SDK Iron Logic Protocol Description

version SDK: 1.1.5-1.1.6 (Ethernet)

1. General information

All the files necessary for the SDK are in the archive offered for downloading.

Modules used:

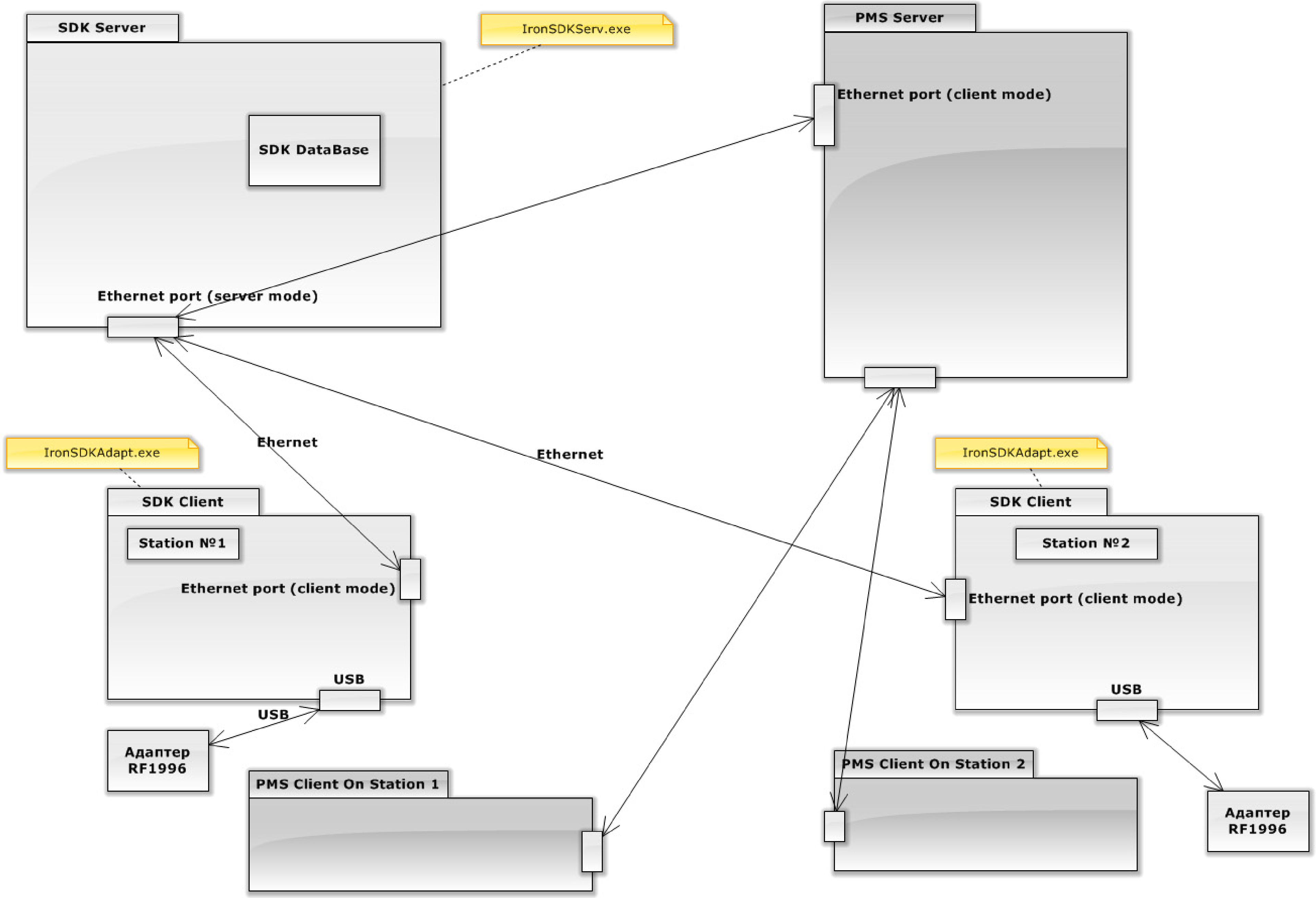
– **SDK server**. Back-end. Can be started via the IronSDKServer.exe file. Runs in a single instance on one of the computers on the network. The PMS back-end that serves workstation queries establishes a single connection to the SDK server.

– **SDK adapter**. Front-end. Can be started via the IronSDKAdapt.exe file. Runs on every workstation that has a card issuance adapter. Establishes a connection to the SDK server. It does not interact directly with the PMS, only through the SDK server.

– **PMS command emulator.** Utility software for testing the SDK protocol. The emulator can be started via the IronClientEth.exe file. It emulates a single PMS server connection to the SDK server.

The SDK server communicates with the PMS and workstations via an Ethernet connection through the TCP/IP protocol. When started, the SDK server opens a port for listening (by default, port 9999), but you can change it in the settings. The SDK server responds to PMS queries with query formats shown below.

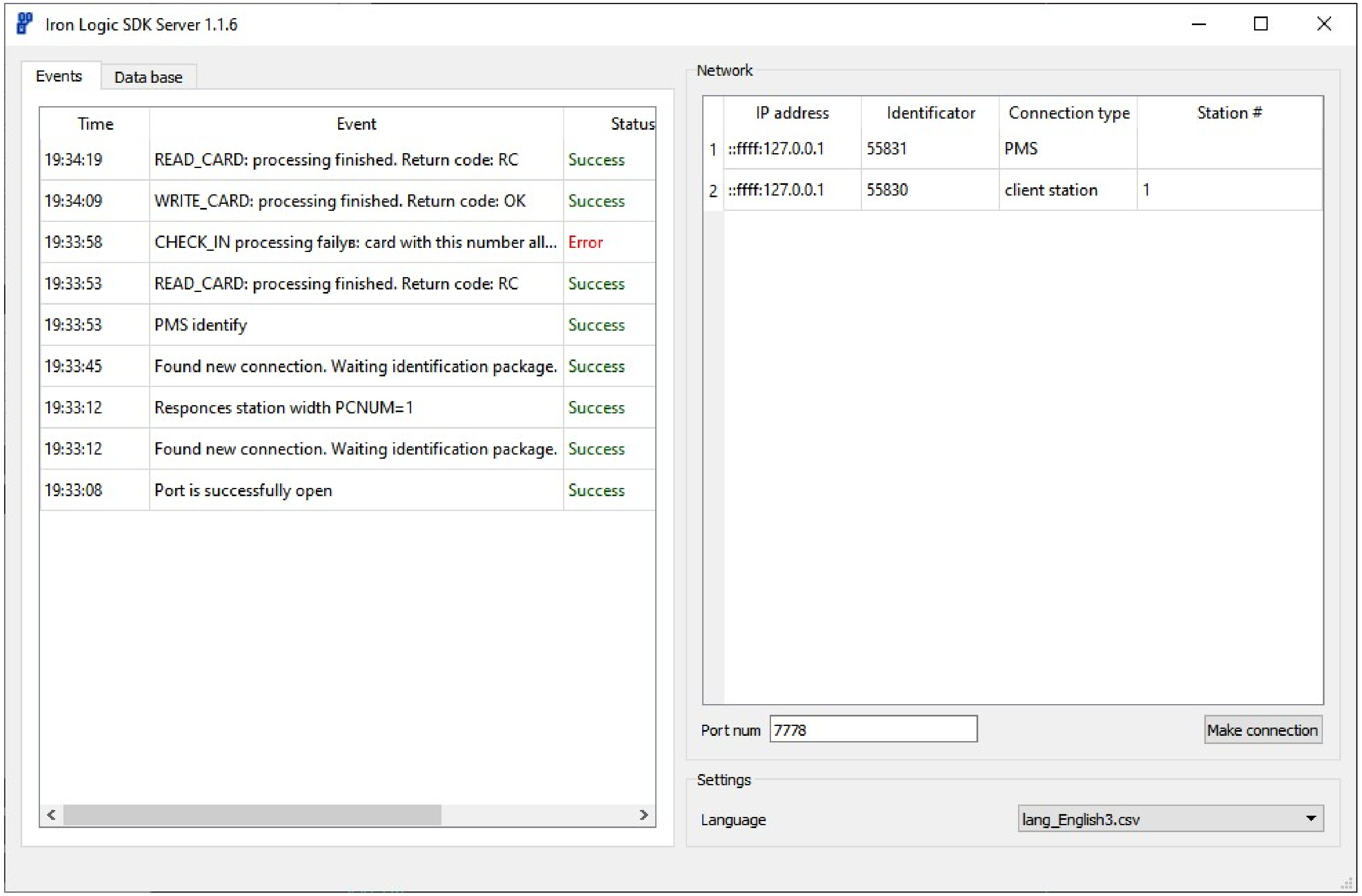
The interaction structure of the SDK back-end, workstations and PMS modules is shown in the figure on the following page:


SDK Server
IronSDKServ.exe
PMS Server
SDK DataBase
Ethernet port (client mode)
Ethernet port (server mode)
IronSDKAdapt.exe
Ehernet
Ethernet
IronSDKAdapt.exe
SDK Client
Station №1
Ethernet port (client mode)
USB
USB
Адаптер RF1996
PMS Client On Station 1
SDK Client
Station №2
Ethernet port (client mode)
USB
PMS Client On Station 2
Адаптер RF1996

1.1. Back-end configuration.

Runs on one of the computers on the local network **in a single instance**.

In the settings, the required port is specified and the "Establish a connection" command is executed:



Time	Event	Status
19:34:19	READ_CARD: processing finished. Return code: RC	Success
19:34:09	WRITE_CARD: processing finished. Return code: OK	Success
19:33:58	CHECK_IN processing failyв: card with this number all...	Error
19:33:53	READ_CARD: processing finished. Return code: RC	Success
19:33:53	PMS identify	Success
19:33:45	Found new connection. Waiting identification package.	Success
19:33:12	Responces station width PCNUM=1	Success
19:33:12	Found new connection. Waiting identification package.	Success
19:33:08	Port is successfully open	Success

	IP address	Identificator	Connection type	Station #
1	::ffff:127.0.0.1	55831	PMS	
2	::ffff:127.0.0.1	55830	client station	1

Port num: 7778

Language: lang_English3.csv

At this point, you may receive a warning from your antivirus software that the program is doing something wrong. This is a reaction to an attempt to open and listen to the port, so the situation can be solved by adding the program to the list of antivirus exceptions.

Visually, the SDK server software display is divided into two parts: left and right. The right side shows the established connections to the client modules and the PMS. Client sides are identified immediately upon connection and the PMS will be identified when the first command is sent. Please note that all client stations must have unique numbers (PC num) - they are used both in the protocol and in the settings of the client modules,

On the left side, the status of the SDK is shown. In the events tab, you can see the history of operations and their status, and on the "database" tab you can see the data on the doors that are registered in the SDK database.

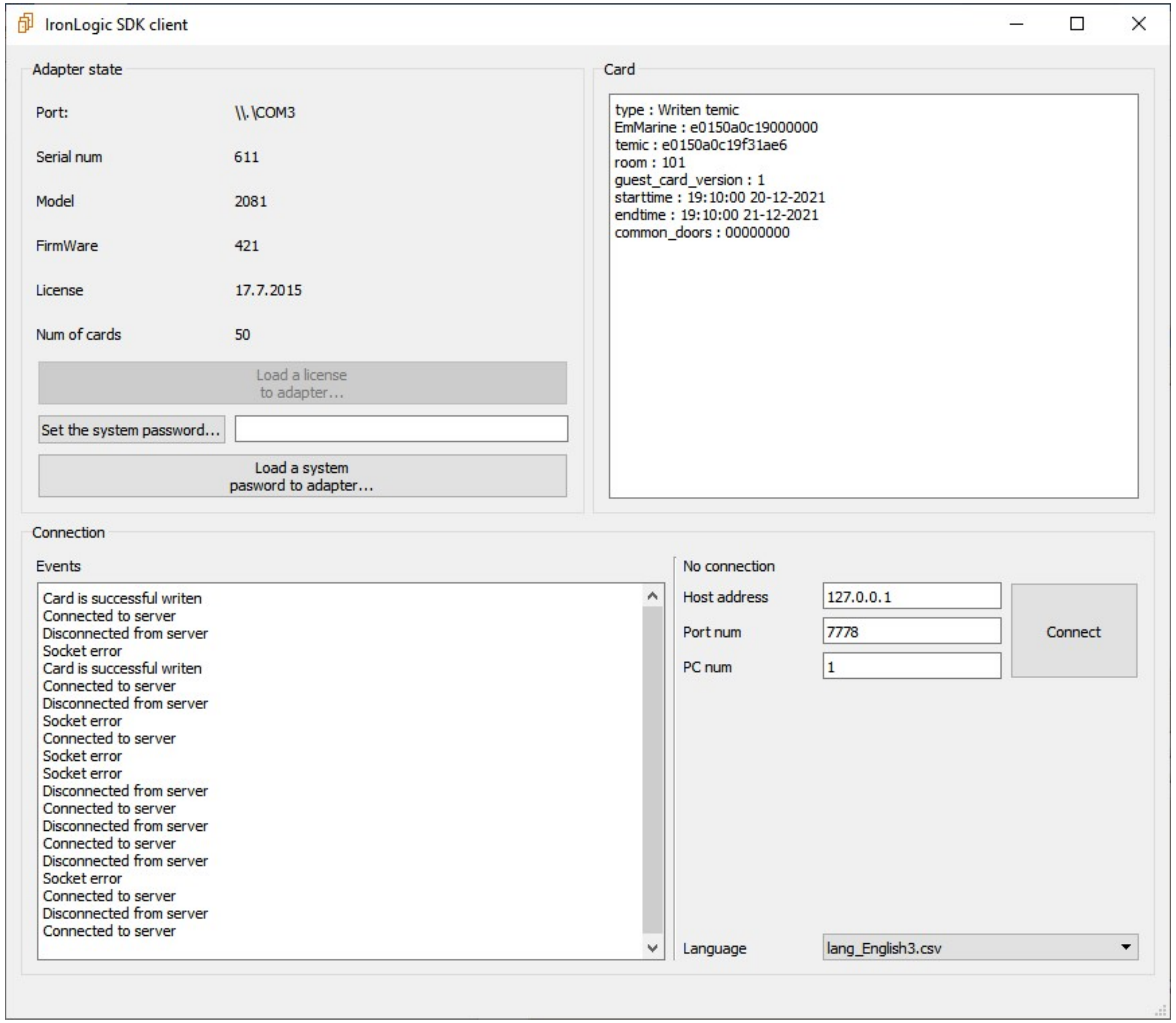
Automatic connection to the network

Starting from version 1.1.5, it is possible to establish a connection automatically without clicking the "establish a connection" button. To do this, in the command line parameters, you need to transmit the "**-autoconnect**" key, i.e. the startup command will look something like this: «**IronSDKServer.exe -autoconnect».**

1.2. Front-end configuration (workstations)

Each workstation connected to the RF1996 adapter must have ironLogic SDK Client software installed and run via the IronSDKAdapt.exe file.

The overview of the software display is shown in the figure:



IronLogic SDK client

Adapter state

Port:	\\.\COM3
Serial num	611
Model	2081
FirmWare	421
License	17.7.2015
Num of cards	50

Load a license to adapter...
Set the system password...
Load a system pasword to adapter...

Card

```
type : Writen temic
EmMarine : e0150a0c19000000
temic : e0150a0c19f31ae6
room : 101
guest_card_version : 1
starttime : 19:10:00 20-12-2021
endtime : 19:10:00 21-12-2021
common_doors : 00000000
```

Connection

Events

```
Card is successful writen
Connected to server
Disconnected from server
Socket error
Card is successful writen
Connected to server
Disconnected from server
Socket error
Connected to server
Socket error
Socket error
Disconnected from server
Connected to server
Disconnected from server
Connected to server
Disconnected from server
Socket error
Connected to server
Disconnected from server
Connected to server
```

No connection

Host address	127.0.0.1
Port num	7778
PC num	1
Language	lang_English3.csv

The RF-1996 adapter is required to issue cards to IronLogic locks. The software searches for it when it first starts. If the adapter is not found, a corresponding red message will be displayed at the top of the software display.

You need to establish a connection to the SDK back-end by setting its address, port number and using the "Connect" button.

Before establishing a connection, pay attention to the "Station Number" parameter - each workstation must have it unique so that the SDK could determine which of the clients the command is intended for.

Do not forget to enter a system password into the adapter, if you have not done this before! (For example, using the LockComander program, which also saves the system password to the adaptor.)

After establishing the connection, the software is completely ready for operation. In the upper-right of the software display ("card"), information about the card touching the adapter is displayed. The lower-left of the software display ("events") shows the status of command processing from the SDK server.

2. Accepted command format

The command system is based on the SDK Inhova. For format compatibility, data not used by the IronLogic system is transmitted as null values.

Data is transmitted as numbers and constants written in ASCII format, with the exception of service characters (they match the Inhova program):

Character designation	Character code (hex)	Character function
<SEP>	B3	Data separator
<STX>	02	Command start
<ETX>	03	Command end. This symbol is followed by a checksum
<ENQ>	05	Request a communication test. In response to such a character, the SDK will respond with a character <ACK>
<ACK>	06	Command receipt confirmation It is sent after receiving any command before the start of its processing.

All commands for the SDK have the following format:

<STX><SEP> data_field_0<SEP> data_field_1<SEP> ….<SEP> data_field_N<SEP><ETX><LRC>

LRC is a checksum. The Description of the Inhova SDK says that for debugging purposes, they replace it with the ODh character. In this version of the SDK, such mode is used to simplify testing, i.e. for the SDK to work, it is enough to transmit OD as a checksum. The same principle is used in the responses.

2.1. Guest registration (check in) command.

If successful, the command writes the guest card. In the event of an error, it returns command codes identical to Inhova.

The SDK maintains a database that contains all the cards that were registered through it. If you try to issue the CheckIn command with the parameters of a guest who has already been registered earlier, the system will respond with an error. Each registered guest must be removed from the database using the CheckOut command.

The following table lists the command data.

Data field No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	<STX>	Station number	CI	Room number (decimal)	Start validity date in dd/mm/YYYY format	Start validity time in HH:mm format	Validity expiration date	Validity expiration time	Not used, "blank" transmitted	Not used	Not used	Not used	Not used	Track1 – not used	Track2 – This is where IronLogic data is transmitted in key-value format (see description after the table)	<ETX><LRC>
Field length, in characters	1	1	2	1-5	9	5	9	5	0	0	0	0	0	0	0 - 1024	2

Data that is not specific to Inhova but necessary for the IronLogic system operation is transmitted in the track2 field.

The transmission format is as follows:

Field_Name1::Field_Value_1,Field_Name2::Field_Value_2,Field_Name3::Field_Value_3,.......

Most of this data is for internal use, however, the following parameters are required for card issuance:

place:: it is necessarily written for guest cards. (By default, 0 is written),

guest_card_version: map version, (see Note 2)

common_doors: permit bitmask. (Transmitted as a decimal number. For example, the number 6 would mean a mask of 00001100, which means an entry to doors 3 and 4)

EmMarine:: The EmMarine value, one of three variants is either the hex-code of the value, the "keep" phrase or the "temic" phrase. (see Note 1)

Note: In track2, if you look at our emulator, it transmits much more parameters including duplicate parameters from cells 1-7. They are optional and are used for internal IronLogic debugging.

Note 1. If the phrase "temic" is transmitted as Em-Marine, then Em-Marine is generated based on the temic card. If the phrase "keep" is transmitted, then the card password and the EmMarine value recorded earlier are not deleted when the card is written or erased.

Note 2. If the card version is not transmitted, then the calculation algorithm is as follows. By default, the card version is 1 if the card with such a number has never been issued. If a card with such a number is found in the history, the validity time of the card is verified. If the validity time of the old card in any way overlaps the validity time of the issued card, the version of the card is incremented by one. You can reset the card's version by explicitly giving the command with the specified version.

You can clearly see all the data of the command if you run the PMS emulator (**IronClientEth**) and click the "Issue the Card" button. In the "protocol debugging" parameter group, all the command batch bytes' values will be displayed.

Attention – all parameters are case-sensitive!

SDK response to the Check In command:

Data field No.	0	1	2	3	4
Description	<STX>	Station number	Completion code (see 3.1)	Empty	<ETX><LRC>
Field length, in characters	1	1	2	0	2

2.2. CheckOut command

When the SDK receives a command, it looks up an entry in the database for the corresponding guest. If such a guest was previously registered, then the guest is removed from the database, and if a card is on the adapter, it is erased. If no such guest was registered, the command returns with an error. The error code is identical to Inhova.

The format of the command is similar to CheckIn.

The following table lists the command data.

Data field No.	0	1	2	3	4	5	6
Description	<STX>	Station number	CO	Room number	Reception (not used)	Track2 – This is where IronLogic data is transmitted in key-value format	<ETX><LRC>
Field length, in characters	1	1	2	1-5	0-128	0 - 1024	2

Note: Track2 is used to enable the transfer of information on how to erase the card: preserving EmMarine or erase it completely.

SDK response to the Check Out command:

Similar to the Check In command.

2.3. CheckInEx command

The command combines the following actions:

1. Verification of the door record history existence with the specified number in the SDK database
2. If there is a door with such a number, the sequence of actions of the CheckOut command is automatically executed.
3. The sequence of actions of the CheckIn command is executed.

Thus, the execution of this command will always end with the issuance of the card (provided that it is present on the adapter), regardless of whether the card with such number was previously registered or not. At the same time, the correctness of the card version installation is ensured for cases when previously issued cards overlap the validity period of the currently issued card.

The following table lists the command data.

Data field No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Descriptio n	<STX>	Station numbe r	CX	Room number (decimal)	Start validity date in dd/mm/YYYY format	Start validity time in HH:mm format	Validity expiration date	Validity expiration time	Not used, "blank" transmitted	Not use d	Not use d	Not use d	Not use d	Track1 – not used	Track2 – This is where IronLogic data is transmitted in key-value format (see description after the table)	<ETX><LRC>
Field length, in characters	1	1	2	1-5	9	5	9	5	0	0	0	0	0	0	0 - 1024	2

The purpose of all of these parameters exactly matches the same parameters of the CheckIn command. The response is also similar to the CheckIn command.

2.4. ClearCard command

The command is not in Inhova and it is added for convenience.

The following table lists the command data.

Data field No.	0	1	2	3	4
Description	<STX>	Station number	CC	Track2 – This is where IronLogic data is transmitted in key-value format	<ETX><LRC>
Field length, in characters	1	1	2	0 - 1024	2

Note: Track2 is used to enable the transfer of information on how to erase the card: preserving EmMarine or erase it completely.

2.5. CopyCard command

Applicable only to guests who had their CheckIn executed previously, otherwise it will return with an error. The format of the command is identical to the format of CheckIn, except for the name of the command.

The following table lists the command data.

Data field No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	<STX>	Station number	CG	Room number	Start validity date in dd/mm/YYYY format	Start validity time in HH:mm format	Validity expiration date	Validity expiration time	Not used	Not used	Not used	Not used	Not used	Track1 – not used	Track2 – This is where IronLogic data is transmitted in key-value format	<ETX><LRC>
Field length (char.)	1	1	2	1-5	9	5	9	5	0	0	0	0	0	0	0 - 1024	2

Note: The SDK stores information about all card pieces for which copies have been made. If you try to make a copy of the card that has already been added to the list of actual copies of cards for the specified door, the card will not be rewritten and the SDK will return an error in the response batch with the code "E9".

2.6. ReadCard command

When receiving the command, the SDK checks for a card on the adapter. If there is a card, the completion code and card data are returned. If there is no card on the adapter, the command returns with an error. The error code is identical to Inhova.

The following table lists the command data.

Data field No.	0	1	2	3
Description	<STX>	Station number	RC	<ETX><LRC>
Field length, in characters	1	1	2	2

The response if a card is detected on the adaptor:

Data field No.	0	1	2	3	4	5
Description	<STX>	Station number	RC	Em-marine code in hex format	Track2 – This is where IronLogic card data is transmitted in key-value format	<ETX><LRC>
Field length, in characters	1	1	2	16	0-1024	2

2.7. EmergencyCard command

Issues an emergency card. The error codes are identical to the CI command. The emergency card is not saved in the SDK database. The card version is used in accordance with the latest current version of the card for this door.

The following table lists the command data.

Data field No.	0	1	2	3	4	5	6	7	8	9
Description	<STX>	Station number	EM	Room number (decimal)	Start validity date in dd/mm/YYYY format	Start validity time in HH:mm format	Validity expiration date	Validity expiration time	Track2 – This is where IronLogic data is transmitted in key-value format (see description after the table)	<ETX><LRC>
Field length, in characters	1	1	2	1-5	9	5	9	5	0 - 1024	2

The purpose of all of these parameters exactly matches the same parameters of the CheckIn command. The response is also similar to the CheckIn command.

3. Appendices:

3.1. Completion codes returned by the SDK

"OK" : "command executed successfully"
"EA" : "communication error"
"E2" : "command format error"
"E1" : "error validating command data"
"E8" : "The card cannot be read or missing"
"EK" : "Adapter key not received"
"DB" : "DataBase Error"
"EF" : "Failed to write the card"
"E3" : "Failed to clear the card"
"**E9**" : "attempted to repeat copying to the same card"

for the CI command:
"ED" : "The card with this room number is already registered"

for the CG command:
"ED" : "The guest with such card was not registered"