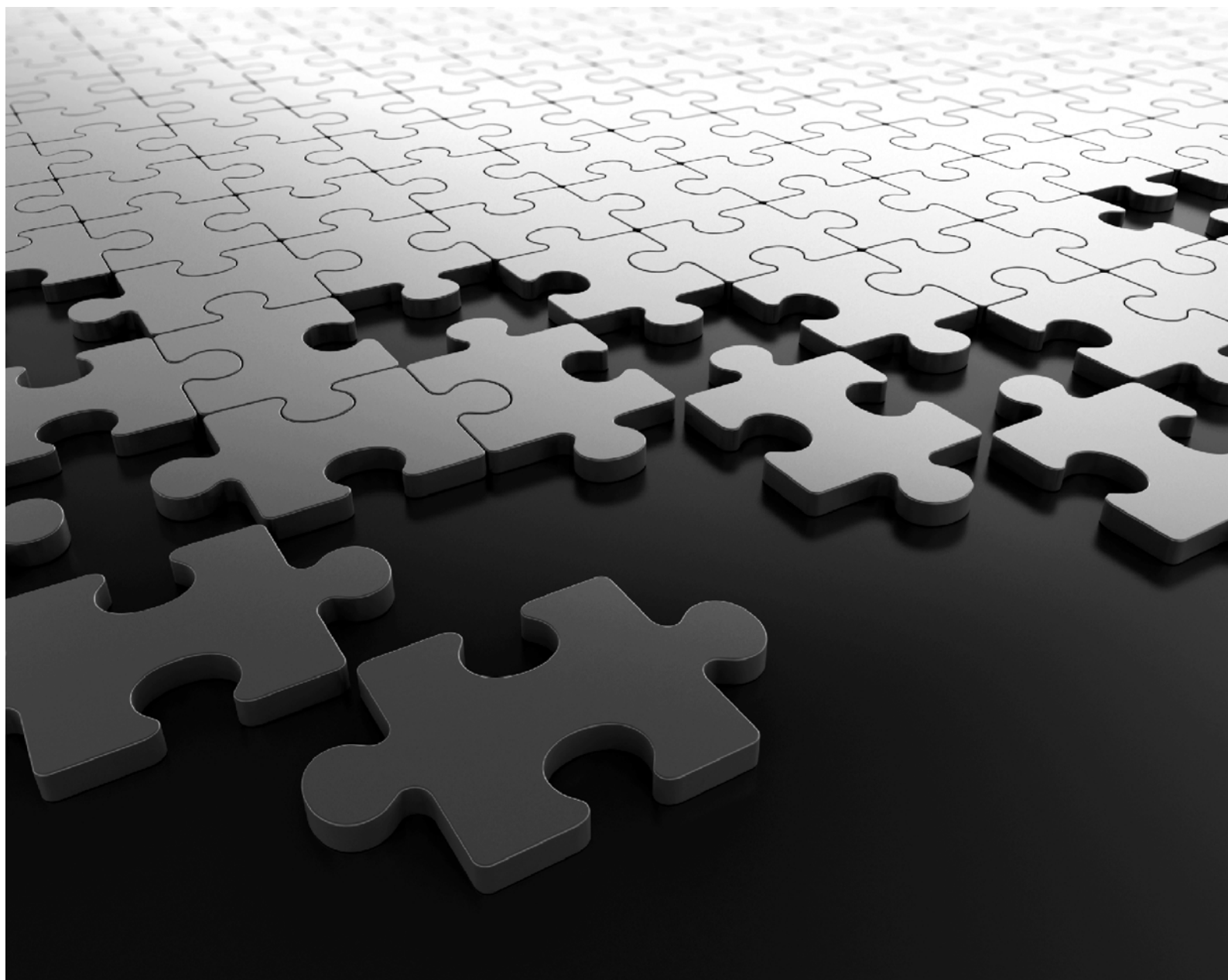


COSEC Devices PUSH API
User Guide



COSEC Devices PUSH API

User Guide



Documentation Disclaimer

Matrix Comsec reserves the right to make changes in the design or components of the product as engineering and manufacturing may warrant. Specifications are subject to change without notice.

This is a general documentation for all variants of the product. The product may not support all the features and facilities described in the documentation.

Information in this documentation may change from time to time. Matrix Comsec reserves the right to revise information in this publication for any reason without prior notice. Matrix Comsec makes no warranties with respect to this documentation and disclaims any implied warranties. While every precaution has been taken in the preparation of this system manual, Matrix Comsec assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

Neither Matrix Comsec nor its affiliates shall be liable to the buyer of this product or third parties for damages, losses, costs or expenses incurred by the buyer or third parties as a result of: accident, misuse or abuse of this product or unauthorized modifications, repairs or alterations to this product or failure to strictly comply with Matrix Comsec operating and maintenance instructions.

Copyright

All rights reserved. No part of this system manual may be copied or reproduced in any form or by any means without the prior written consent of Matrix Comsec.

Version 1

Release date: July 11, 2019

Contents

Contents	i
List of Tables	iii
About the Document	1
API Overview.....	3
Login & Poll.....	11
Login	11
Poll	13
Get Command & Update commands	15
Get Command	15
Update Command	17
Command Specific API	19
1. Enroll Credential	19
2. Delete Credentials	25
3. Get Credentials	27
4. Set Credentials	30
5. Delete All Credentials	33
6. Get Credential Count	35
7. Delete User	37
8. Get User Photo	39
9. Set User Photo	41
10. Clear Alarm	43
11. Acknowledge Alarm	45
12. Lock Door	47
13. Unlock Door	49
14. Normalize Door	51
15. Open Door	53
16. Get Current Event Sequence Number	55
17. Default Device	57
18. Reboot Device	59
19. Activate Aux Relay	61
20. Deactivate Aux Relay	63
21. Firmware Update	65
22. Get User Count	67
Get Configuration and Update Configuration	69
Get Configuration	69
Update Configuration	84
Set Events	85

<i>Set Events API(s)</i>	85
Appendix	87

List of Tables

Table: Login- Request Parameters.....	11
Table: Login- Response Parameters	11
Table: Poll-Request Parameters.....	13
Table: Poll-Response Parameters	13
Table: Get Command- Request Parameters	15
Table: Get Command- Response Parameters	16
Table: Update Command- Request Parameters	17
Table: Update Command- Response Parameters.....	18
Table: Getting Enrollment Credential Command Response.....	19
Table: Updating Enrolled Credential command Request- Parameters.....	22
Table: Updating Enrolled Credential command - Response Fields	24
Table: Deleting Credential Command-Response Fields.....	25
Table: Updating Deleted Credential Command Request- Parameters.....	25
Table: Updating Deleted Credential command - Response Fields	26
Table: Getting Credential Command -Response Fields	27
Table: Updating Get Credential command Request- Parameters	28
Table: Updating Get Credential command - Response Fields	29
Table: Setting Credential command -Response Fields	30
Table: Updating Set Credential command Request- Parameters.....	31
Table: Updating Set Credential command - Response Fields.....	32
Table: Deleting All Credential command -Response Fields	33
Table: Updating Delete All Credential command Request- Parameters.....	33
Table: Updating Delete All Credential command - Response Fields	34
Table: Getting Credential Count Command -Response Fields.....	35
Table: Updating Get Credential Count Command Request- Parameters.....	35
Table: Updating Get Credential command - Response Fields	36
Table: Deleting User Command -Response Fields	37
Table: Updating Delete User Command Request- Parameters.....	37
Table: Updating Delete User Command - Response Fields.....	38
Table: Getting User Photo Command -Response Fields.....	39
Table: Updating Get User Photo Command Request- Parameters.....	39
Table: Updating Get User Photo Command - Response Fields	40
Table: Setting User Photo Command -Response Fields	41
Table: Updating Set User Photo Command Request- Parameters	41
Table: Updating Set User Photo Command - Response Fields	42
Table: Clear Alarm Command -Response Fields	43
Table: Updating Clear Alarm Command Request- Parameters.....	43
Table: Updating Clear Alarm Command - Response Fields.....	44
Table: Acknowledge Alarm Command -Response Fields	45
Table: Updating Acknowledge Alarm Command Request- Parameters.....	45
Table: Updating Acknowledge Alarm Command - Response Fields	46
Table: Lock Door Command -Response Fields.....	47
Table: Updating Lock Door Command Request- Parameters	47
Table: Updating Lock Door Command - Response Fields	48
Table: Unlock Door Command -Response Fields	49
Table: Updating Unlock Door Command Request- Parameters.....	49
Table: Updating Unlock Door Command - Response Fields	50
Table: Getting Normalized Door Command -Response Fields.....	51
Table: Updating Normalized Door Command Request- Parameters.....	51
Table: Updating Normalized Door Command - Response Fields	52
Table: Getting Door Open Command -Response Fields	53
Table: Updating Open Door Command Request- Parameters.....	53
Table: Updating Open Door Command - Response Fields	54
Table: Getting Current Event Sequence Event Command -Response Fields.....	55
Table: Updating Current Event Sequence Number Command Request- Parameters	55

Table: Updating Current Event Sequence Number Command - Response Fields.....	56
Table: Getting Default Device Command -Response Fields	57
Table: Updating Default Device Command Request- Parameters	57
Table: Updating Default Device Command - Response Fields	58
Table: Getting Reboot Device Command -Response Fields	59
Table: Updating Reboot Device Command Request- Parameters	59
Table: Updating Reboot Device Command - Response Fields	60
Table: Getting Aux Relay Activate Command -Response Fields	61
Table: Updating Aux Relay Activate Command Request- Parameters	61
Table: Updating Aux Relay Activate Command - Response Fields.....	62
Table: Getting Aux Relay Deactivate Command -Response Fields	63
Table: Updating Aux Relay Deactivate Command Request- Parameters	63
Table: Updating Aux Relay Deactivate Command - Response Fields	64
Table: Getting Firmware Update Command -Response Fields	65
Table: Updating Firmware Update Command Request- Parameters	65
Table: Updating Firmware Update Command - Response Fields	66
Table: Getting User Count Command -Response Fields	67
Table: Updating Get User Count Command Request- Parameters.....	67
Table: Updating Get User Count Command - Response Fields.....	68
Table: Get Configuration -Request Parameters	69
Table: Get Configuration -date time Response Fields.....	70
Table: Get Configuration -device basic configuration Response Fields	72
Table: Get Configuration -function key configuration Response Fields.....	74
Table: Get Configuration -Enrollment option Response Fields.....	74
Table: Get Configuration -Reader Parameter of Direct Doors except COSEC ARC Response Fields	75
Table: Get Configuration -Reader Parameter of COSEC ARC DC 200 Response Fields	76
Table: Get Configuration -Finger Reader Parameter Response Fields.....	78
Table: Get Configuration -Palm Reader Parameter Response Fields.....	79
Table: Get Configuration -Door Feature Response Fields	80
Table: Get Configuration -System Timers Response Fields.....	81
Table: Get Configuration -System Timers Response Fields.....	81
Table: Get Configuration -FTP Server Response Fields	83
Table: Update Configuration -Request Parameters.....	84
Table: Update Configuration - Response Fields.....	84
Table: Set Event- Request Parameters	85
Table: Set Event- Response Parameters	86
Table: Universal Time Zone Reference	87
Table: List of Events	88
Table: Size of Event Fields	89
Table: User Events	89
Table: Special Function Codes Reference	90
Table: Information of Bit 0 and Bit 1	91
Table: Information of Bit 4 and Bit 9	91
Table: Field 3 Detail (User Events) Reference	91
Table: Door Events	92
Table: System Events.....	93

About the Document

Welcome to the *COSEC Devices PUSH API User Guide*. This document will provide you a comprehensive overview and complete user-guidance for all *COSEC Devices PUSH APIs*. You can learn more about COSEC Device PUSH APIs, browse through detailed descriptions of individual APIs and test them using sample scenarios.

Document Conventions

This API User Guide will follow a set of document conventions to make it consistent and easier for you to read. These are as follows:

1. Text within angle brackets (e.g. “<request-type>”) denotes content in URL syntax and should be replaced with either a value or a string. The angle brackets should be omitted in all instances except those used to denote “tags” within XML responses (e.g. “<name></name>”).
2. Cross-references and other links appear as follows: [Document Conventions](#)

For e.g. To learn more about APIs, please refer to section [Who Can Use This Document](#)
3. The term *device* used in this document, will refer only to direct doors.
4. Any expression resembling <x~y>, indicates that the field should be repeated for index values x to index values y. This is to avoid duplicating the same parameter for multiple index numbers.
5. Additional information about any section appears in the form of notices. The following symbols have been used for notices to draw your attention to important items.



Important: to indicate something that requires your special attention or to remind you of something you might need to do when you are using the system.



Caution: to indicate an action or condition that is likely to result in malfunction or damage to the system or your property.



Warning: to indicate a hazard or an action that will cause damage to the system and or cause bodily harm to the user.



Tip: to indicate a helpful hint giving you an alternative way to operate the system or carry out a procedure, or use a feature more efficiently.

Document Organization

This document has been organized into the following topics:

1. About the Document
2. Introduction
3. Login and Poll
4. Get and Update command
5. Get and Update Configuration
6. Set Events
7. Appendix

Topics 1 and 2 will provide a general understanding of COSEC Devices PUSH APIs and the basic interface communication. Topic 3 provides information about Login and Poll API. Topics 4 provide an information regarding Get and Update command API categories with detailed explanation of individual commands. Topic 5 provides explanation regarding Get and Update configuration APIs, while Event APIs are mentioned in Topic 6. The following information has been provided on each request type:

- Description of the functionality.
- Action requested.
- Generic query syntax.
- Mandatory and optional parameters (argument-value table).
- Examples (*Sample Request* and *Sample Response*).

The *Appendix* will provide additional material for the user's reference.



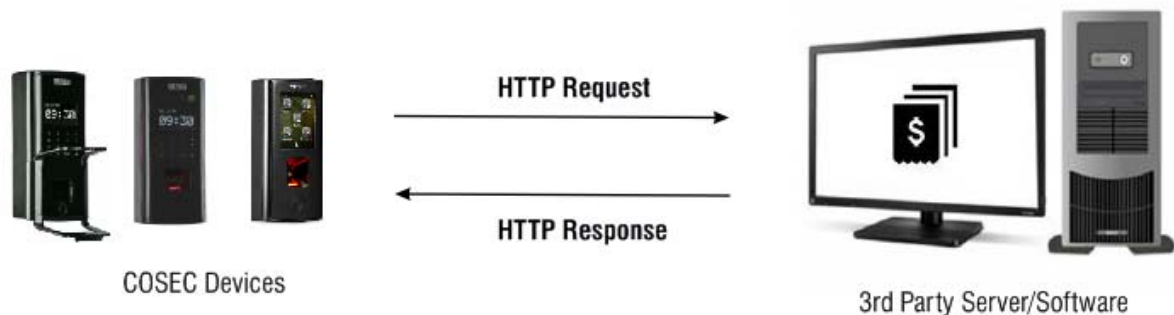
For a list of all tables provided in the document, refer to [List of Tables](#). Click on the links to view the respective tables for the required data.

Who Can Use This Document

The COSEC Devices PUSH API User Guide is meant for *third-party software developers* who wish to operate COSEC Devices via another remote server/software. This guide will provide information to users on how to request and receive services from COSEC Devices using a COSEC PUSH API.

COSEC Device PUSH API provides the solution for integrating Matrix devices to 3rd party server/software for Time and Attendance purpose. To fulfill this, COSEC Devices uses HTTP(s) PUSH API support, which helps devices to communicate directly with 3rd party server/software. This extends support to cover basic functionalities like - User configuration, device configuration, event syncing, credential handling and commands etc.

How It Works



By the support of HTTP(s) PUSH API, device will be communicating with 3rd Party Server/Software. For establishing the connection, open the device web page. On device web page and display screen there is a provision to change the device mode to Third party communication.

By following these below given steps, device mode can be changed to Third Party:

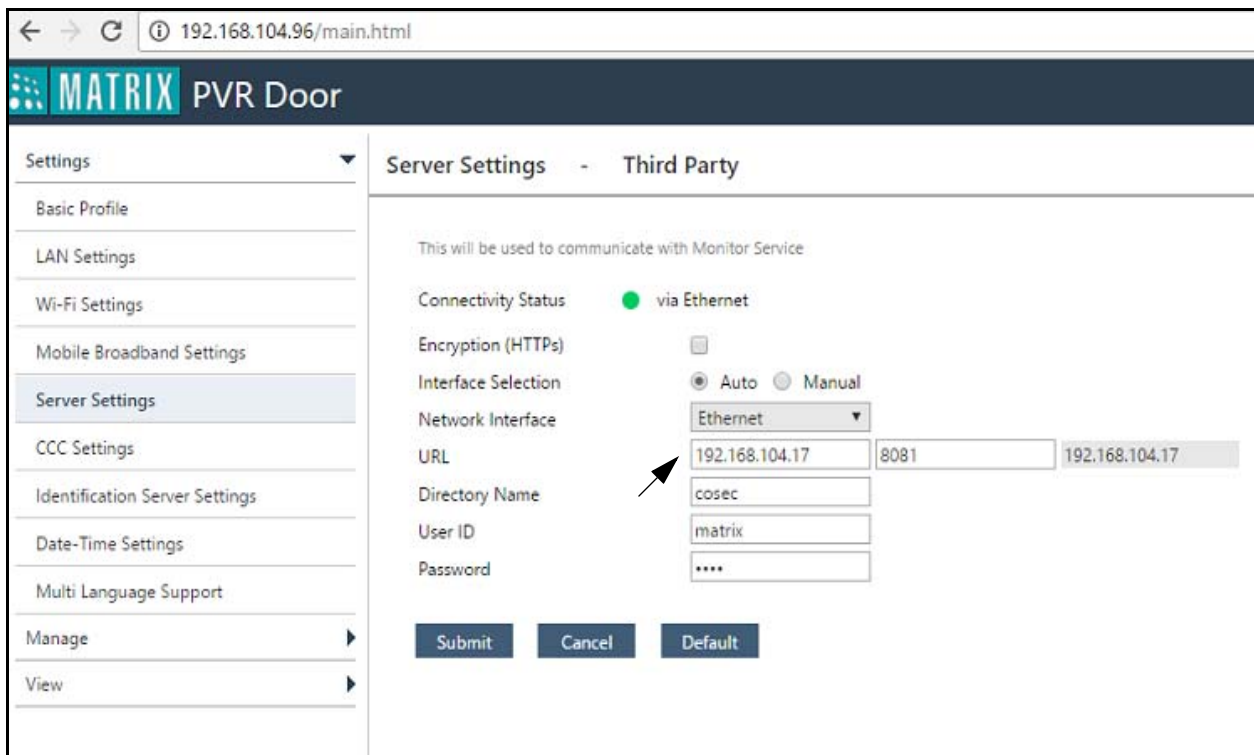
- Enter the IP address of device in web browser to open the device web page. In case if IP address is not yet configured, then default IP address is 192.168.50.1
- Enter User-name and password. By default User-name is 'admin' and password is user configurable.



Refer Quick start for configuring particular device.

- Go to **Settings > Basic Profile > Server Connection.**
- Choose Third Party Server from the drop-down list from Server Connection.

- On changing the mode, Device will be rebooted and device connectivity status will be shown as off-line. Now third party server URL, port, User ID, Password, Encryption can be configured from web page of server settings as shown below:



- From the display screen of device, only limited functionalities can be configured such as updating server URL and port.

Supported Devices

Currently, COSEC Device PUSH APIs are supported on the following COSEC Direct Door Controllers only:

- COSEC Door V3
- COSEC PVR Door
- COSEC Vega Controller
- COSEC ARC DC 200
- COSEC Door FMX
- COSEC ARGO

What the User Should Know

It is assumed that developers using this document have prior knowledge of:

- Basic functioning of the COSEC Devices
- Basic HTTP request-response communication
- XML

Prerequisite

In order to use a COSEC PUSH API, the user will require:

- A COSEC Device (pre-installed)
- 3rd Party Server/Software
- Communication must be established between device and server.



For information on installing a COSEC device and assigning an IP address to it, please refer to the respective device documentation.

HTTP Request-Response

Basic HTTP communication is based on a request-response paradigm. The message structure for both request and response has a generic format.

HTTP-message = Request Response ; HTTP/1.1 messages

Generic-message = start-line	<i>The start line</i>
*(message-header CRLF)	<i>Zero or more header fields or 'headers'</i>
CRLF	<i>An empty line</i>
[Message-body]	<i>A message-body (chunk or payload)</i>

Start-line = Request-Line Status-Line

Request Format

All HTTP Requests follow a generic message format. It consists of the following components:

1.	Request Line	<p>This line is constituted by the following three elements which must be separated by a space:</p> <ul style="list-style-type: none"> • The method type (GET, HEAD, POST, PUT etc.) • The requested URL • The HTTP version to use <p>For e.g.:</p> <p>GET http://192.168.1.2/device.cgi/command?action=geteventcount HTTP/1.0</p>
----	--------------	---

2.	Header Fields	<p>Add information about the request using these header fields:</p> <ul style="list-style-type: none"> • A General Header (<Header-name>:<value>). • A Request Header (<Header-name>:<value>). • An Entity Header (<Header-name>:<value>).
3	Empty Line	This is an empty line separating headers from the message body.
4	Message Body	This is the chunk or payload.

Example:

```

POST /login?device-type=1&serial-no=123456 HTTP/1.1\r\n
[Expert Info (Chat/Sequence): POST /login?device-type=1&serial-no=123456 HTTP/1.1\r\n]
Request Method: POST
Request URI: /login?device-type=1&serial-no=123456
Request Version: HTTP/1.1

```

Response Format

An HTTP response is a collection of lines sent by the server to the client. A generic HTTP response format will resemble the following:

```

VERSION-HTTP CODE EXPLANATION<cr1f>
HEADER: Value<cr1f>
.
.
.
HEADER: Value<cr1f>
Empty line<cr1f>
BODY OF THE RESPONSE

```

It consists of the following components:

1.	A status line	<p>This line is constituted by the following three elements which must be separated by a space:</p> <ul style="list-style-type: none"> • The version of the protocol used (e.g. <i>HTTP/1.0</i>). • The status code (indicates the status of the request being processed). • The explanation of the code.
2.	The response header fields	These optional lines allow additional information to be added to the response header. This information appears in the form of a name indicating the header type followed by a value for the header type. The name and value are separated by a colon (:).
3.	The body of the response	Contains the requested data.

Example

When the server gets a request, it will respond with a standard HTTP status code as illustrated in the following sample response:

```
HTTP/1.0 200 OK\r\n
  [Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
  Request Version: HTTP/1.0
  Status Code: 200
  Response Phrase: OK
  Date: Tue May 21 04:59:32 2019\r\n
  Server: Server: GoAhead-http\r\n
  Pragma: no-cache\r\n
  Cache-Control: no-cache\r\n
  Content-Type: text/html\r\n
  File Data: 205 bytes

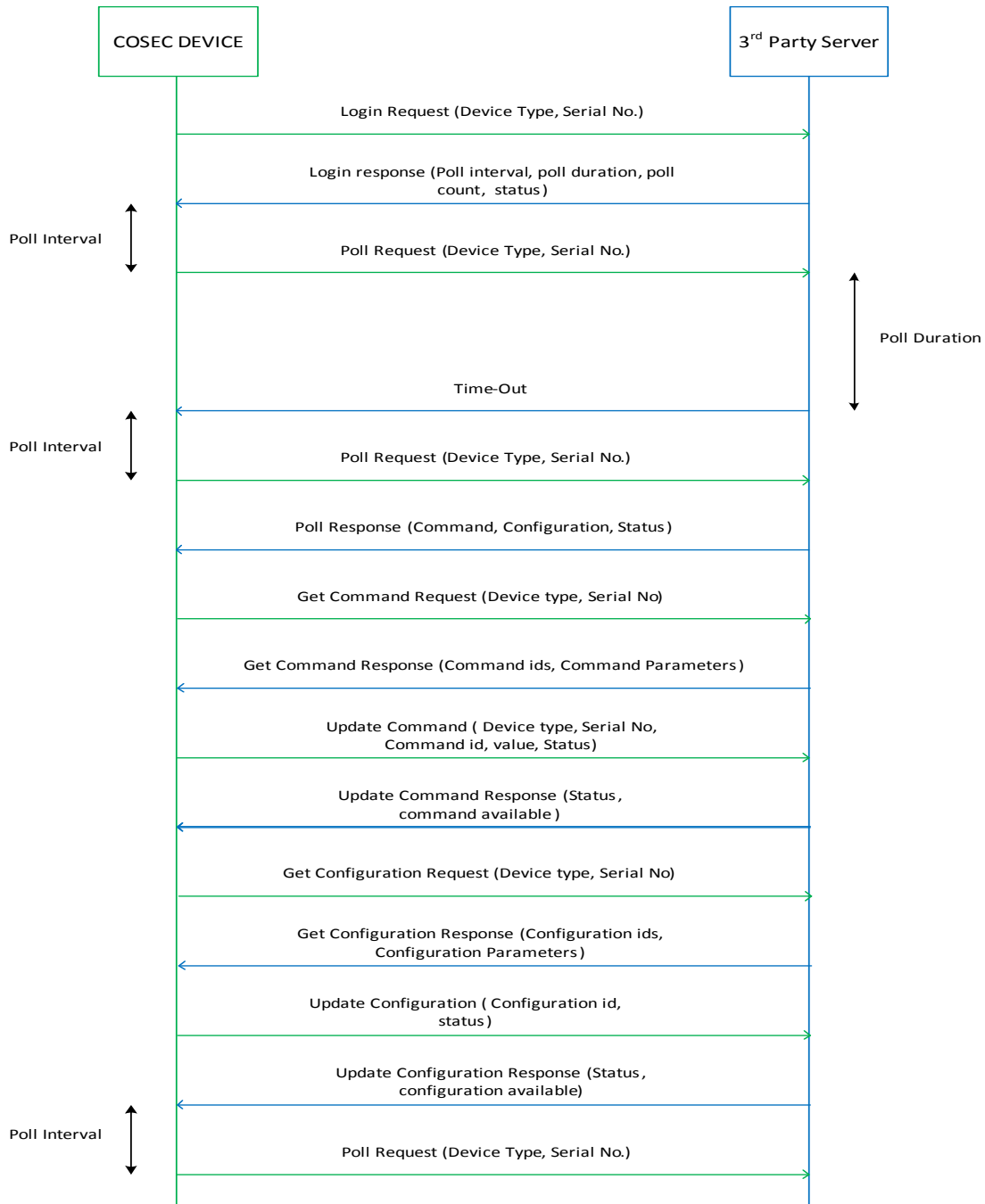
Line-based text data: text/html
  poll-duration=5
  poll-interval=5
  poll-count=3
  status=1
  format=0
```



HTTP Status Codes: Status codes are 3-digit numeric codes returned in HTTP responses that enable recipients to understand the successful or failed status of the request issued. In general, codes in the 1xx range indicate an informational message only, 2xx codes indicate a successful request, 3xx codes indicate an incomplete request that requires further action, 4xx codes point at client-side errors while 5xx codes point at server-side errors.

Process:

On saving the configuration, the device will follow the process (typically Client- Server Based) as shown in the figure below:



- On successful server configuration Save, Device will try to perform login on provided Server URL/Port.
- On successful Login response, Device will start polling.
- Poll Interval is the time between each poll request.
- Poll Duration is the time for which device will wait for server response of poll request.
- Poll count is the number of consecutive poll attempts done by device without poll response to declare device dis-connectivity.

- Device Online-Offline status will be derived from this API flow and connectivity status will be reflected on device display screens as well as web pages.
- If time equivalent to poll duration has elapsed waiting for response of any request API then response time-out will be declared and the device will send poll request again to server.
- Device type and device serial number will be sent in each device request so that server can identify device. here serial number will be MAC address of device without colon(:) separation.
- Upon poll request if both command and configuration are available then get command request will be issued first by the device.
- In whole process of commands execution buzzer and led cadence will be as existing in each functionality.
- When device will initiate Get Command API, then in response the command id and its relevant parameters must be sent by server.
- The device will then execute the said command.
- While the command is being executed on device, in mean time device will not do any communication with server.
- After execution success/failure the device will initiate Update Command API and will wait for its response.
- Configuration will work in same flow as command.

URL Syntax

All COSEC APIs follow a common HTTP query syntax for the device to generate a request. The generic URL is stated below.

Syntax

```
http://<ServerURL>/APIName?<argument>=<value> [&<argument>=<value>... ]
```

Take a close look at the URL and its basic elements:

URL element	Description
<i>http://</i>	This is the protocol used to communicate with the client. Note: All HTTP commands are in plain text, and almost all HTTP requests are sent using TCP port 80, though any port can be used.
<i><ServerURL></i>	URL of the server communicating with the device.
<i>APIName</i>	Name of API.
<i><argument></i>	This defines a specific action or command depending on the function to be performed.
<i><value></i>	These are argument values that determine the output.

Example

```
http://GoAhead.com/Login?device-type=1&serial-no=123456
```


Login

Login is used to set the login parameters of a device on provided server URL/port.

Syntax

```
http://<ServerURL>/login?device-type=<value>&serial-no=<value>
```

Parameters

Table: Login- Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0, 1, 2, 3, 5, 7 0 = Door V3 1 = PVR 2 = Vega 3 = FMX 5 = ARC DC 200 7 = ARGO	Yes	To specify type of device/devices whose list is to be fetched. (only one value at a time)
serial-no	Alphanumeric, Max. 12 characters	Yes	To specify the serial number of the device.

Table: Login- Response Parameters

Parameter	Valid Values	Description
poll-interval	5-60sec	Defines the poll interval (in seconds).
poll-duration	3-30sec	Defines the poll interval (in seconds).
poll-count	1-5	Defines the polling count before declaring that the device as offline
status	0,1 0 = Failure 1 = Successful	Defines the status for the acknowledgment
format	0 & 1 0 = text 1 = xml	Defines the API response format so further, all API responses from server will be in provided format. The device will remember this value and parse accordingly. Also, the expected response is to be in text format for login.

Example in text

```
poll-interval=5 poll-duration=5 poll-count=3 status=1 format=0
```

Poll

After performing Login, the device will start polling. The values for polling will be received from server side on login request.

Syntax

```
http://<ServerURL>/poll?device-type=<value>&serial-no=<value>
```

Parameters

Table: Poll-Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0, 1, 2, 3, 5, 7 0 = Door V3 1 = PVR 2 = Vega 3 = FMX 5 = ARC DC 200 7 = ARGO	Yes	To specify type of device/devices whose list is to be fetched. (only one value at a time)
serial-no	Alphanumeric, Max. 12 characters	Yes	To specify the serial number of the device.

Table: Poll-Response Parameters

Parameter	Valid Values	Description
cmd-avlbl	0 & 1 0 = New commands are not available 1 = New commands are available	Defines whether any command is available to be issued with server for device
cnfg-avlbl	0 & 1 0 = New configurations are not available 1 = New configurations are available	Defines whether any configuration is available to be issued with server for device
status	0,1 0 = Failure 1 = Successful	Defines the status for the acknowledgment

Example in text

```
cmd-avlbl=1 cnfg-avlbl=1 status=1
```

Example in XML

```
<api>  
<cmd-avlbl>1</cmd-avlbl>  
<cnfg-avlbl>1</cnfg-avlbl>  
<status>1</status>  
</api>
```


Get Command & Update commands

Get Command

Device will initiate 'get command' API request and send to server with device id and serial number as parameters. In response command id and required parameters will be send from the server side. The same command will be executed by the device. If device receives any command where any mandatory parameter is not provided, then device will fail the command execution.

Syntax

```
http://< ServerURL >/getcmd?device-type=<value>&serial-no=<value>
```

Parameters

Table: Get Command- Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0, 1, 2, 3, 5, 7 0 = Door V3 1 = PVR 2 = Vega 3 = FMX 5 = ARC DC 200 7 = ARGO	Yes	To specify type of device/devices whose list is to be fetched. (only one value at a time)
serial-no	Alphanumeric, Max. 12 characters	Yes	To specify the serial number of the device.

Table: Get Command- Response Parameters

Parameter	Valid Values	Mandatory	Description
cmd-id	1 to 22 1 = Enroll Credential 2 = Delete Credential 3 = Get Credential 4 = Set Credential 5 = Delete all Credentials 6 = Get Credential Count 7 = Delete User 8 = Get User Photo 9 = Set User Photo 10 = Clear Alarm 11 = Acknowledge Alarm 12 = Lock Door 13 = Unlock Door 14 = Normalize Door 15 = Open Door 16 = Get current Event sequence number 17 = Default Device 18 = Reboot Device 19 = Activate Aux Relay 20 = Deactivate Aux Relay 21 = Firmware Upgrade 22 = Get User Count	Yes	Command ID specific to Command
Command Parameter 1	-	-	-
Command Parameter n	-	-	-



The values of cmd-id are described individually. See [“Command Specific API” on page 19.](#)

Update Command

The Device initiates update command API where it update the status of command executed in terms of Failure/ Success and waits for the response from server side.

Syntax

```
http://< ServerURL >/updatecmd?device-type=<value>&serial-no=<value>&status=<value>&cmd-id=<value>&<argument>=<value> [&<argument>=<value>.....]
```

Parameters

Table: Update Command- Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0, 1, 2, 3, 5, 7 0 = Door V3 1 = PVR 2 = Vega 3 = FMX 5 = ARC DC 200 7 = ARGO	Yes	To specify type of device/devices whose list is to be fetched. (only one value at a time)
serial-no	Alphanumeric, Max. 12 characters	Yes	To specify the serial number of the device.
status	0 & 1 0 = Failure 1 = Successful	Yes	Status of command execution
cmd-id	1 to 22 1 = Enroll Credential 2 = Delete Credential 3 = Get Credential 4 = Set Credential 5 = Delete all Credentials 6 = Get Credential Count 7 = Delete User 8 = Get User Photo 9 = Set User Photo 10 = Clear Alarm 11 = Acknowledge Alarm 12 = Lock Door 13 = Unlock Door 14 = Normalize Door 15 = Open Door 16 = Get current Event sequence number 17 = Default Device 18 = Reboot Device 19 = Activate Aux Relay 20 = Deactivate Aux Relay 21 = Firmware Upgrade 22 = Get User Count	Yes	Command ID
Command Parameter 1	-	-	-
Command Parameter n	-	-	-

Table: Update Command- Response Parameters

Parameter	Valid Values	Mandatory	Description
cmd-avlbl	0 & 1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device
status	0 & 1 0 = Failure 1 = Successful	Yes	Status of Acknowledgment

Example in text

```
cmd-avlbl=1 status=1
```

Example in XML

```
<api>  
<cmd-avlbl>1</cmd-avlbl>  
<status>1</status>  
</api>
```



The values of cmd-id are described individually. See [“Command Specific API” on page 19.](#)

Command Specific API

1. Enroll Credential

• *Getting Enrollment Credential*

Enroll Credential is used to enroll credentials like card, finger-print, palm, etc. Single or Multiple count credential can be provided for specific user in command. If the count is not specified for enroll command then, by default it will be consider as one and enrollment operation will be performed. For Card enrollment, Device will do enrollment against user for Card1/Card 2 as per provided input. Similarly for biometric enrollment, Device will do enrollment of user as per command input.

Parameters

Same as [Get Command](#) See “[Get Command](#)” on page 15.

Response Fields

Table: Getting Enrollment Credential Command Response

Argument	Valid Values	Mandatory	Description
cmd Id	1	Yes	Command ID
cred-type	1 to 4 1 = Read Only Card 2 = Smart Card 3 = Finger 4 = Palm	Yes	Type of credential to be enrolled
user-id	15 characters	Yes	Alphanumeric user id of user for which credential is to be enrolled
finger-no	1 to 10 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	Specifies the number of fingers to be enrolled if credential type provided is finger Default =1 if not provided
card=no	1,2 1 = 1 Card 2 = 2 Card	No	Specifies the number of cards to be enrolled if credential type provided is read only /smart card Default =1 if not provided

Table: Getting Enrollment Credential Command Response

Argument	Valid Values	Mandatory	Description
palm-no	1 to 10 1 = 1 Palm 2 = 2 Palm 3 = 3 Palm 4 = 4 Palm 5 = 5 Palm 6 = 6 Palm 7 = 7 Palm 8 = 8 Palm 9 = 9 Palm 10=10 Palm	No	Specifies the number of palms to be enrolled if credential type provided is palm Default =1 if not provided

Table: Getting Enrollment Credential Command Response

Argument	Valid Values	Mandatory	Description
w-ref-user-id	0,1 0 = Inactive 1 = Active	No	When credential type is Smart Card , all mentioned parameters will be written on card as currently configured smart card format
w-name	0,1 0 = Inactive 1 = Active	No	
w-asc	0,1 0 = Inactive 1 = Active	No	
w-fc	0,1 0 = Inactive 1 = Active	No	
w-designation	0,1 0 = Inactive 1 = Active	No	
w-branch	0,1 0 = Inactive 1 = Active	No	
w-department	0,1 0 = Inactive 1 = Active	No	
w-bg	0,1 0 = Inactive 1 = Active	No	
w-contact	0,1 0 = Inactive 1 = Active	No	
w-medical-history	0,1 0 = Inactive 1 = Active	No	
w-fp-template	0,1,2 0 = No Template 1 = 1 Finger Template 2 = 2 Finger Template	No	
name	15 Chars, ASCII Code	No	
designation	15 chars, ASCII Code	No	
branch	15 chars, ASCII Code	No	
department	15 chars, ASCII Code	No	

Table: Getting Enrollment Credential Command Response

Argument	Valid Values	Mandatory	Description
bg	A+ A- B+ B- AB+ AB- O+ O- A1- A1+ A1B- A1B+ A2- A2+ A2B- A2B+ B1+	No	When credential type is Smart Card, all mentioned parameters should be written on card as currently configured smart card format
contact	15 chars, ASCII Code	No	
medical-history	15 chars, ASCII Code	No	

Example in text

```
cmd-id = 1 cred-type=3 user-id=1234 finger-no=1
```

Example in XML

```
<api>
<cmd-id >1</cmd-id>
<cred-type>3</cred-type>
<user-id>1234</user-id>
<finger-no>1</finger-no>
</api>
```

• Updating Enrolled Credential

After the 'get' command for enrollment of user, the device will update enroll credentials by giving values directly (for card) or by giving biometric credential's actual enrolled count in finger number/ palm number (for biometric credential).

Parameters

Table: Updating Enrolled Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	1	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type

Table: Updating Enrolled Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
finger-no	1 to 10 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	Count of finger enrolled
palm-no	1 to 10 1 - 1 Palm 2 - 2 Palm 3 - 3 Palm 4 - 4 Palm 5 - 5 Palm 6 - 6 Palm 7 - 7 Palm 8 - 8 Palm 9 - 9 Palm 10-10 Palm	No	Count of Palm enrolled
card-1	-	No	card 1 Value
card-2	-	No	card 2 Value
data-1	-	No	Finger /Palm template data
data-2	-	No	Finger /Palm template data
data-3	-	No	Finger /Palm template data
data-4	-	No	Finger /Palm template data
data-5	-	No	Finger /Palm template data
data-6	-	No	Finger /Palm template data
data-7	-	No	Finger /Palm template data
data-8	-	No	Finger /Palm template data
data-9	-	No	Finger /Palm template data
data-10	-	No	Finger /Palm template data

Response Fields

Table: Updating Enrolled Credential command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device



For enrollment, it is mandatory for user to give proper input aligned with command parameters or else operation will fail.

2. Delete Credentials

- **Deleting Credential**

Delete Credential is used to delete specific type of credential for specific user.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Deleting Credential Command-Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	2	Yes	Command ID
cred-type	1,2,3 1 = Card 2 = Finger 3 = Palm	Yes	Define the type of credential to be deleted
user-id	15 characters	Yeas	Defines the alphanumeric user id of the user to delete credential

Example in text

```
HTTP 200 OK  
cmd-id=3 user-id=1234 cred-type=3
```

Example in XML

```
<api>  
<cmd-id >2</cmd-id>  
<cred-type>3</cred-type>  
<user-id>1234</user-id>  
</api>
```

- **Updating Deleted Credential**

Device will acknowledge (Update) success in status if it will delete credential as per the input as per user, else failure will be updated in status.

Parameters

Table: Updating Deleted Credential Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	2	Yes	Command ID

Table: Updating Deleted Credential Command Request- Parameters

Argument	Valid Values	Mandatory	Description
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Deleted Credential command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

3. Get Credentials

- **Getting Credential**

Get Credential is used to get credential data of specific credential type for specific user. (*from device to server*)

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Credential Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	3	Yes	Command ID
cred-type	1,2,3 1 = Card 2 = Finger 3 = Palm	Yes	Defines the type of credential
user-id	15 characters	Yes	Defines the alphanumeric user id of the user to get credential data
finger-no	0 to 10 0 = All 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	Defines the index of finger for which template data is required (if not specified consider as 0=all)
card-no	0,1,2 0 = All 1 = 1 Card 2 = 2 Card	No	Defines the index of card for which data is required (if not specified consider as 0=all)
palm-no	0 to 10 0 = All 1 = 1 Palm 2 = 2 Palm 3 = 3 Palm 4 = 4 Palm 5 = 5 Palm 6 = 6 Palm 7 = 7 Palm 8 = 8 Palm 9 = 9 Palm 10=10 Palm	No	Defines the index of palm for which template data is required (if not specified consider as 0=all)

Example in text

```
cmd-id = 3 user-id = 1234 cred-type=1 card-no=2
```

Example in XML

```
<api>
<cmd-id >3</cmd-id>
<cred-type>1</cred-type>
<user-id>1234</user-id>
<card-no>2</card-no>
</api>
```

• **Updating Get Credential**

After the 'get' command, the device will update specific credential type for specific user. If no template is available at the requested index against the user then failure will be sent in update command request.

Parameters

Table: Updating Get Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	3	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
finger-no	1 to 10 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	Count of finger templates provided in data part (if only one template is requested, finger-no=1 is sent, else if all are requested then the count for available templates are sent).
palm-no	1 to 10 1 - 1 Palm 2 - 2 Palm 3 - 3 Palm 4 - 4 Palm 5 - 5 Palm 6 - 6 Palm 7 - 7 Palm 8 - 8 Palm 9 - 9 Palm 10-10 Palm	No	Count of palm templates provided in data part (if only one template is requested palm-no=1 is sent, else if all are requested then the count for available templates are sent)
card-1	-	No	card 1 Value

Table: Updating Get Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
card-2	-	No	card 2 Value
data-1	-	No	Finger /Palm template data
data-2	-	No	Finger /Palm template data
data-3	-	No	Finger /Palm template data
data-4	-	No	Finger /Palm template data
data-5	-	No	Finger /Palm template data
data-6	-	No	Finger /Palm template data
data-7	-	No	Finger /Palm template data
data-8	-	No	Finger /Palm template data
data-9	-	No	Finger /Palm template data
data-10	-	No	Finger /Palm template data

Response Fields**Table: Updating Get Credential command - Response Fields**

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

4. Set Credentials

- **Setting Credential**

Set Credential is used to set credential data of specific credential type for specific user. (from server to device)

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Setting Credential command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	4	Yes	Command ID
cred-type	1,2,3 1 = Card 2 = Finger 3 = Palm	Yes	Define type of credential
user-id	15 characters	Yes	Defines the alphanumeric user id of the user.
finger-no	1 to 10 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	To specify the index of fingers. Default =1 if not provided
palm-no	1 to 10 1 - 1 Palm 2 - 2 Palm 3 - 3 Palm 4 - 4 Palm 5 - 5 Palm 6 - 6 Palm 7 - 7 Palm 8 - 8 Palm 9 - 9 Palm 10-10 Palm	No	To specify the index of palms. Default =1 if not provided
card-1	-	No	card 1 Value
card-2	-	No	card 2 Value
data-1	-	No	To send the templates
data-2	-	No	To send the templates
data-3	-	No	To send the templates
data-4	-	No	To send the templates
data-5	-	No	To send the templates

Table: Setting Credential command -Response Fields

Argument	Valid Values	Mandatory	Description
data-6	-	No	To send the templates
data-7	-	No	To send the templates
data-8	-	No	To send the templates
data-9	-	No	To send the templates
data-10	-	No	To send the templates

Example in text

```
cmd-id = 4 user-id = 1234 cred-type=1 card1=123456
```

Example in XML

```
<api>
<cmd-id >4</cmd-id>
<cred-type>1</cred-type>
<user-id>1234</user-id>
<card1>123456<card1>
</api>
```

• **Updating Set Credential**

After the 'set' command, the device will update specific credential type for specific user. If no template is available at the requested index against the user then failure will be sent in update command request.

Parameters

Table: Updating Set Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	4	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Set Credential command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

5. Delete All Credentials

- **Deleting All Credential**

Delete All Credential is used to delete specific type for credential for all users.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Deleting All Credential command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	5	Yes	Command ID
cred-type	1,2,3 1 = Card 2 = Finger 3 = Palm	Yes	Define type of credential to be deleted

Example in text

```
cmd-id = 5 cred-type=1
```

Example in XML

```
<api>  
<cmd-id >5</cmd-id>  
<cred-type>1</cred-type>  
</api>
```

- **Updating Delete All Credential**

After the 'Delete All' command, the device will update specific credential type for all user. Device will acknowledge success in status if it will delete credential as per input for user else failure will be given.

Parameters

Table: Updating Delete All Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	5	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device

Table: Updating Delete All Credential command Request- Parameters

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Delete All Credential command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

6. Get Credential Count

- **Getting Credential Count**

Get Credential Count is used to get the specific type of credential count against user.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Credential Count Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	6	Yes	Command ID
cred-type	1,2,3 1 = Card 2 = Finger 3 = Palm	Yes	Define type of credential
user-id	15 characters	Yes	Defines the alphanumeric user id of the user.

Example in text

```
cmd-id = 6 cred-type=1 user-id = 1234
```

Example in XML

```
<api>  
<cmd-id >6</cmd-id>  
<cred-type>1</cred-type>  
<user-id>1234</user-id>  
</api>
```

- **Update Get Credential Count**

After the 'get credential count' command, the device will update credential count for specific user. If no count is available against the user then failure will be sent in update command request.

Parameters

Table: Updating Get Credential Count Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	6	Yes	Command ID

Table: Updating Get Credential Count Command Request- Parameters

Argument	Valid Values	Mandatory	Description
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
Cred-count-palm	0-10	No	Will show the count of enrolled palm templates
Cred-count-finger	0-10	No	Will show the count of enrolled finger templates
Cred-count-card	0-2	No	Will show the count of enrolled cards

Response Fields**Table: Updating Get Credential command - Response Fields**

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

7. Delete User

• *Deleting User*

Delete user is used to delete user from device.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Deleting User Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	7	Yes	Command ID
user-id	15 characters	Yes	Defines the alphanumeric user id of the user.

Example in text

```
cmd-id = 7 user-id = 1234
```

Example in XML

```
<api>  
<cmd-id >7</cmd-id>  
<user-id>1234</user-id>  
</api>
```

• *Updating Delete User*

After the 'Delete user' command is used, Device will acknowledge success in status if user is deleted as per input else failure will be given.

Parameters

Table: Updating Delete User Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	7	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device

Table: Updating Delete User Command Request- Parameters

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Delete User Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

8. Get User Photo

- **Getting User Photo**

Get User Photo is used to get user photo from device. User Photo size should be max 50kb.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting User Photo Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	8	Yes	Command ID
user-id	15 characters	Yes	Defines the alphanumeric user id of the user.
photo-format	0-3 0 = jpeg 1 = jpg 2 = png 3 = bmp	Yes	To define the format of photograph.

Example in text

```
cmd-id = 8 user-id = 1234 photo-format=0
```

Example in XML

```
<api>  
<cmd-id >8</cmd-id>  
<user-id>1234</user-id>  
< photo-format>0</ photo-format>  
</api>
```

- **Updating Get User Photo**

After the ‘Get User Photo’ command is used, Device will acknowledge success in status if user photo is updated in specified formate as per input for user else failure will be given.

Parameters

Table: Updating Get User Photo Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	8	Yes	Command ID

Table: Updating Get User Photo Command Request- Parameters

Argument	Valid Values	Mandatory	Description
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
data	Encoded value of user photo data in base 64 format.	Yes	user photo data in base64 format

Response Fields

Table: Updating Get User Photo Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

9. Set User Photo

- **Setting User Photo**

Set User Photo is used to set user photo from server to device. User Photo size should be max 50kb.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Setting User Photo Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	9	Yes	Command ID
user-id	15 characters	Yes	Defines the alphanumeric user id of the user.
photo-format	0-3 0 = jpeg 1 = jpg 2 = png 3 = bmp	Yes	To define the format of photograph.
data	Encoded value of user photo data in base 64 format.	Yes	To send the user photo in base64 format

Example in text

```
cmd-id = 9 user-id = 1234 photo-format=0 data= raw data of image
```

Example in XML

```
<api>  
<cmd-id >9</cmd-id>  
<user-id>1234</user-id>  
< photo-format>0</ photo-format>  
<data>user photo in base64 format </data>  
</api>
```

- **Updating Set User Photo**

After the ‘Set User Photo’ command is used, Device will acknowledge success in status if user photo is updated in device with specified formate as per input for user else failure will be given.

Parameters

Table: Updating Set User Photo Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	8	Yes	Command ID

Table: Updating Set User Photo Command Request- Parameters

Argument	Valid Values	Mandatory	Description
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Set User Photo Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

10. Clear Alarm

- **Clearing Alarm**

Clear Alarm is used to clear on-going alarm on device

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Clear Alarm Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	10	Yes	Command ID

Example in text

```
cmd-id = 10
```

Example in XML

```
<api>  
<cmd-id >10</cmd-id>  
</api>
```

- **Updating clear Alarm**

After the ‘Clear Alarm’ command is used, Device will update success in status if alarm is cleared else failure will be given.

Parameters

Table: Updating Clear Alarm Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	10	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Clear Alarm Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

11. Acknowledge Alarm

- **Acknowledging Alarm**

Acknowledge Alarm is used to acknowledge alarm to stop it.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Acknowledge Alarm Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	11	Yes	Command ID

Example in text

```
cmd-id = 11
```

Example in XML

```
<api>  
<cmd-id>11</cmd-id>  
</api>
```

- **Updating Acknowledge Alarm**

Parameters

Table: Updating Acknowledge Alarm Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	11	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Acknowledge Alarm Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

12. Lock Door

- **Getting Door Locked**

Lock door is used to lock the door.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Lock Door Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	12	Yes	Command ID

Example in text

```
cmd-id = 12
```

Example in XML

```
<api>  
<cmd-id>12</cmd-id>  
</api>
```

- **Updating Lock Door**

After the 'Lock Door' command is used, Device will update success in status if door is Locked else failure will be given.

Parameters

Table: Updating Lock Door Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	12	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Lock Door Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

13. Unlock Door

- **Getting Door Unlocked**

Unlock door is used to unlock the door.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Unlock Door Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	13	Yes	Command ID

Example in text

```
cmd-id = 13
```

Example in XML

```
<api>  
<cmd-id>13</cmd-id>  
</api>
```

- **Updating Unlock Door**

After the ‘Unlock Door’ command is used, Device will update success in status if door is Unlocked else failure will be given.

Parameters

Table: Updating Unlock Door Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	13	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Unlock Door Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

14. Normalize Door

- **Getting Door Normalized**

Normalize door is used to normalize door state from locked or unlocked state.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Normalize Door Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	14	Yes	Command ID

Example in text

```
cmd-id = 14
```

Example in XML

```
<api>  
<cmd-id>14</cmd-id>  
</api>
```

- **Updating Normalize Door**

After the 'Normalize Door' command is used, Device will update success in status if door is in Normalized state else failure will be given.

Parameters

Table: Updating Normalized Door Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	14	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Normalized Door Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

15. Open Door

- **Getting Door Open**

Open door is used to open door through relay operation.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Door Open Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	15	Yes	Command ID

Example in text

```
cmd-id = 15
```

Example in XML

```
<api>  
<cmd-id>15</cmd-id>  
</api>
```

- **Updating Open Door**

After the ‘Open Door’ command is used, Device will update success in status if door is in Open state else failure will be given.

Parameters

Table: Updating Open Door Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	15	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Open Door Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

16. Get Current Event Sequence Number

- **Getting Current Event Sequence Number**

Get Current Event Sequence Number is used to get current sequence number of event from device.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Current Event Sequence Event Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	16	Yes	Command ID

Example in text

```
cmd-id = 16
```

Example in XML

```
<api>
<cmd-id>16</cmd-id>
</api>
```

- **Updating Current Event Sequence Number.**

Parameters

Table: Updating Current Event Sequence Number Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	16	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
Cur-rollover-count	roll over count	Yes	Will show the current rollover count

Table: Updating Current Event Sequence Number Command Request- Parameters

Argument	Valid Values	Mandatory	Description
Cur-Seq-number	event sequence number	Yes	Will show the current event sequence number

Response Fields

Table: Updating Current Event Sequence Number Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

17. Default Device

- **Getting Default Device**

Default Device is used to soft factory default device.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Default Device Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	17	Yes	Command ID

Example in text

```
cmd-id = 17
```

Example in XML

```
<api>
<cmd-id>17</cmd-id>
</api>
```

- **Updating Default Device.**

Parameters

Table: Updating Default Device Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	17	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Default Device Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

18. Reboot Device

- **Getting Device Rebooted**

Reboot Device is used to Restart device.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Reboot Device Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	18	Yes	Command ID

Example in text

```
cmd-id = 18
```

Example in XML

```
<api>  
<cmd-id>18</cmd-id>  
</api>
```

- **Updating Reboot Device.**

Parameters

Table: Updating Reboot Device Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	18	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Reboot Device Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

19. Activate Aux Relay

- **Getting Aux Relay Activated**

Activate Aux Relay is used to activate Aux Relay.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Aux Relay Activate Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	19	Yes	Command ID

Example in text

```
cmd-id = 19
```

Example in XML

```
<api>  
<cmd-id>19</cmd-id>  
</api>
```

- **Updating Aux Relay Activate.**

Parameters

Table: Updating Aux Relay Activate Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	19	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Aux Relay Activate Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

20. Deactivate Aux Relay

- **Getting Aux Relay Deactivated**

Deactivate Aux Relay is used to deactivate Aux Relay.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Aux Relay Deactivate Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	20	Yes	Command ID

Example in text

```
cmd-id = 20
```

Example in XML

```
<api>
<cmd-id>20</cmd-id>
</api>
```

- **Updating Aux Relay Deactivate.**

Parameters

Table: Updating Aux Relay Deactivate Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	20	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Aux Relay Deactivate Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

21. Firmware Update

• *Getting Firmware Updated*

Firmware Update command is used to upgrade device firmware from configured FTP server. On receiving this command, device will check connectivity with configured FTP server and if connected it will download firmware.

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting Firmware Update Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	21	Yes	Command ID

Example in text

```
cmd-id = 21
```

Example in XML

```
<api>  
<cmd-id>21</cmd-id>  
</api>
```

• *Updating Firmware Updates.*

Parameters

Table: Updating Firmware Update Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	21	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed

Response Fields

Table: Updating Firmware Update Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

22. Get User Count

• *Getting User Count*

Get User Count is used to get total user count from device

Parameters

Same as Get Command See “Get Command” on page 15.

Response Fields

Table: Getting User Count Command -Response Fields

Argument	Valid Values	Mandatory	Description
cmd Id	22	Yes	Command ID

Example in text

```
cmd-id = 22
```

Example in XML

```
<api>  
<cmd-id>22</cmd-id>  
</api>
```

• *Updating Get User Count.*

Parameters

Table: Updating Get User Count Command Request- Parameters

Argument	Valid Values	Mandatory	Description
cmd Id	22	Yes	Command ID
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Successful	Yes	status of command executed
User count	1-50000	Yes	Will give total user count

Response Fields

Table: Updating Get User Count Command - Response Fields

Argument	Valid Values	Mandatory	Description
status	0,1 0 = Failure 1 = Successful	Yes	status for acknowledgment
cmd-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any command is available to be issued with server for device

Get Configuration and Update Configuration

Get Configuration

Configurations will be received in device from server through get configuration. Device will save configuration and Update it.

Syntax

```
http://<ServerURL>/getconfig?device-type=<value>&serial-no=<value>&<argument>=<value>
```

Parameters

Table: Get Configuration -Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device

Response Fields

Date Time

The date and time received from server will be local time only, no manipulation based on Time Zone received is required to be done on device.

Table: Get Configuration -date time Response Fields

Argument	Valid Values	Mandatory	Description
config-id	1	Yes	To define the type of config
year	2000to 2037	No	To set year value
month	01 to 12	No	To set month value
date	01 to 31	No	To set date
hour	00 to 23	No	To set hour
minute	00 to 59	No	To set minute
second	00 to 59	No	To set second
time-format	0,1 0 = 24 hour 1 = 12 hour	No	Defines the time format to be displayed on the device display.
time-zone	-	No	To define the universal time zone.
update-mode	0,1 0 = Auto 1 = Manual	No	Defines whether the update mode is manual or through NTP server.
ntp-server-type	0,1 0 = Predefined 1 = User Defined	No	Defines whether the NTP server is a predefined server or user defined server address.
ntp-server	0,1,2 0 = ntp1.cs.wisc.edu 1 = time.windows.com 2 = time.nist.gov	No	To define the NTP address.
user-defined-ntp	40 characters	No	To define the user defined NTP.
dst-enable	0,1,2 0 = Disable 1 = Day Month wise 2 = Date Month wise	No	To enable or disable DST

Table: Get Configuration -date time Response Fields

Argument	Valid Values	Mandatory	Description
fwd-month	0 to 11 0 = January 1 = February 2 = March 3 = April 4 = May 5 = June 6 = July 7 = August 8 = September 9 = October 10 = November 11 = December	No	Forward clock day
fwd-week	0,1,2,3,4 0 = 1st 1 = 2nd 2 = 3rd 3 = 4th 4 = Last	No	Forward week
fwd-day	0 to 6 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday	No	Forward day
fwd-date	1 to 31	No	Forward date
fwd-time-hh	0 to 23	No	Forward clock hour
fwd-time-mm	0 to 59	No	Forward clock Minutes
rev-month	0 to 11 0 = January 1 = February 2 = March 3 = April 4 = May 5 = June 6 = July 7 = August 8 = September 9 = October 10 = November 11 = December	No	Reverse clock day
rev-week	0,1,2,3,4 0 = 1st 1 = 2nd 2 = 3rd 3 = 4th 4 = Last	No	Reverse week

Table: Get Configuration -date time Response Fields

Argument	Valid Values	Mandatory	Description
rev-day	0 to 6 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 =Saturday	No	Reverse Day
rev-date	1 to 31	No	Reverse Date
rev-time-hh	00 to 23	No	Reverse clock hours
rev-time-mm	00 to 59	No	Reverse clock minutes
duration-hh	00 to 23	No	Time by which clock should be forwarded or reversed in hours
duration-mm	00 to 59	No	Time by which clock should be forwarded or reversed in minutes

Example:

To configure device time as 12:34:57 and date as 24/02/2018

```
config-id = 1 year= 2018 month=02 date=24 hour=12 minute=34 second=57
```

Device Basic Configuration

Table: Get Configuration -device basic configuration Response Fields

Argument	Valid Values	Mandatory	Description
config-id	2	Yes	To define the type of config
app	1, 2 1 = Access Control + T&A 2 = T&A	No	To define the application.
name	30 characters	No	To identify/configure the device name.
asc-code	16 bits, 1-65535 range	No	To configure the ASC code
max-fingers	1 to 10 1 = 1 Finger 2 = 2 Finger 3 = 3 Fingers 4 = 4 Fingers 5 = 5 Fingers 6 = 6 Fingers 7 = 7 Fingers 8 = 8 Fingers 9 = 9 Fingers 10=10 Fingers	No	Maximum no. of templates that can be stored per user on this device.

Table: Get Configuration -device basic configuration Response Fields

Argument	Valid Values	Mandatory	Description
max-palms	1 to 10 1 = 1 Palm 2 = 2 Palm 3 = 3 Palm 4 = 4 Palm 5 = 5 Palm 6 = 6 Palm 7 = 7 Palm 8 = 8 Palm 9 = 9 Palm 10=10 Palm	No	Maximum no. of templates that can be stored per user on this device.
generate-invalid-user-events	0,1 0 = No 1 = Yes	No	To generate the invalid user events when invalid user is punched in.
generate-exit-switch-events	0,1 0 = No 1 = Yes	No	To generate the exit switch events.
mifare-custom-key-enable	0,1 0 = Disable 1 = Enable	No	To enable the custom key for mifare card variants.
mifare- custom-key	6 Bytes (12 Hex Digits)	No	To define custom keys for Mifare card variants
hid-iclass-custom-key-enable	0,1 0 = Disable 1 = Enable	No	To enable the custom key for hid-iclass variants
hid-iclass-custom-key	8 Bytes (16 Hex Digits)	No	To define custom keys for HID i-class card variants.
card-custom-key-auto-update	0,1 0 = Disable 1 = Enable	No	To enable the auto update of custom key
finger-format	0,1,2 0 = Proprietary 1 = ISO 2 = Lumidigm ISO	No	To set the mode of the finger template. By default proprietary format will be selected.
manual-door-mode-selection	0,1 0 = Inactive 1 = Active	No	To enable/disable the Door Mode Selection feature

Example:

To configure device as Access Control+TA Mode and max no of finger templates that can be stored per user as 3 fingers

```
config-id = 2 app=1 max-fingers=2
```

Function Key Configuration

Table: Get Configuration -function key configuration Response Fields

Argument	Valid Values	Mandatory	Description
config-id	3	Yes	To define the type of config
F1	0 to 10 0 = None 1 = Official Work - IN 2 = Official Work - OUT 3 = Short Leave - IN 4 = Short Leave - OUT 5 = Regular - IN 6 = Regular - OUT 7 = Break End 8 = Break Start 9 = Overtime - IN 10=Overtime - OUT	No	Assigning special functions to respective function keys.
F2			
F3			
F4			

Example:

To configure function key on device

```
config-id = 3 F1=0 F2=2 F3=1 F4=0
```

Enrollment Option

Table: Get Configuration -Enrollment option Response Fields

Argument	Valid Values	Mandatory	Description
config-id	4	Yes	To define the type of config
enroll-on-device	0,1 0 = Inactive 1 = Active	No	To enable/disable the feature to enroll
enroll-using	0,1 0 = User ID 1 = Reference No.	No	To define the option to enroll the credential with alphanumeric user ID or numeric user ID.
temp-per-finger	0,1 0 = Single Template/Finger 1 = Dual Template/Finger	No	To define the template per finger-mode.
enroll-finger-count	1 to 10	No	No. of finger/ palm / card that is allowed to be enrolled in one enrollment cycle.
enroll-palm-count	1 to 10	No	
enroll-card-count	1 to 4	No	

Table: Get Configuration -Enrollment option Response Fields

Argument	Valid Values	Mandatory	Description
enroll-mode	0,1,2,3 0 = Read Only Card 1 = Smart Card 2 = Biometric 3 = Biometric then Card	No	To define the enrollment mode for enrollment through device

Reader Parameters

Table: Get Configuration -Reader Parameter of Direct Doors except COSEC ARC Response Fields

Argument	Valid Values	Mandatory	Description
config-id	5	Yes	To define the type of config
reader1	0 to 5 0 = None 1 = EM Prox Reader 2 =HID Prox Reader 3 =MiFare Reader 4 =HID iCLASS-U Reader 5 =HID iCLASS-W Reader	No	To define the internal card reader.
reader2	0,1,2 0 = None 1 = Finger Reader 2 = Palm Vein Reader	No	To define the internal biometric reader.
reader3	0 to 14 0 = None 1 = EM Prox Reader 2 = HID Prox Reader 3 = MiFare U Reader 4 = HID iCLASS-U Reader 5 = Finger Reader 6 = HID iCLASS-W Reader 7 = UHF Reader 8 = Combo Exit Reader 9 = Mifare-W Reader 10=PIN-W-Reader 11= 12 =Card + PIN - W Reader 13 =CB - U Reader 14 =CB - W Reader	No	To define the external reader.

Table: Get Configuration -Reader Parameter of Direct Doors except COSEC ARC Response Fields

Argument	Valid Values	Mandatory	Description
door-access-mode	0,1,2,3,4,5,6,11,12,14,15,16,17,18 0 = Card 1 = Biometric 2 = Card+ PIN 3 = PIN+ Biometric 4 = Card + Biometric 5 = Card + PIN +Biometric 6 = Any One 11 = Biometric + Group 12 = Biometric then Card 14 = None 15 = Face 16 = Card + Face 17 = PIN + Face 18 = Biometric + Face	No	To define the door access mode.
door-entry-exit-mode	0,1 0 = Entry 1 = Exit	No	To define the door entry/exit mode.
reader-access-mode	0,1,2,4,6,12,14,19 0 = Card 1 = Biometric 2 = Card + PIN 4 = Card + Biometric 6 = Anyone 12 = Biometric then Card 14 = None 19 = BLE	No	To define the external reader access mode.
reader-entry-exit-mode	0,1 0 = Entry 1 = Exit	No	To define the door entry/exit mode.
tag-re-detect-delay	00 - 3600 Seconds Default- 600 s	No	To define the tag re-detection delay time.

Table: Get Configuration -Reader Parameter of COSEC ARC DC 200 Response Fields

Argument	Valid Values	Mandatory	Description
config-id	5	Yes	To define the type of config
rs-485-readergrp1	0,1,2,3,4,5,13 0 = None 1 = EM Prox Reader 2 =HID Prox Reader 3 =MiFare Reader 4 =HID iCLASS-U Reader 5 =Combo Reader 13=CB-U Reader	No	To define the RS-485 reader in reader1 group.

Table: Get Configuration -Reader Parameter of COSEC ARC DC 200 Response Fields

Argument	Valid Values	Mandatory	Description
wiegand-readergrp1	0,1,2,11,12,14 0 = None 1 = Short - Range Reader 2 = Long - Range Reader 11= PIN-W-Reader 12=Card+PIN -W Reader 14=CB - W Reader	No	To define the Wiegand reader in reader1 group.
readergrp1-entry-exit-mode	0,1 0 = Entry 1 = Exit	No	To define the mode (entry/exit) of the reader1 group.
readergrp1-access-mode	0,1,2,4,6,12,14,19 0 = Card 1 = Biometric 2 = Card + PIN 4 = Card + Biometric 6 = Anyone 12 = Biometric then Card 14 = None 19 = BLE	No	To define the access mode of Reader Group1.
reader-access-mode	0,1,2,4,6,12,14,19 0 = Card 1 = Biometric 2 = Card + PIN 4 = Card + Biometric 6 = Anyone 12 = Biometric then Card 14 = None 19 = BLE	No	To define the external reader access mode.
rs-485-readergrp2	0,1,2,3,4,5,13 0 = None 1 = EM Prox Reader 2 =HID Prox Reader 3 =MiFare Reader 4 =HID iCLASS-U Reader 5 =Combo Reader 13=CB-U Reader	No	To define the RS-485 reader in reader2 group.
wiegand-readergrp2	0,1,2,11,12,14 0 = None 1 = Short - Range Reader 2 = Long - Range Reader 11= PIN-W-Reader 12=Card+PIN -W Reader 14=CB - W Reader	No	To define the Wiegand reader in reader2 group.
readergrp2-entry-exit-mode	0,1 0 = Entry 1 = Exit	No	To define the mode (entry/exit) of the reader2 group.

Table: Get Configuration -Reader Parameter of COSEC ARC DC 200 Response Fields

Argument	Valid Values	Mandatory	Description
readergrp2-access-mode	0,1,2,4,6,12,14,19 0 = Card 1 = Biometric 2 = Card + PIN 4 = Card + Biometric 6 = Anyone 12 = Biometric then Card 14 = None 19 = BLE	No	To define the access mode of Reader Group2.
tag-re-detect-delay	00 - 3600 Seconds Default- 600 s	No	To specify the tag re-detection delay time. This parameter is common for both reader groups.

Example:

To configure internal card reader as HID Prox Reader and internal reader mode as entry

```
config-id = 5 reader1=2 door-access-mode=0
```

Finger Reader Parameter

Table: Get Configuration -Finger Reader Parameter Response Fields

Argument	Valid Values	Mandatory	Description
config-id	6	Yes	To define the type of configuration
security	0 to 4 0 = Level 1(Low) 1 = Level 2 2 = Level 3 3 = Level 4 4 = Level 5(high)	No	To define the security type while enrollment
lighting-cond	0,1 0 = Out Door 1 = In Door	No	To define the lighting condition.
sensitivity	0 to 7 0 = Level 1(Low) 1 = Level 2 2 = Level 3 3 = Level 4 4 = Level 5 5 = Level 6 6 = Level 7 7 = Level 8(high)	No	To define the sensitivity levels from low to high.

Table: Get Configuration -Finger Reader Parameter Response Fields

Argument	Valid Values	Mandatory	Description
fast-mode	0 to 6 0 = 1(Normal) 1 = 2 2 = 3 3 = 4 4 = 5 5 = 6(Fastest) 6 = Auto	No	To define the mode to be used during enrollment.
image-quality	0 to 3 0 = Weak 1 = Moderate 2 = Strong 3 = Strongest	No	To define the acceptable image quality for enrollment.

Example:

To configure sensitivity level of FR Reader as 7

```
config-id = 6 sensitivity=6
```

Palm Reader Parameter

Table: Get Configuration -Palm Reader Parameter Response Fields

Argument	Valid Values	Mandatory	Description
config-id	7	Yes	To define the type of configuration
security	0 to 4 0 = Normal 1 = Highest 2 = High 3 = Low 4 = Lowest	No	To define the security type while enrollment.
palm-matching-timeout	0 to 9999 sec	No	To define the acceptable image quality for enrollment.
palm-temp-quality	0,1,2 0 = Good 1 = Moderate 2 = Poor	No	To define the acceptable image quality for enrollment.

Example:

To configure security level as low

```
config-id = 7 security=3
```

Door Feature

Table: Get Configuration -Door Feature Response Fields

Argument	Valid Values	Mandatory	Description
config-id	8	Yes	To define the type of configuration
allow-exit-when-locked	0,1 0 = Inactive 1 = Active	No	To allow exit when door is locked.
auto-relock	0,1 0 = Inactive 1 = Active	No	To activate and deactivate the auto re-lock.
asc-active	0,1 0 = Inactive 1 = Active	No	To enable/disable the additional security code.
buzzer-mute	0,1 0 = Un mute 1 = Mute	No	To mute/un-mute the buzzer.
door-sense-active	0,1 0 = Inactive 1 = Active	No	To activate/deactivate the door sense.
door- sense	0,1 0 = NO 1 = NC	No	To give the option to sense the door as open or closed.
supervised	0,1 0 = Unsupervised 1 = Supervised	No	To define the option to supervise the door sensing.
exit-switch	0,1 0 = Inactive 1 = Active	No	To enable/disable the exit switch.
aux-output-enable	0,1 0 = Inactive 1 = Active	No	To enable/ Disable the defined aux output port Aux output.
greeting-msg-enable	0,1 0 = Inactive 1 = Active	No	To enable/disable the display message.
greeting-msg<1~4>	-	No	To define the message.
greeting-start-time-hh<1~4>	00-23	No	To define the start time to display the greeting message.
greeting-start-time-mm<1~4>	00-59	No	
greeting-end-time-hh<1~4>	00-23	No	To define the end time to display the greeting message.
greeting-end-time-mm<1~4>	00-59	No	

Example:

To mute buzzer on door

```
config-id = 8 buzzer-mute-1
```

System Timers**Table: Get Configuration -System Timers Response Fields**

Argument	Valid Values	Mandatory	Description
config-id	9	Yes	To define the type of configuration
alarm-ack-timer	10 to 65535 sec.	No	To define the timer for auto alarm acknowledgment.
idwt	1-99 sec	No	To define the inter digit wait timer.
multi-access-wait-timer	3-99 sec	No	To define the multi access wait timer.
palm-enroll-time-out	3-99 sec	No	To define the timer.
pulse-time	1 - 65535 sec	No	To define the door pulse time.
auto-relock-timer	1 to 65535 sec	No	To define the auto relock timer

Example:

To configure Inter digit wait timer as 30 s

```
config-id = 9 idwt=30
```

User Configuration**Table: Get Configuration -System Timers Response Fields**

Argument	Valid Values	Mandatory	Description
config-id	10	Yes	To define the type of configuration
ref-user-id	8 digits	Yes	To select the numeric ID on which the specified operation is to be done
user-index		Yes	To identify the index number for the selected user id
user-id	15 characters	Yes	To set a user's alphanumeric user ID.
name	15 Characters	No	To define the user name.
user-active	0,1 0 = Inactive 1 = Active	No	To activate or deactivate a user.

Table: Get Configuration -System Timers Response Fields

Argument	Valid Values	Mandatory	Description
vip	0,1 0 = Inactive 1 = Active	No	To define a user as VIP.
validity-enable	0,1 0 = Inactive 1 = Active	No	To enable/disable the user validity.
validity-date-dd	1-31	No	To define the end date for user validity.
validity-date-mm	1-12	No	
validity-date-yyyy	2000-2037	No	
user-pin	1 to 6 digits		To set the user PIN
by-pass-finger	0,1 0 = Inactive 1 = Active	No	To enable/disable the bypass finger option.
by-pass-palm	0,1 0 = Inactive 1 = Active	No	To enable/disable the bypass palm option.
card1	64 Bits (8 bytes) (max value - 18446744073709551615)	No	To set the card value against a user.
card2	64 Bits (8 bytes) (max value - 18446744073709551615)	No	To set the card value against a user.
dob-enable	0,1 0 = Enable 1 = Disable	No	To enable/disable the date of birth for a user.
dob-dd	1-31	No	To set dob
dob-mm	1-12	No	
dob-yyyy	2000-2037	No	
user-group	0 to 999	No	To set the user group number.

Example:

To configure a User

```
config-id = 10 user-id= 1234 ref-user-id=1111 vip=1
```

FTP Server

Table: Get Configuration -FTP Server Response Fields

Argument	Valid Values	Mandatory	Description
config-id	11	Yes	To define the type of configuration
auto-upgd	0,1 0 = Enable 1 = Disable	No	To enable/disable auto upgrade
ftp-url	253 char	No	URL of the FTP Server
ftp-user-name	40 char	No	User-name for the FTP Server
ftp-pass	40 char	No	Password for the FTP Server
upgd-on	0,1 0 = Firmware Version 1 = Firmware Time	No	To define the basis of auto upgrade

Example:

To enable auto firmware upgrade

```
config-id = 11 auto-upgd= 1
```

Update Configuration

After device saves configuration, it sends update configuration with status as success. If in case device receives any configuration with any field having invalid values then device will not store configuration and will send status as failure in update configuration.

Syntax

```
http://<ServerURL>/updateconfig?device-type=<value>& serial-no=<value>&config-id=<value>&status=<value>
```

Parameters

Table: Update Configuration -Request Parameters

Argument	Valid Values	Mandatory	Description
config-id	1 to 11 1 = Date Time 2 = Device Basic Configuration 3 = Function Key Configuration 4 = Enrollment Option 5 = Reader Parameter 6 = Finger Reader parameter 7 = Palm Reader Parameter 8 = Door Features 9 = System timers 10=User Configuration 11=FTP Server	Yes	Configuration table Index To define the type of config
device-type	0,1,2,3,5,7 0 =Door V3 1 = Door PVR 2 = Door Vega 3 = Door FMX 5 = ARC DC 200 7 = Door ARGO	Yes	Device Type
serial-no	12 characters	Yes	Serial number of device
status	0,1 0 = Failure 1 = Success	Yes	status of configuration stored or not

Response Fields

Table: Update Configuration - Response Fields

Argument	Valid Values	Mandatory	Description
cnfg-avlbl	0,1 0 = No 1 = Yes	Yes	Defines whether any configuration is available to be issued with server for device
status	0,1 0 = Failure 1 = Success	Yes	status for acknowledgment

Set Events API(s)

This group of APIs enables users for the events to be performed.

Syntax

```
http://<ServerURL>/setevent?device-type=<value>& serial-no=<value>&argument=<value>[&argument=<value>...]
```

Parameters

Table: Set Event- Request Parameters

Argument	Valid Values	Mandatory	Description
device-type	0, 1, 2, 3, 5, 7 0 = Door V3 1 = PVR 2 = Vega 3 = FMX 5 = ARC DC 200 7 = ARGO	Yes	To specify type of device/devices whose list is to be fetched. (only one value at a time)
serial-no	Alphanumeric, Max. 12 characters	Yes	To specify the serial number of the device.
seq-no	1 - 500000	Yes	To define the sequence number of the event
evt_id	For User, Door and System See "Appendix" on page 87.	Yes	To specify the Id of the event
date-dd	0 - 31	Yes	To define the Date and Time
date-mm	0-12	Yes	
date-yyyy	2000-2037	Yes	
time-hh	00-23	Yes	
time-mm	00-59	Yes	
time-ss	00-59	Yes	
field-1	Custom Value	No	To define the field-1 data

Table: Set Event- Request Parameters

Argument	Valid Values	Mandatory	Description
field-2	Custom Value	No	To define the field-2 data
field-3	Custom Value	No	To define the field-3 data
field-4	Custom Value	No	To define the field-4 data
field-5	Custom Value	No	To define the field-5 data

Table: Set Event- Response Parameters

Argument	Valid Values	Mandatory	Description
status	0 & 1 0 = Failure 1 = Successful	Yes	Status of Acknowledgment

Appendix

Table: Universal Time Zone Reference

Index	Universal Time Zone
Index=0	Text="(GMT-12:00) International Date Line West"
Index=1	Text="(GMT-11:00) Midway Island, Samoa"
Index=2	Text="(GMT-10:00) Hawaii"
Index=3	Text="(GMT-09:00) Alaska"
Index=4	Text="(GMT-08:00) Pacific Time (Us & Canada); Tijuana"
Index=5	Text="(GMT-07:00) Arizona"
Index=6	Text="(GMT-07:00) Chihuahua, La Paz, Mazatlan"
Index=7	Text="(GMT-07:00) Mountain Time (Us & Canada)"
Index=8	Text="(GMT-06:00) Central America"
Index=9	Text="(GMT-06:00) Central Time (Us & Canada)"
Index=10	Text="(GMT-06:00) Guadalajara, Mexico City, Monterrey"
Index=11	Text="(GMT-06:00) Saskatchewan"
Index=12	Text="(GMT-05:00) Bogota, Lima, Quito"
Index=13	Text="(GMT-05:00) Eastern Time (Us & Canada)"
Index=14	Text="(GMT-05:00) Indiana (East)"
Index=15	Text="(GMT-04:00) Atlantic Time (Canada)"
Index=16	Text="(GMT-04:00) Caracas, La Paz"
Index=17	Text="(GMT-04:00) Santiago"
Index=18	Text="(GMT-03:30) Newfoundland"
Index=19	Text="(GMT-03:00) Brasilia"
Index=20	Text="(GMT-03:00) Buenos-Aires, Georgetown"
Index=21	Text="(GMT-03:00) Greenland"
Index=22	Text="(GMT-02:00) Mid-Atlantic"
Index=23	Text="(GMT-01:00) Azores"
Index=24	Text="(GMT-01:00) Cape Verde Is"
Index=25	Text="(GMT) CASABLANCA, MONROVIA"
Index=26	Text="(GMT) Dublin, Edinburgh, Lisbon, London"
Index=27	Text="(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna"
Index=28	Text="(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague"
Index=29	Text="(GMT+01:00) Brussels, Copenhagen, Madrid, Paris"
Index=30	Text="(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb"
Index=31	Text="(GMT+01:00) West Central Africa"
Index=32	Text="(GMT+02:00) Athens, Beirut, Istanbul, Minsk"
Index=33	Text="(GMT+02:00) Bucharest"
Index=34	Text="(GMT+02:00) Cairo"
Index=35	Text="(GMT+02:00) Harare, Pretoria"
Index=36	Text="(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius"
Index=37	Text="(GMT+02:00) Jerusalem"
Index=38	Text="(GMT+03:00) Baghdad"
Index=39	Text="(GMT+03:00) Kuwait, Riyadh"
Index=40	Text="(GMT+03:00) Moscow, St Petersburg, Volgograd"
Index=41	Text="(GMT+03:00) Nairobi"
Index=42	Text="(GMT+03:30) Tehran"
Index=43	Text="(GMT+04:00) Abu Dhabi, Muscat"
Index=44	Text="(GMT+04:00) Baku, Tbilisi, Yerevan"
Index=45	Text="(GMT+04:30) Kabul"
Index=46	Text="(GMT+05:00) Ekaterinburg"
Index=47	Text="(GMT+05:00) Islamabad, Karachi, Tashkent"
Index=48	Text="(GMT+05:30) Chennai, Kolkata, New Delhi, Mumbai"
Index=49	Text="(GMT+05:45) Kathmandu"
Index=50	Text="(GMT+06:00) Almay, Novosibirsk"
Index=51	Text="(GMT+06:00) Astana, Dhaka"
Index=52	Text="(GMT+06:00) Sri Jayewardenepura"
Index=53	Text="(GMT+06:30) Rangoon"
Index=54	Text="(GMT+07:00) Bangkok, Hanoi, Jakarta"
Index=55	Text="(GMT+07:00) Krasnoyarsk"
Index=56	Text="(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi"
Index=57	Text="(GMT+08:00) Irkutsk, Ulaanbataar"
Index=58	Text="(GMT+08:00) Kuala Lumpur, Singapore"
Index=59	Text="(GMT+08:00) Perth"
Index=60	Text="(GMT+08:00) Taipei"

Table: Universal Time Zone Reference

Index	Universal Time Zone
Index=61	Text="(GMT+09:00) Osaka, Sapporo, Tokyo"
Index=62	Text="(GMT+09:00) Seoul"
Index=63	Text="(GMT+09:00) Yakutsk"
Index=64	Text="(GMT+09:30) Adelaide"
Index=65	Text="(GMT+09:30) Darwin"
Index=66	Text="(GMT+10:00) Brisbane"
Index=67	Text="(GMT+10:00) Canberra, Sydney, Melbourne,"
Index=68	Text="(GMT+10:00) Guam, Port Moresby"
Index=69	Text="(GMT+10:00) Hobart"
Index=70	Text="(GMT+10:00) Vladivostok"
Index=71	Text="(GMT+11:00) Magadan, Solomon Is, New Caledonia"
Index=72	Text="(GMT+12:00) Auckland, Wellington"
Index=73	Text="(GMT+12:00) Fiji, Kamchatka, Marshall Is"
Index=74	Text="(GMT+13:00) Nuku'alofa"

Event Configuration Reference

Table: List of Events

Event ID	Event Description
101	User Allowed
105	User Allowed – Door Not open
151	User Denied – User Invalid
154	User Denied – Time Out
157	User Denied – Disabled User
158	User Denied – Blocked User
162	User Denied – Door Lock
164	User Denied – Validity date expired
172	User Denied – FP sensor busy
201	Door Status changed
204	Aux input status changed
205	Aux output status changed
206	Door sense input status
207	Door Controller Communication status
208	Door Open/ Close
209	Lock Open/Close changed
405	Enrollment
406	Master Alarm sense input status
407	Master Aux Output status
409	Credentials Deleted
451	Configuration Change
452	Roll over of events

Table: List of Events

Event ID	Event Description
453	Master Controller Power ON
454	Configuration Defaulted
455	Soft Override
456	Backup and Update
457	Default System

Table: Size of Event Fields

Door	Field 1	Field 2	Field 3	Field 4	Field 5	Event Log Capacity
V3	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	5,00,000 events
PVR Door	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	1,00,000 events
Vega Controller	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	5,00,000 events
ARC DC 200	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	5,00,000 events
ARGO	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	5,00,000 events
Door FMX	4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	5,00,000 events

Table: User Events

Event Details						Applicable Device					
Event ID	(Field 1) User ID	(Field 2) Special Code	(Field 3) Entry/Exit	(Field 4)	(Field 5)	Door V3	ARC DC 200	PVR Door	Vega Controller	Door FMX	ARGO
User Allowed Events											
101	Xxxx (user ID=0 for REX input)	Special Function code	Detail	Reference ID of Job Code that is selected by User. Reference ID for None = FFFFF FFF Reference ID for default = FFFFF FFE Reference ID for Continue Job = FFFFF FFD	User ID will be sent if available. else blank will be sent	✓	✓	✓	✓	✓	✓
105	Xxxx	Special Function code	Detail			✗	✗	✗	✗	✗	✗
User Denied Events											

Table: User Events

Event Details						Applicable Device					
Event ID	(Field 1) User ID	(Field 2) Special Code	(Field 3) Entry/Exit	(Field 4)	(Field 5)	Door V3	ARC DC 200	PVR Door	Vega Controller	Door FMX	ARGO
151	(User ID = 0 if not identified)	Special Function code	Detail		User ID will be sent if available. else blank will be sent	✓	✓	✓	✓	✓	✓
154	Xxxx		Detail			✓	✓	✓	✓	✓	✓
157	Xxxx		Detail			✓	✓	✓	✓	✓	✓
158	Xxxx		Detail			✓	✓	✓	✓	✓	✓
162	Xxxx		Detail			✓	✓	✓	✓	✓	✓
164	Xxxx		Detail			✓	✓	✓	✓	✓	✓
172	Xxxx		Detail			✗	✗	✗	✗	✗	✗

Table: Special Function Codes Reference

S.No.	Special Function Name	Special Function Code	Applicable for Allowed Events	Applicable for Denied Events
1	Official Work-IN Marking in T&A	1	✓	✗
2	Official Work-OUT Marking in T&A	2	✓	✗
3	Short Leave-IN Marking in T&A	3	✓	✗
4	Short Leave-OUT Marking in T&A	4	✓	✗
5	Regular - IN Marking in T&A	5	✓	✗
6	Regular - OUT Marking in T&A	6	✓	✗
7	Break End Marking in T&A	7	✓	✗
8	Break Start Marking in T&A	8	✓	✗
9	Over time – IN Marking in T&A	9	✓	✗
10	Over time – OUT Marking in T&A	10	✓	✗
11	Late –IN Allowed Marking in T&A	11	✓	✗
12	Early - OUT Allowed Marking in T&A	12	✓	✗

Table: Special Function Codes Reference

S.No.	Special Function Name	Special Function Code	Applicable for Allowed Events	Applicable for Denied Events
13	Access in Degrade Mode Marking	99	✓	✓
14	Smart Identification	98	✗	✓

Table: Field 3 Detail (User Events) Reference

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU				BLE	Face	API	Group	Palm	Finger	Card	PIN	RFU		Time Stamp	Entry/Exit

Table: Information of Bit 0 and Bit 1

Credential	Bit 1	Bit 0	Value	
Entry	0	0	0	✓
Exit	0	1	1	✓
Entry with Time Stamp Active	1	0	2	✗
Exit with Time Stamp Active	1	1	3	✗

Table: Information of Bit 4 and Bit 9

Credential	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Value
PIN	0	0	0	0	0	1	1
Card	0	0	0	0	1	0	2
Card + PIN	0	0	0	0	1	1	3
Finger	0	0	0	1	0	0	4
Finger + PIN	0	0	0	1	0	1	5
Finger + Card	0	0	0	1	1	0	6
Finger + Card + PIN	0	0	0	1	1	1	7
Palm	0	0	1	0	0	0	8
PIN + Palm	0	0	1	0	0	1	9
Card + Palm	0	0	1	0	1	0	10
PIN + Card + Palm	0	0	1	0	1	1	11
Group + Palm	0	1	1	0	0	0	24
API	1	0	0	0	0	0	32
API + Finger	1	0	0	1	0	0	36
API + Palm	1	0	1	0	0	0	40
API + PIN	1	0	0	0	0	1	33

Table: Information of Bit 4 and Bit 9

Credential	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Value
Group + Finger	0	1	0	1	0	0	20
API + CARD	1	0	0	0	1	0	34
Face	0	0	0	0	0	0	64
PIN + Face	0	0	0	0	0	1	65
Card + Face	0	0	0	0	1	0	66
Finger + Face	0	0	0	1	0	0	68
BLE	0	0	0	0	0	0	128
BLE + Card	0	0	0	0	1	0	130
BLE + Finger	0	0	0	1	0	0	132
BLE + Palm	0	0	1	0	0	0	136

Table: Door Events

Event Details						Applicable Devices					
Event ID	(Field 1) Status	(Field 2)	(Field 3)	(Field 4)	(Field 5)	Door V3	ARC DC 200	PVR Door	Vega Controller	Door FMX	ARGO
201	1= Normal 2= Locked 3= Unlocked 8= Armed 9= Disarmed					✓	✓	✓	✓	✓	✓
204	4= Activated 1= Normal 6= Fault (open) 7= Fault (short) 11= Disabled		0-7= Aux Input Port of IO Controller			✓	✓	✓	✓	✓	✓
205	4= Activated 1= Normal 11= Disabled		0-7=Aux Output Port of IO Controller			✓	✓	✓	✓	✓	✓
206	1= Normal 6= Fault (open) 7= Fault (short) 11= Disabled					✓	✓	✓	✓	✓	✓
207			0=ON Line 1=Off Line 2=Upgrading			✓	✓	✓	✓	✓	✓
208	Reference ID xxxx	0= Open 1= Closed 2= Not Applicable			User ID will be sent if available, else blank will be sent	✓	✓	✓	✓	✓	✓
209	1= Normal 4= Activated	0=Lock Open 1=Lock Close				✗	✗	✗	✗	✗	✗

Table: System Events

Event Details						Applicable Devices					
Event ID	(Field 1)	(Field 2)	(Field 3)	(Field 4)	(Field 5)	Door V3	ARC DC 200	PVR Door	Vega Controller	Door FMX	ARGO
405	ID: Xxxx	8 = User Card 9 = User Finger 10 = Special Cards 14 = Palm 16 = Palm template with guide mode 17 = User Finger-Suprema ISO format 18 = User Finger-Lumidigm ISO format	0= FP-1/ Palm-1 1= Card1/ FP-2/Palm-2 2= Card-2/ F-3/Palm-3 3 = Card-3/ FP-4/Palm- 4 = Card-4/ FP-5/Palm-5 5= FP-6/ Palm-6 6= FP-7/ Palm-7 7= FP-8/ Palm-8 8= FP-9/ Palm-9 9= FP-10/ Palm-10	Palm template enrollment quality: 1=Good Quality 2=moderate quality 3=poor quality	User ID will be sent if available. else blank will be sent	✓	✓	✓	✓	✓	✓
406			1=Normal 2=Fault (Open) 3= Fault(Short) 4= Activated			✗	✗	✗	✗	✗	✗
407			1=Normal 4=Activated			✗	✗	✗	✗	✗	✗
409	ID: Xxxx	8 = User Cards 9 = User Fingers 14 = Palm	5= Web Jeeves 6= ACMS 7= Special Function			✓	✓	✓	✓	✓	✓
451	Configuration Table ID xxx	Index start	Index end			✓	✓	✓	✓	✓	✓
452	Roll over number 00 to 99					✓	✓	✓	✓	✓	✓
453						✓	✓	✓	✓	✓	✓
454	Configuration Table ID xxx	Index start	Index end			✓	✓	✓	✓	✓	✓
456	1=Backup 2=Update	1=Configuration 2=Event 3=Firmware	0=Fail 1=Success 2 = CRC Check Fail			✓	✓	✓	✓	✓	✓
457			6 = from ACMS 8 = from Hardware			✓	✓	✓	✓	✓	✓



MATRIX COMSEC PVT. LTD.

Head Office:

394-GIDC, Makarpura, Vadodara - 390010, India.

Ph.: (+91)18002587747

E-mail: Support@MatrixComSec.com

www.matrixaccesscontrol.com