SDK Iron Logic Protocol Description

(version SDK: 0.5.3)

1. General information

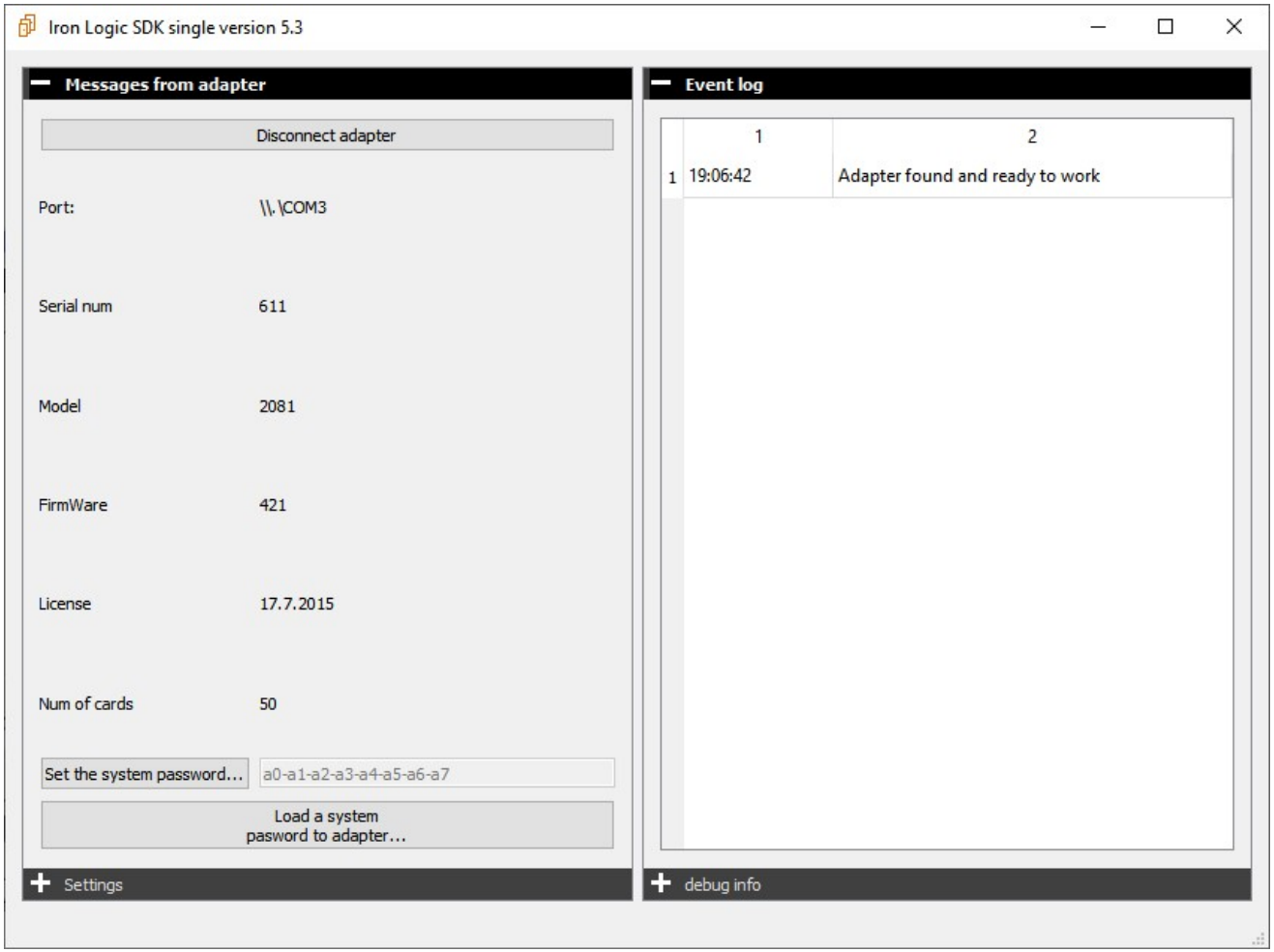
All the files necessary for the SDK are in the archive offered for downloading. One additional file is a simplified PMS emulator we used for testing the passage of commands and their processing by the SDK software.

The SDK can be started via the IronSDK.exe file.

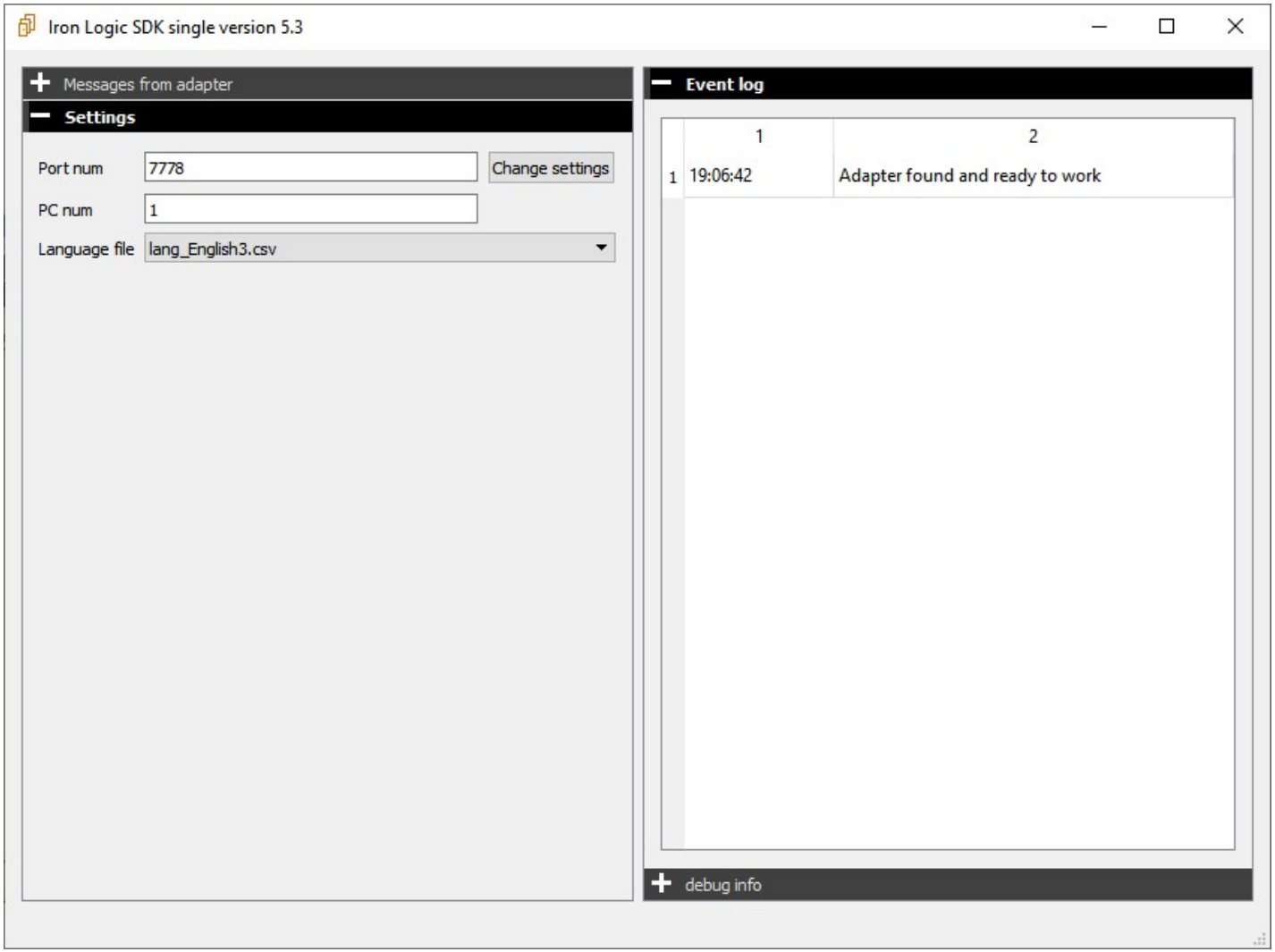
The emulator can be started via the IronClientEth.exe file if you need to check the passage of the commands and their format.

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

Port settings can be changed in the "+ Settings" field group located in the lower-left area of the SDK working display.



In the settings, the required port is specified and the "Reconfigure" command is executed:



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.

An RF-1996 adapter is required to issue cards to IronLogic locks. *The program searches for it at startup*, and it can also be found manually by clicking the "Connect to adapter" button.

When the port is open and the adapter connection is established, you can start working.

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.)

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.

Learn more about command formats.

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
Description	<STX>	Station number	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 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

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. WriteEmergencyCard 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"