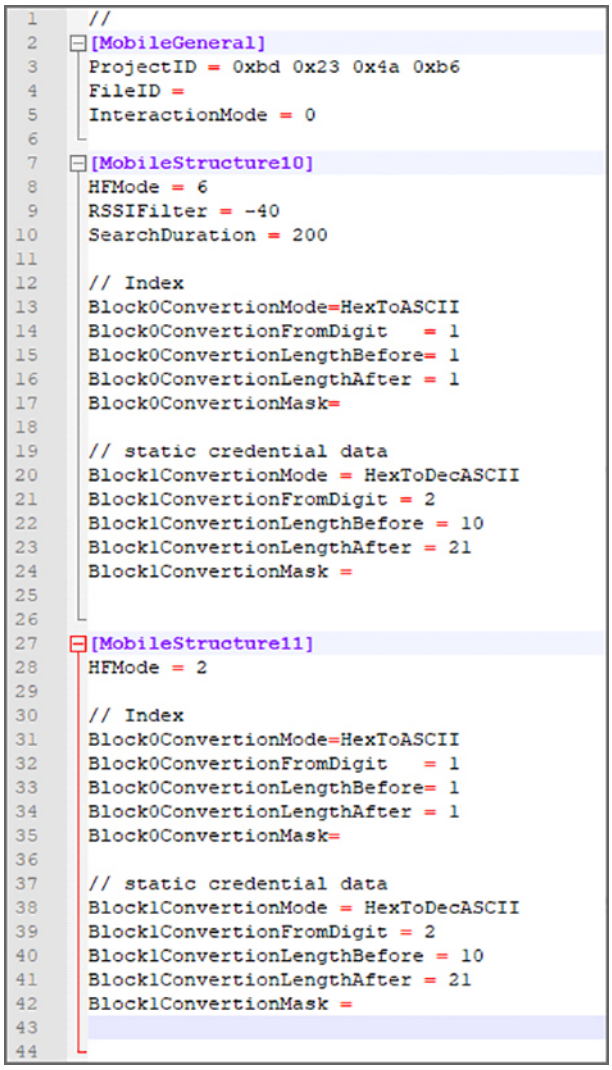
Parameter	Possible values	Meaning
HFMode=	6	The **HFMode** parameter defines the transmission technology to be used. 6= Bluetooth Low Energy.
BleOptionFilters=	3	The BleOptionFilters parameter is required for the use of TapGo and must be set to the value =3.
RSSIFilter=	-128 to -1 Default: -45 dBm for 9600 -40 dBm for 9700	Specification of a value in [dBm]. The Received Signal Strength Indicator (RSSI) is an indicator of the reception field strength. Reception can be limited to a specific field strength range, thus also defining a certain reception range, by specifying a filter value. Note the negative sign preceding the value. The following applies: The closer to "0" the set value is, the shorter the distance required for reception will be and the closer the cell phone must be held to the reader.
SearchDuration=	50 to 2,000 Default: 200	Specification of a value in milliseconds (ms). The value specifies the interval within which the reader searches for a Bluetooth component (here: a cell phone) for data transmission.
BlockXConvertionMode BlockXConvertionFromDigit BlockXConvertionLengthBefore BlockXConvertionLengthAfter BlockXConvertionMask		Parameter for data conversion. The **BlockXConvertionMode** to **BlockXConvertionMask** parameters are described in detail in chapter [▶ 14.1.8].

[MobileStructure11]

This section contains general parameters for the transmission technology NFC with ISO 14443 A.

Parameter	Possible values	Meaning
HFMode=	2	The **HFMode** parameter defines the transmission technology to be used. 2= NFC with ISO 14443 A
BlockXConvertionMode BlockXConvertionFromDigit BlockXConvertionLengthBefore BlockXConvertionLengthAfter BlockXConvertionMask		Parameter for data conversion. The **BlockXConvertionMode** to **BlockXConvertionMask** parameters are described in detail in chapter [▶ 14.1.8].

Configuration example (not copyable) for a "mediadef.ini" file set without TapGo function.



```
//
[MobileGeneral]
ProjectID = 0xbd 0x23 0x4a 0xb6
FileID =
InteractionMode = 0

[MobileStructure10]
HFMode = 6
RSSIFilter = -40
SearchDuration = 200

// Index
Block0ConvertionMode=HexToASCII
Block0ConvertionFromDigit   = 1
Block0ConvertionLengthBefore= 1
Block0ConvertionLengthAfter = 1
Block0ConvertionMask=

// static credential data
Block1ConvertionMode = HexToDecASCII
Block1ConvertionFromDigit = 2
Block1ConvertionLengthBefore = 10
Block1ConvertionLengthAfter = 21
Block1ConvertionMask =


[MobileStructure11]
HFMode = 2

// Index
Block0ConvertionMode=HexToASCII
Block0ConvertionFromDigit   = 1
Block0ConvertionLengthBefore= 1
Block0ConvertionLengthAfter = 1
Block0ConvertionMask=

// static credential data
Block1ConvertionMode = HexToDecASCII
Block1ConvertionFromDigit = 2
Block1ConvertionLengthBefore = 10
Block1ConvertionLengthAfter = 21
Block1ConvertionMask =

```

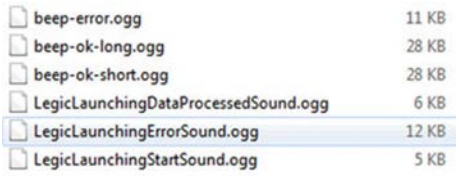
14.2 Files in the path /data/data/com.kaba.apps.ba/files/audio

The /audio directory contains several audio files in the .ogg format that can be played, depending on the parameter settings.

Users can also store self-created audio files.

In this case, the following hints must be observed:

- audio files may only be stored in the .ogg file format.
- the existing file names may **not** be changed! Self-created audio files must be stored under the given file names. The preinstalled audio files should be backed up before.
- The total file size of all audio files may not exceeded due to performance reasons. Before storing self-created audio files the file size must have been checked. The max. file size of **all** audio files is **100 kb**.



14.3 Files in the path /data/data/com.kaba.apps.hr/files/init

14.3.1 application.ini

After changes have been made in the application.ini file, a restart of the client is necessary.

[ProgramNumbers]

The [ProgramNumbers] section contains the automatically generated program number.

Parameter	Possible values	Meaning
MainClass=	Automatically generated program number.	This parameter is applied automatically and is used for update purposes. It may not be changed!

[MasterRecordDefinition]

The [MasterRecordDefinition] section defines the number of digits of the ID-number and the PIN used in the master record.

Parameter	Possible values	Meaning
IDLength=	1 to 20 Default: 7	The **IDLength** parameter defines the number of digits of the ID-number.
PINLength=	1 to 10 Default: 4	The **PINLength** parameter defines the number of digits of the PIN.

[InputsOutputs]

The [InputsOutputs] section defines properties of the digital inputs and outputs.

Parameter	Possible values	Meaning
Enabled=	true	For development purposes only. The parameter **Enabled** allows deactivating of digital inputs and outputs. Do not change the factory settings.

[Keyboard]

The [Keyboard] section contains settings affecting the keyboard.

Parameter	Possible values	Meaning
Scrambling=	true false	If the parameter is set to "true", the keyboard ciphers 0-9 will be rearranged after the numeric PIN entry when calling the dialog again. No scrambling takes place when entering the biometric ID via keyboard.

[FunctionKeys]

The [FunctionKeys] section contains all parameters affecting applications of the function keys. The specification KeyXX defines the number of the function key the parameters are valid for. The numbering of the function keys depends on the device type. XX may range from 1....5 and 31....40

Parameter	Possible values	Meaning
KeyXXURL=	Complete URL Default: no URL	The parameter **KeyXURL** defines the start page for the browser if enabled with the function key parameters. The URL can be defined with tags to make it individual (note 6).
KeyXXURLOperationTime=	Default: operating timeout defined with X01 record.	The parameter is currently not used.
Key0PartnerClass=		The parameter **Key0PartnerClass** defines the function key that is used to call an Android app via the class path. If an app without data transfer is used, the class definition is sufficient. If the partner application needs more static parameters, up to 10 <name=value> pairs (in angle brackets) can be defined. Only possible with Key0. **Example:** <url=http://10.10.15.1:8080/appname><version=4.7>
Key0PartnerService=		The parameter **KeyXXPartnerService** defines the function key that is used to call a partner service. A service must be defined, if an app is used together with data exchange. If the partner application needs more static parameters, up to 10 <name=value> pairs (in angle brackets) can be defined. Only possible with Key0. **Example:** <url=http://10.10.15.1:8080/appname><version=4.7>
KeyXXApp=	Name of the app package	The parameter **KeyXXApp** is used to call any App by operating a function key.
KeyXXReader2Active=	true false (default)	The **KeyXXReader2Active** parameter defines whether Reader2 is also permitted as an alternative to Reader1 for an input process. The parameter is only effective if Reader1 is active for the 1st or 2nd badge input in the operating mode 6 of the corresponding function key. Only then Reader2 can be activated as an alternative. Adjusting the raw badge data: The raw badge data of Reader 1 must be adapted to the format provided by Reader2.

[Buffers]

The [Buffers] section allows to adapt various memory parameters.

NOTICE

Changing of the memory structure due to parameter setting.

All stored data in the memory will be lost if changing one of these parameters.

Backup all data before changing any parameter.

Parameter	Possible values	Meaning
BufferMemoryRecordSize=	Default: 10,000	The **BufferMemoryRecordSize** parameter defines the size (= number of records) of the memory for the booking records (../Bedanet/data/ps.dat). The value can only be changed if the buffer is empty. (buffer backup must be taken in account).
BookingControlRecordSize=	Default: 1,000	The **BookingControlRecordSize** parameter defines the size of a backup buffer saving every booking record independently of terminal status (Online / Offline) and the property "Buffer data" or coming from standard or partner application. If the buffer is full, the oldest record will be overwritten. If the buffer size is changed, all records will be lost.
TransmitControlRecordSize=	Default: 1,000	The **TransmitControlRecordSize** parameter defines the size of a backup buffer saving every transmitted or received record from communication, ftp transfer and partner application (bookings, uploads, downloads, alarms etc). Every record is indicated with a header (T for transmitted, R for received) and a time tag. If the buffer is full, the oldest record will be overwritten. If the buffer size is changed, all records will be lost.

[Measurement]

In the [Measurement] section the measuring unit for the temperature monitoring in the terminal is defined.

Parameter	Possible values	Meaning
Temperature=	C (default) or F or K	The "**Temperature**" parameter defines the measuring unit in which the temperature measuring in the terminal shall take place. Possible values are C for °Celsius, F for °Fahrenheit and K for °Kelvin. The parameter may be defined if necessary. If the parameter does not exist or the defined parameter value is invalid, the default value "C" for degree Celsius is valid.

[CapacitiveNavigationKeys]

The [CapacitiveNavigationKeys] section allows you to define operating times for the "Menu" and "Back" key. This section does not exist in delivery state and must be created if necessary.

Parameter	Possible values	Meaning
MenuTimeDelay=	1-15 Default: 4	The **MenuTimeDelay** parameter defines a time measured in seconds the "Menu" key must be operated to open the menu bar after password input. If the value is <1, 1 second is used internally. If the value is >15, 15 seconds are used internally.
BackTimeDelay=	1-15 Default: 4	The **BackTimeDelay** parameter defines a time measured in seconds the "Back" key must be operated to open the password input menu for shutting down the client. If the value is <1, 1 second is used internally. If the value is >15, 15 seconds are used internally.

[TextToSpeech]

The [TextToSpeech] section contains parameters for the audio response of texts and booking responses.

For this feature, the TextToSpeech engine in the Android operating system must be activated, see the related chapter in the technical manual of the terminal.

Parameter	Possible values	Meaning
FunctionKeyTextsEnabled=	true false (default)	The parameter activates (=true) or deactivates (=false) the audio output of function key texts und dialog texts. After changing the parameter a reboot of the client is mandatory.
ShortBookingResponseEnabled=	true false (default)	The parameter activates (=true) or deactivates (=false) the acoustic output of booking responses. This concerns the - R1 record - RU record and - internal positive booking response. After changing the parameter a reboot of the client is mandatory. s

The following sections are intended for a variable assingment of inputs and outputs.

After changing parameters in one of these sections the B-Client HR40 needs to be restarted.

A practical example of such an assignment is given in chapter [▶ 3.16.3].

[IOMapping]

The [IO mapping] section is used for the flexible assignment of inputs and outputs.

This is done via the IO assignment table in conjunction with one section each for inputs and outputs.

Notation rules

- BIxx: logical input
- BOxx: logical output
- xx: number of the in-/ output. A leading zero can be omitted, see the examples below.
- Int: internal in-/ output in the terminal

In this section, a physical input/output is assigned to a logical input/output.

Examples:

Example 1: BI01 = Int I01. Without leading zero: BI01 = Int I1

Example 2: BO01 = Int O01. Without leading zero: BO01 = Int O1

[InputFunctions]

In this section, a specific function is assigned to a logical input. Only functions that are assigned to a logical input must be listed.

Logical inputs can be assigned to a function multiple times.

This may be:

- FrameContact (= frame contact of the door)
- DoorOpenerKeyInside (= door opener button inside the room)
- DoorOpenerKeyOutside (= door opener button outside the room)
- BlockTerminal (= blocking the terminal).

Examples:

Example 1: FrameContact = BI01

Example 2: DoorOpenerKey = BI02

[OutputFunctions]

In this section, a specific function is assigned to a logical output. Only functions that are assigned to a logical output must be listed.

Logical outputs can be assigned to a function multiple times.

This may be:

- DoorOpenerRelayInside (= door opener relay for "OUT" bookings)
- DoorOpenerRelayOutside (= door opener relay for "IN" bookings)

Examples:

Example 1: DoorOpenerRelayInside = BO01

Example 2: DoorOpenerRelayOutside = BO01

Without entries having been made in the "application.ini file", the following default settings are valid:

```
[IOMapping]
BI01 = Int I01
BI02 = Int I02
BO01 = Int O01

[InputFunctions]
FrameContact = BI01
DoorOpenerKeyInside = BI02

[OutputFunctions]
DoorOpenerRelayOutside = BO01
```

14.3.2 encoding.ini

The encoding.ini file may be generated if necessary. It must contain a character map chart if the device is supposed to display special characters via Special Encoding by means of a self-defined Unicode reference chart, e. g. for Japanese. A detailed instruction on how to define and use a self-defined Unicode reference chart is given in chapter [▶ 13.8.8].

14.3.3 text.ini

The text.ini file contains the column headings for use in dynamic lists in the template-dependent sections. More detailed information on the design of dynamic lists can be found in chapter.

[Table_Template_nnn]

Within a section, the column headings are listed template-dependent. The extension "nnn" must be the same as the extension of the list the column headings are to belong to, e.g. [Table_Template_002] for list <list name>.002.

Parameter	Possible values	Meaning
Col_Caption_nn=	Text	Column heading for dynamic tables without language identification. For the extension "nn", the number of the table column this heading is to apply to must be indicated, e.g. 01
Col_Caption_nn_de= Col_Caption_nn_en= Col_Caption_nn_fr=	Text	Column heading for dynamic tables with language identification. For the extension "nn", the number of the table column this heading is to apply to must be indicated, e.g. 01.

14.3.4 interface.ini

The "interface.ini" file contains parameters that define the "look&feel" i.e. the appearance of the terminal´s display surface. This includes colors, graphic elements, font definitions, gaps between labels or elements and other.

In order to allow customer specific layouts, several predefined designs are included in delivery that makes an adaption to customer needs very simple. These designs may be freely adapted or reconfigured.

After having make changes in a design, a reboot is mandatory!

These designs have the following designation:

Designation	Appearance/Remark
Design_00_interface.ini	Classic design (colored buttons on the bottom)
Design_01_interface.ini	Design 1, dark, equates to default interface.ini file
Design_02_interface.ini	Design 1, bright
Design_03_interface.ini	Design 2, dark
Design_04_interface.ini	Design 2, bright
Design_05_interface.ini	Design 3, dark
Design_06_interface.ini	Design 3, bright
Design_07_interface.ini	Design 4, dark

For displaying, the system always uses the design with the designation "interface.ini". If one of the above designs is to be used, the desired file must be renamed to "interface.ini". This way all the definitions in this file are used for displaying.

The existing default interface.ini file should be renamed as backup file and saved.

The parameters in the "interface.ini" file may be adapted within reasonable limits. Self defined designs should be saved as a backup in order to keep them in cases of overwriting the definitions.

[LocalEnrollment]

The parameters in the [LocalEnrollment] section refer to the images shown during the local enrollment process. This parameters must be changed only if the default settings does not match the requirements.

Parameter	Possible values	Meaning
BioImageLeft=	Default: BioImageLeft.png	Image: place finger more to the left
BioImageRight=	Default: BioImageRight.png	Image: place finger more to the right
BioImageUp=	Default: BioImageUp.png	Image: place finger more to the top edge
BioImageDown=	Default: BioImageDown.png	Image: place finger more to the bottom
BioImagePressHarder=	Default: BioImagePressHarder.png	Image: place finger more tightly
BioImageLatency=	Default: BioImageLatency.png	Image: latent fingerprint image detected.

[Languages]

The [Languages] section defines all parameters that are used to change the user language.

Parameter	Possible values	Meaning
	Example:	
Language0=	de	With parameter **LanguageX** it is possible to define up to 5 user languages with a corresponding image (LanguageXImage). The language is defined by a locale (see note 4); for many languages the default texts and flag images are predefined, see note 7. Note: If no language is defined the terminal is working in compatibility mode (texts without locale).
Language0Image=	Germany.png	
Language1=	en_GB	
Language1Image=	UnitedKingdom.png	
....	...	
Language4=	Default:	
Language4Image=	No language	

[FunctionKeys]

The [FunctionKeys] section contains all parameters affecting the function keys. The specification KeyXX defines the number of the function key the parameters are valid for. The numbering of the function keys depends on the device type. XX may range from 1....5 and 31....40.

Parameter	Possible values	Meaning
KeyXXBounds=	Depending on device type	**KeyXBounds** defines the position and size of the function key represented by an image, see note 2. If no settings are made, the default values are valid.
KeyXXLabelColor=	Depending on device type Example: white	**KeyXXLabelColor** defines the background color of the function key text; **KeyXXFontColor** the color of the text itself, see note 1. If no settings are made, the default values are valid.
KeyXXFontColor=		
KeyXXImageEnabled=	Example: KommenEnabled.png	These parameters allow to change the image of the function keys, depending on the terminal status. If it is not possible to press a function key, the image **KeyXXImageDisabled** is shown; otherwise the image **KeyXXImageEnabled**.
KeyXXImageDisabled=	Example: KommenDisabled.png	
KeyXXImageActive= KeyXXImageActiveDisabled= KeyXXImageEnabled_de= KeyXXImageDisabled_de= KeyXXImageEnabled_en_GB= KeyXXImageDisabled_en_GB= KeyXXActive_de =		In active state (pressed or time controlled) the images **KeyXXImageActive** or **KeyXXImageActiveDisabled** are shown. For all images a language specific version can be set (e.g.: KeyXXImageEnabled_en_GB, see note 7). If one of the images is not indicated, the logical next image is used. If no image is indicated, the function key is invisible. For all images a language specific version can be set (e.g.: KeyXXImageEnabled_en_GB, see note 7). If one of the images is not indicated, the logical next image is used. If no image is indicated, the function key is invisible. The **Tags**... parameters allow the labeling of the function keys (M00 – M05; m00 – m05 M31 – M40; m31 – m40). The definitions made here are valid for all function keys that have no particular definitions made via the KeyXXTag... parameters.

Parameter	Possible values	Meaning
TagsEnabled=	true false (default)	The **TagsEnabled** parameter activates the visibility of the tags.
TagsRelativeBounds=		**TagsRelativeBounds** defines the relative position and size of the tag, related to the KeyXXBounds parameter. For the definition of position and size see note 2. Negative values are possible.
TagsColor=		**TagColor** defines the color of the tag, see note 1.
TagsFont=		**TagFont** defines the font, see note 3.
TagsFontColor=		**TagFontColor** defines the color of the font, see note 1.
TagsInFront=	true false	**TagInFront** defines, whether the text is situated in front of the image or behind it.
TagsAlignment=	Depending on device type	**TagsAlignment** defines the horizontal representation, see note 5.
TagsAlignmentV=	Depending on device type	**TagsAlignmentV** defines the vertical representation, see note 5.
KeyXXTagSpecificEnabled=	true false	The **KeyXXTag**... parameters allow the labeling of the function keys (M00 – M05; m00 – m05 M31 – M40; m31 – m40). The values may be defined for each function key individually. See **TagsEnabled.**
KeyXXTagBounds=	Depending on device type	**KeyXXTagBounds** defines the position and size of the tag in relation to the parameter KeyXXBounds. For the definition of position and size see note 2. Negative values are possible.
KeyXXTagAlignment=	Depending on device type	**TagsAlignment** defines the horizontal representation, see note 5.
KeyXXTagAlignmentV=	Depending on device type	**TagsAlignmentV** defines the vertical representation, see note 5.
KeyXXTagColor=		See TagsColor
KeyXXTagFont=		See TagsFont
KeyXXTagFontColor=		See TagsFontColor
KeyXXTagInFront=		See TagsInFront

[SurfaceDesign]

The [SurfaceDesign] section contains all parameters that influence the design of the user interface. There are lots of possibilities to design an individual surface. The parameters relating to dynamic lists are described in chapter.

Parameter	Possible values	Meaning
DisplayInfoVisibleNumber=	1 bis 10 Default =4	**DisplayInfoVisibleNumber** defines the number of visible entries (1 to 10; default = 4).
ShowListDelay=	Default=0	**ShowListDelay** defines a delay time in seconds between the displaying of the booking response and the displaying of the display info.
BackgroundColor=	Default: 0 0 0 200	**BackGroundColor** defines the background color of the screen, see note 1.
BackgroundImage=	Depending on device type	**BackgroundImage** defines a background image.
DialogBackgroundColor=	Depending on device type	**DialogBackgroundColor** defines the background color of the menus and the info screen of host and network settings, see note 1.
DialogForegroundColor=	Depending on device type	**DialogForegroundColor** defines foreground color of the menus (font and frame), see note 1.
DialogLabelImage=	Depending on device type	**DialogLabelImage** defines a dialog image.
BadgeImageBounds=	Depending on device type	**BadgeImageBounds** defines bounds and size of an image that represents a badge, see note 2.
TimeBounds= TimeFont= TimeFontColor= TimeStyle= TimeColor= TimeAlignment= TimeAlignmentV= DateBounds= DateFont= DateFontColor= DateStyle= DateColor= DateAlignment= DateAlignmentV=	Depending on device type	**TimeX** and **DateX** define the presentation of the date and the time. **XBounds** defines position and size, see note 2. **XFont** defines the font, see note 3. **XFontColor** defines the color of the font, see note 1. **XStyle** defines the format of the data, see note 11. **XColor** defines the background color of the label, see note 1. The format of the data depends on the current locale und corresponds to normal format used with that language and country. **XAlignment** defines the horizontal representation, see note 5. **XAlignmentV** defines the vertical representation, see note 5.

Parameter	Possible values	Meaning
KeyLabelBounds= KeyLabelFont= KeyLabelSignalColor= KeyLabelFontSignalColor= KeyLabelAlignment= KeyLabelAlignmentV=	Depending on device type	**KeyLabelX** define the labels for the function key texts in the upper line. **KeyLabelBounds** defines position and size, see note 2. **KeyLabelFont** defines the font, see note 3. **KeyLabelSignalColor** and **KeyLabelFontSignalColor** define the colors of label and font in case of an error message, see note 1. **KeyLabelAlignment** defines the horizontal representation, see note 5. **KeyLabelAlignmentV** defines the vertical representation, see note 5. The colors for normal state are defined with the function keys (parameters **KeyXLabelColor** and **KeyXFontColor** in the section [FunctionKeys]).
HeaderLineEnabled=	true false	**HeaderLine** defines position and size of the line which is shown in the upper part of the display, see note 2.
HeaderLineBounds=	Depending on device type	**HeaderLineBounds** defines position and size of the line in the upper part of the display, see note 2.
HeaderLineColor=	Depending on device type	**HeaderLineColor** defines the color of the line in the upper part of the display, see note 1.
WeekdayEnabled=	true false	**WeekdayX** define the representation of the weekday labels in the display. **WeekdayEnabled** defines whether the weekday labels are displayed (true) or not (false). **WeekdayBounds** defines position and size, see note 2. **WeekdayFontColor** defines the color of the font, see note 1. **WeekdayFont** defines the font, see note 3. **WeekdayStyle** defines the display format of the date, see note 11. **WeekdayColor** defines the background color of the date label, see note 1. **WeekdayAlignment** defines the horizontal orientation, see note 5. **WeekdayAlignmentV** defines the vertical orientation, see note 5.
WeekdayBounds= WeekdayFont= WeekdayFontColor= WeekdayStyle= WeekdayColor= WeekdayAlignment= WeekdayAlignmentV=	Depending on device type	

Parameter	Possible values	Meaning
DialogLabel0Bounds= ... DialogLabel5Bounds=	Depending on device type	The parameters **DialogLabelX** define labels for user dialogs. The parameters **DialogLabelXBounds** define position and size, see note 2.
DialogLabel0Font= ... DialogLabel5Font=	Depending on device type	The parameters **DialogLabelXFont** define the font, see note 3. The parameters **DialogLabelXLength** define the maximum number of characters that can be shown. Enter a numeric value.
DialogLabel0Length= ... DialogLabel5Length=	Depending on device type	**Note:**The parameters **DialogLabelXBounds**, **DialogLabelXFont** and **DialogLabelXLength** must be adjusted in a way that the user information will be displayed completely.
DialogLabelStandardColor= DialogLabelFontStandardColor= DialogLabelSignalColor= DialogLabelFontSignalColor= DialogLabelExtraColor= DialogLabelExtraColor= DialogLabelFontExtraColor= DialogILabelAlignment= DialogLabelAlignmentV=	Depending on device type	**DialogLabelStandardColor**, **DialogLabelSignalColor**, **DialogLabelFontStandardColor**, **DialogLabelFontSignalColor**, define the colors of the label or font in normal or error mode, see note 1. The parameters **DialogLabelExtraColor** and **DialogLabelFontExtralColor** define element and font color only for mailbox texts, see note 1. The parameter **DialogLabelAlignment** defines the horizontal orientation, see note 5. The parameter **DialogLabelAlignmentV** defines the vertical orientation, see note 5.

Parameter	Possible values	Meaning
DialogInfoLineLength= DialogInfoFont= DialogInfoColor= DialogInfoFontColor= DialogInfoAlignment= DialogInfoAlignmentV=	Depending on device type	The parameters **DialogInfoX** define an input field. They have the following meaning: **XInfoLineLength** defines the number of characters in each line. **XInfoFont** defines the font and its size; **XInfoColor** defines the color of the field; **XInfoFontColor** defines the color of the font. **DialogInfoAlignment** defines the horizontal orientation, see note 5. **DialogInfoAlignmentV** defines the vertical orientation, see note 5.
ProgressIndicatorBounds=	Depending on device type	**ProgressIndicatorBounds** defines size and position of the graphic progress indicator, see note 2.
ProgressDisplayDelay=	Angabe in Millisekunden. Default: 100	**ProgressDisplayDelay** defines the rotation speed of the graphic progress indicator.
LogoImage=	<dateiname.png>	The parameters **LogoX** define properties for a company logo or background image. **LogoImage** defines the image, see note 7. Remark: the logo is placed in the foreground.
LogoBounds=	Depending on device type	**LogoBounds** defines position and size, see note 2.
LanguageEnabled=	true false	**LanguageEnabled** defines, whether the image for language switchover (flag) is displayed (true) or not.
LanguageBounds=	Depending on device type	**LanguageBounds** defines position and size of the image (e.g. a flag) used for switching the user language, see note 2.
LegicWriteImageBounds=	Depending on device type	**LegicWriteImageBounds** defines size and position of the following symbols: (see note 2). • Update required (CardLink) and • Error during update process (CardLink). The parameter may be defined if required.
DisplayLegicWriteErrorImage=	true false (default)	If an error occurs during the writing operation of the update or validation data, it can be displayed for the user at the terminal by means of an error symbol along with the data record. **DisplayLegicWriteErrorImage** defines, whether the symbol is displayed or not. The parameter may be defined if required.

Parameter	Possible values	Meaning
DisplayInfoShowText=	true (text visible) false (text not visible)	The parameters **DisplayInfoX** define the properties of the window showing the display info. **DisplayInfoShowText** defines whether the defined info texts are visible or not.
DisplayInfoFont= DisplayInfoColor= DisplayInfoTextFontColor= DisplayInfoHeaderColor= DisplayInfoHeaderFontColor= DisplayInfoHeaderFont= DisplayInfoHeaderAlignment= DisplayInfoHeaderAlignmentV= DisplayInfoLabelColor= DisplayInfoLabelFontColor= DisplayInfoLabelFont= DisplayInfoLabelAlignment= DisplayInfoLabelAlignmentV=	Depending on device type	The parameters **DisplayInfoX** define the properties of the window showing the display info. **DisplayInfoShowText** defines whether the defined info texts are visible or not. **DisplayInfoFont** defines the font, see note 1. The parameters **DisplayInfoColor** and **DisplayInfoTextFontColor** define the colors of the labels, see note 1. **DisplayInfoHeaderX** define the upper line (Header) of the display info. The parameters have the following meaning: **DisplayInfoHeaderColor** defines the background color of the header line, see note 1. **DisplayInfoHeaderFontColor** defines the color of the font of the header line, see note 3. **DisplayInfoHeaderAlignment** defines the horizontal orientation, see note 5. **DisplayInfoHeaderAlignmentV** defines the vertical orientation, see note 5. The parameters **DisplayInfoLabelX** define the label area of the display info. **DisplayInfoLabelColor** defines the background color of the label area, see note 1. **DisplayInfoLabelFontColor** defines the color of the font of the label text, see note 1. **DisplayInfoLabelFont** defines the font, see note 3. **DisplayInfoLabelAlignment** defines the horizontal orientation, see note 5. **DisplayInfoLabelAlignmentV** defines the vertical orientation, see note 5.

Parameter	Possible values	Meaning
DisplayInfoTextColor= DisplayInfoTextFont= DisplayInfoTextFontColor= DisplayInfoTextAlignment= DisplayInfoTextAlignmentV=	Depending on device type	The parameters **DisplayInfoTextX** define the text area (data to be displayed) of the display info. The parameters have the following meaning: **DisplayInfoTextColor** defines the color of the data area, see note 1. **DisplayInfoTextFont** defines the font, see note 3. **DisplayInfoTextFontColor** defines the color of the font of the data area, see note 1. **DisplayInfoTextAlignment** defines the horizontal orientation, see note 5. **DisplayInfoTextAlignmentV** defines the vertical orientation, see note 5.
Theme=	01 (default) 02 0x (numeric value)	The **Theme** parameter is used for multi-colored user guidance. The graphics for user guidance that can be selected for display using this parameter, see chapter [▶ 13.7]. In the delivery state, this parameter is set to "Theme=01". Accordingly, the graphics for user guidance are displayed with white lines (default). If the parameter is set to "Theme=02", the (delivered) graphics for user guidance will be displayed with dark lines. Customized graphics in other colors receive a "theme extension" in the filename. Reference is then made to this extension, e.g. "Theme=03". The customized graphics must be generated accordingly and saved to the /data/data/com.kaba.apps.ba/files/init/ images file.

Parameter	Possible values	Meaning
		The **InfoLabelX** parameters (X = 1 or 2) allow to display 2 labels on the terminal´s screen. The content of the labels is provided in the labelinfo.ini file. This file is read-out periodic after a definable time. This way it is possible to update and display information that are changing time to time again, e. g. stock quotations.
InfoLabelRefresh=	1 (default) bis 1.500	The **InfoLabelRefresh** parameter defines the time in minutes, the labelinfo.ini file is read out again. The parameter is valid for both labels.
InfoLabelXEnabled=	true false (default)	The **InfoLabelXEnabled** defines, whether the label is displayed (=true) or not (false).
InfoLabelXBounds=	Depending on device type	The **InfoLabelXBounds** parameter defines the position and size of the label, see note 2.
InfoLabelXDefaultColor=	black, blue,green, red, white, yellow or no value für transparent.	The **InfoLabelXDefaultColor** parameter defines the color of the label.
InfoLabelXFont=		The **InfoLabelXFont** parameter defines the font, see note 3.
InfoLabelXFontColor=	black (default) blue, green, red, white yellow	The **InfoLabelXFontColor** parameters defines the default font color. This parameter is valid, if no font color is set in the InfoXColor parameter in the labelinfo.ini file
InfoLabelXAlignment=	left center right	The **InfoLabelXAlignment** parameter defines the alignment of the text within the label.

[Table_Images]

The [Table_Images] section links the graphics to be displayed when using dynamic lists to a variable name. The variable name is used in the list definition (e.g. on a server). You have to make sure that the extension of each variable name is identical to that of the associated link.

More detailed information on the design of dynamic lists can be found in chapter [▶ 7.7.8].

Parameter	Possible values	Meaning
Image1=	<filename>.png	Links the "Image<n>" variable to a graphic.
Image2=	<filename>.png	
Image3=	<filename>.png	
Image4=	<filename>.png	
...=	...	
...=	...	

[Table_Template_nnn]

The [Table_Template_nnn] section contains all parameters for the appearance and properties of a dynamic list. The extension "nnn" must be the same as the extension of the list the template is to belong to.

Example:

[Table_Template_002] for list <list name>.002.

If different lists must have the same appearance, they can be linked to the same section via the extension.

Example:

[Table_Template_002] for list <list name1>.002.

[Table_Template_002] for list <list name1>.002.

The parameters of this section are described in chapter .

[Upload]

The [Upload] section contains parameters used for uploading parameter files.

Parameter	Possible values	Meaning
ParameterFile0=	Default:	With the entries **ParameterFileX** those .ini-files are defined that shall be uploaded with a complete upload request. Before uploading, the parameter files are divided in records of type >3z00. It is possible to transmit every file using the .ini-file format.
ParameterFile1=	File upload not active.	
...=	**Example:**	
ParameterFile9=	data/data/com.kaba.apps.ba / files/init/debug.log	

14.3.5 labelinfo.ini

Section [Label1] and section [Label2]

The sections [Label1] and [Label2] contain info texts and info text colors for the labels 1 and 2. If the labelinfo.ini file does not (or no more) exist, no text will be displayed in the label.

Parameter	Possible values	Meaning
Info1=	Text	Info text line 1
Info1Color=	black, blue, green, red, white, yellow	Color of the info text. If no color is defined here, the color defined in the parameter InfoLabel1FontColor in the [SurfaceDesign] section of the interface.ini file is used.
Info2=		Info text line 2
Info2Color=		See Info1Color
Info3=		Info text line 3
Info3Color=		See Info1Color
Info4=		Info text line 4
Info4Color=		See Info1Color

Delivery state:

In the delivery state the labels defined in the "interface.ini" file are disabled and therefore invisible.

14.3.6 kllck.ini

The kllck.ini file is an important system file. **No customizations are permitted in this file!**

14.3.7 sop.ini

The sop.ini file is the license file of the device. For optional extensions of the terminal software, a new license file must be stored in this file.

NOTICE

Loss of the license

Changes or modifications at an existing license file make it invalid and therefore unusable.

- Do not change or modify the license file manually.

14.3.8 Registration

The Registration file is an important system file. **No customizations are permitted in this file!**

15 Notes

1) Color

Colors (keyword Color included in parameter name) are defined with a predefined color name or a special RGB-value. If no value is defined, the application specific default color is used. All values between (0 0 0) and (255 255 255) are valid. The values are divided by blanks.

Name	RGB value		Name	RGB value
white	255 255 255		orange	255 200 0
lightGray	192 192 192		yellow	255 255 0
gray	128 128 128		green	0 255 0
darkGray	64 64 64		magenta	255 0 255
black	0 0 0		cyan	0 255 255
red	255 0 0		blue	0 0 255
pink	255 175 175			
Examples: xxxColor=green yyyColor=200 100 50				

Transparency

The color can additionally be assigned a transparency value. This value can be between 0 and 255. It defines to what extent a background color shines through a foreground color.

0 = Background shines completely through the foreground (foreground exhibits maximum transparency).

255 = Background does not shine through the foreground (foreground exhibits complete lack of transparency)..

Example: xxxColor= 200 100 50 255

2) Bounds

The bounds of a component (keyword Bounds included in parameter name) are defined by left upper edge in relation to left upper edge of the screen or parts of it (for example a button field) with x-position, y-position, width and height in this order. If no value is defined, the component is shown with its default position and size.

Allowed are all position and sizes that can be shown on the screen (standard: 800 x 480 pixel). The values are divided by blanks.

Example: xxxBounds=100 100 200 50

3) Font

Fonts (keyword Font included in parameter name) are defined by font name, style and size. If no font is defined the application specific default font is used.

Allowed values are:

- Font name: SansSerif
- Style: PLAIN (normal), BOLD, ITALIC and BOLD+ITALIC
- Size: Every value in point. The font size must be adapted to the display size.

In the dormakaba NexT terminal series the font size is determined overridingly by a parameter in the Android system settings. This parameter should be in the "normal size" setting. If displayed characters show any discrepancies from the defined font size in the .ini file, e. g. the characters are too large, please check the Android system settings.

The values are divided by blanks.

Examples:

- xxxFont=SansSerif PLAIN 20
- yyyFont=SansSerif BOLD+ITALIC 25

4) Language

Parameters which define a locale (e.g. languages with keyword Language included in parameter name) are set as follows. A locale is built of 1 to 3 components: language, country, variant (not used). At least the language must be set. The components are divided by blank or underline. If no value is defined the default language is used (in some cases the language of the operating system).

On enquiry it is possible to define other languages and countries corresponding to ISO 639:1988 "Code for the representation of names of languages" (2 characters) and ISO 3166 "Code for Countries" (2 characters). Predefined values can be found in chapter [▶ 13.8].

Examples:

- xxxLanguage=de
- yyyLanguage=en GB
- zzzLanguage=en_US

5) Alignment und AlignmentV

The alignment of texts (keyword Alignment and AlignmentV included in parameter name) is defined with predefined values. If no value is defined, the application specific alignment is used.

Allowed values are:

Alignment: no value (standard), LEFT, CENTER, RIGHT

AlignmentV: no value (standard), TOP, CENTER, BOTTOM

Example: xxxAlignment=CENTER

6) URL

A web page (keyword URL included in parameter name) is defined by an URL and shown by a browser. The URL has to be defined corresponding to internet standard for URLs in the format protocol://host:port/path/document; in the most cases the port is not used.

The URL can also be used to display files locally stored in the "scdard" directory or it´s subdirectories, see chapter [▸ 2.7.3].

Important: The directory name "sdcard" is **not** to be mentioned here in the URL, see examples.

Examples:

- Calling an internet page:

  URL=http://www.dormakaba.com
- Displaying of a local HTML page from the "sdcard/Download"... directory:

  URL=file:///Download/<Filename>.htm
- Displaying of a local graphics file from the "sdcard/Pictures"... directory:

  URL=file:///Pictures/<Filename>.jpeg
- Displaying of a local video file from the "sdcard/Movies"... directory:

  URL=file:///Movies/<Filename>.mp4.

7) Image

Images (keyword Image included in parameter name) can be defined in .png format. The images must be located in the directory .data/data/com.kaba.apps.ba/files/images. If no image is defined by the customer, the application defined default image is used.

Example: xxxImage=Image.png

8) Insets

Insets (keyword Insets included in parameter name) define the distance of the border of an inner component to the border of the outer component. The insets are defined by distance from the top, left, bottom and right border in this order. If no value is defined the component is shown with its default insets. Allowed are all values that are in the outer component. The values are divided by blanks.

Example: xxxInsets=5 5 5 5

9) Period

Times are defined by hour, minute and second; minute and second are optional. Time periods (keyword Period included in parameter name) are defined by two times (begin and end). Hour, minute and second are divided by ‚:' or ‚.'. Times are divided by blanks.

Please note: the time format is European style. This means, time data after noon must be defined in 24-h counting. (1.00 pm =13.00; 2.00 pm = 14.00 a.s.o).

Examples:

- xxxPeriod=10 12 — Period of time from 10 a.m. to noon
- yyyPeriod=5:30:00 7:30:30 — Period of time including minutes and seconds
- zzzPeriod=19.00 20.00 — Period of time from 7 p.m. to 8 p.m.

10) Size

The size of a component (keyword Size in the parameter name) is defined by width and height in this order. If no value is defined, the component is shown in its default size. Allowed are all sizes which can be shown on the screen (standard: 640 x 480 pixels). The values are divided by blanks.

Example: xxxSize=200 50

11) Style

The Style parameter defines the format of the data, where

FULL is the format with complete data

SHORT is the format with least data.

Allowed values are:

FULL

LONG

DEFAULT

MEDIUM

SHORT.

The format of the data depends on the current locale und corresponds to normal format used with that language and country.

Index

Icons

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z

Notes

040500544 - 08/2024

www.dormakaba.com

dormakaba Deutschland GmbH
Albertistraße 3
78056 Villingen-Schwenningen
Germany
T: +49 7720 603-0
www.dormakaba.com
Company headquarters: Ennepetal