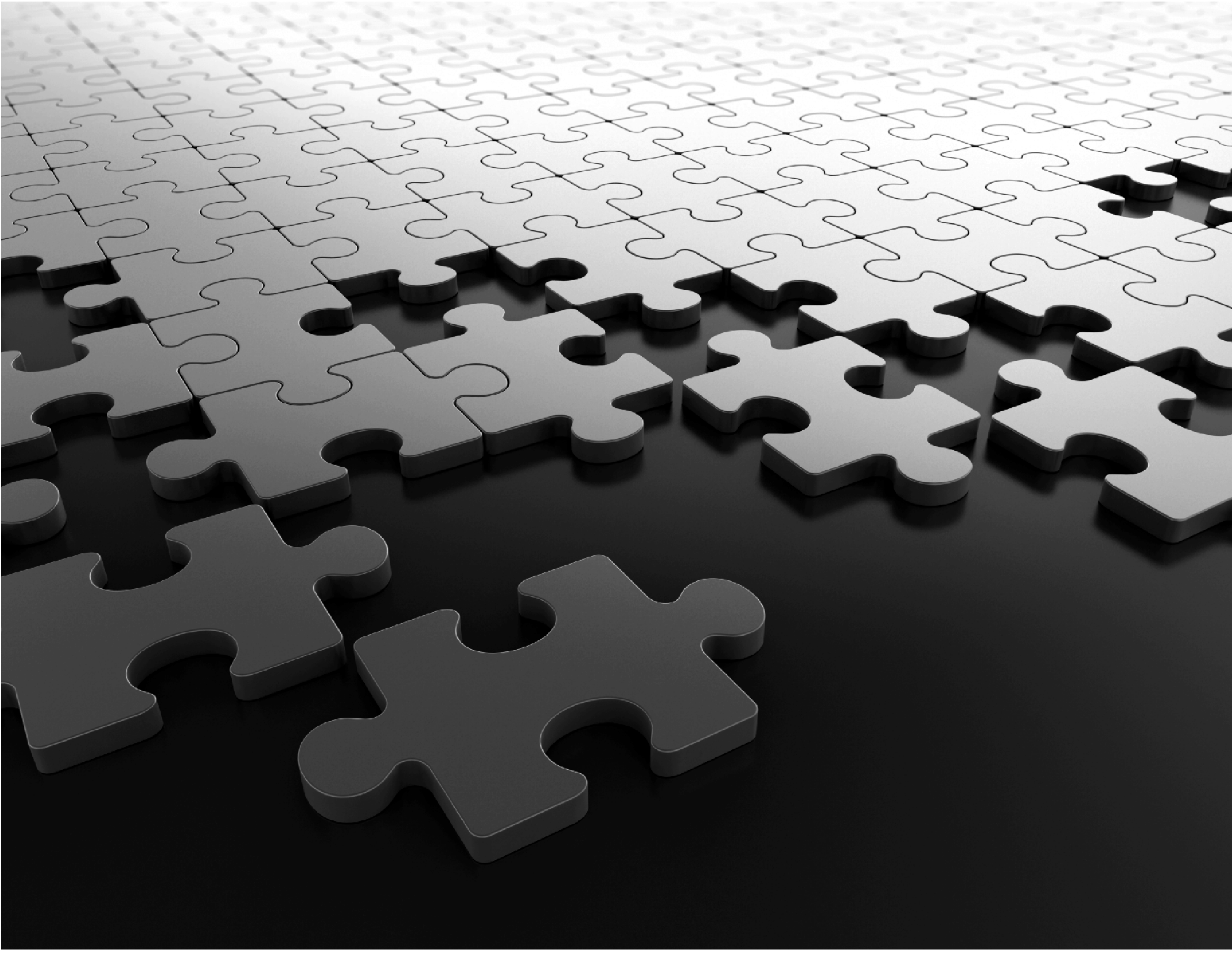
# COSEC Devices API
# User Guide



MATRIX
SECURITY SOLUTIONS

MATRIX
SECURITY SOLUTIONS

# COSEC Devices 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

*Version 1.0*
*Release date: May 31, 2014*

# Contents

# List of Tables

# About the Document

Welcome to the *COSEC Devices API User Guide*. This document will provide you a comprehensive overview and complete user-guidance for all *COSEC Devices APIs*. You can learn more about COSEC 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 ommitted 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. API Overview
3. Supported APIs
4. Device Configuration
5. User Configuration
6. Enrollment
7. Events
8. Sending Commands to Devices
9. Error Responses
10. API Response Codes
11. Appendix

Topics 1 and 2 will provide a general understanding of COSEC Devices APIs and the basic interface communication. Topic 3 provides a list of all supported APIs with a quick reference list for the user. Topics 4-8 provide an overview of API categories with detailed explanation of individual APIs. 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*).

Topic 9 provides illustrations of error messages. Topic 10 provides a list of API Response Codes and their meaning. 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 API User Guide is meant for *third-party software developers* who wish to operate COSEC Devices via another remote application. This guide will provide information to users on how to request and receive services from COSEC Devices using a COSEC API.

# API Overview

COSEC Devices APIs provide an interface for communication with COSEC Devices via HTTP methods. These APIs will enable specific functions to be performed on your remote devices such as setting basic and advanced device configurations, configuring users on device, performing enrollment of credentials, monitoring events and sending commands to device. For a complete list of COSEC Device APIs, refer to *Supported APIs*.

## How It Works

Following is an illustration of how the COSEC system typically communicates in a client-server based architecture.

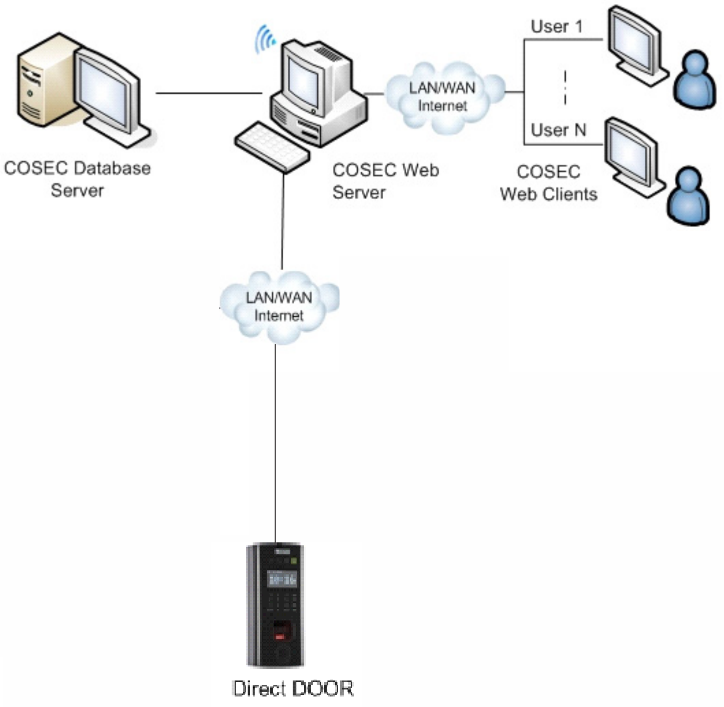


Fig. Communication through COSEC Web Server

However, here the communication with COSEC devices occurs via the COSEC Web server. On the other hand, Devices APIs enable a client application to access and monitor a remotely installed COSEC device directly, without installing the COSEC server/Monitor.

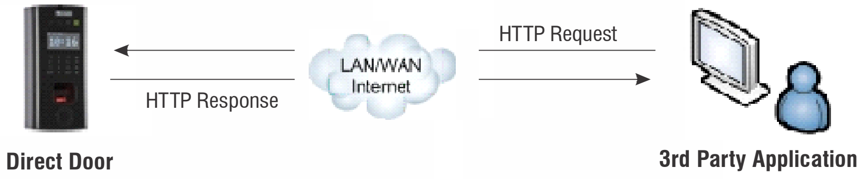


Fig. Communication through COSEC API

Using APIs, the third party can send a simple HTTP request to configure, control or command a device. The device then processes and executes this request to return an appropriate response.

## Supported Devices

COSEC Devices APIs are dependant on the device type. Currently, Device APIs are supported on the following COSEC Door Controllers and their variants:

- COSEC Direct Door V2
- COSEC Path Controller
- COSEC Wireless Door
- COSEC NGT Door
- COSEC PVR Door
- COSEC Vega Controller

## General Features

All COSEC APIs -

- Are Web-based *HTTP* APIs.
- Use basic *HTTP Request-Response* for interface communication.
- Generate response in either *text* or *XML* (Extensible Markup Language) format.
- Use simple *HTTP commands* such as *GET*, *SET*, *DELETE* etc.
- Use a generic syntax for all queries.
- Support some predefined parameters and their corresponding values for each action. Each parameter will either be mandatory or bear a system-defined default value (when no value is specified).
- Use a mandatory parameter ***action*** universally, which takes action values (such as ***get, set, delete*** etc.) and specifies the action to be requested.

## What the User Should Know

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

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

## Prerequisite

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

- A COSEC Device (pre-installed)
- A network enabled for accessing the COSEC Device.
- The credentials for API Authentication

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

## Authentication

The device shall request basic authentication for granting access. Default username and password for HTTP session authentication are:

Username: admin
Password: 1234

## 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` | *The start line* |
| `*(message-header CRLF)` | *Zero or more header fields or ‘headers’* |
| `CRLF` | *An empty line* |
| `[Message-body]` | *A message-body (chunk or payload)* |

```
Start-line = Request-Line | Status-Line
```

## Communication Flow

The communication takes place in the following manner:

1. The client checks availablility of the device.

2. If available, the client issues a request for the device.

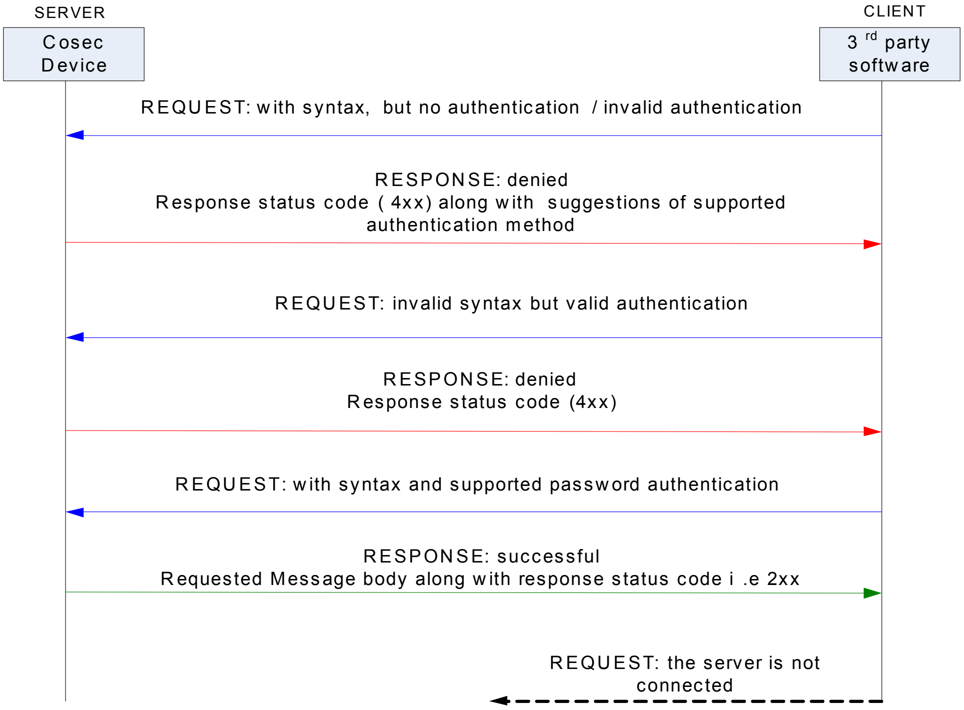

SERVER
Cosec Device
CLIENT
3 rd party software
REQUEST: with syntax, but no authentication / invalid authentication
RESPONSE: denied
Response status code ( 4xx) along with suggestions of supported authentication method
REQUEST: invalid syntax but valid authentication
RESPONSE: denied
Response status code (4xx)
REQUEST: with syntax and supported password authentication
RESPONSE: successful
Requested Message body along with response status code i .e 2xx
REQUEST: the server is not connected

Fig: communication flow

3. The device parses the request for the action to be taken.

4. In case of an error (*invalid syntax*, *invalid authentication* etc.), the request is denied and an error response is returned. Else, the requested data is returned with the appropriate response code.

## Request Format

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

| | | |
|---|---|---|
| 1. | Request Line | This line is constituted by the following three elements which must be separated by a space:<br>• The method type (GET, HEAD, POST, PUT etc.)<br>• The requested URL<br>• The HTTP version to use<br>For e.g.:<br>`GET http://192.168.1.2/device.cgi/command?action=geteventcount HTTP/1.0` |

| | | |
|---|---|---|
| 2. | Header Fields | Add information about the request using these header fields:<br>• A General Header (<Header-name>:<value>).<br>• A Request Header (<Header-name>:<value>).<br>• 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:**

```
GET http://matrix.com/ HTTP/1.0
Accept: text/html
If-Modified-Since: Saturday, 15-January-2000 14:37:11 GMT
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
```

## 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<crlf>
HEADER: Value<crlf>
.
.
.
HEADER: Value<crlf>
Empty line<crlf>
BODY OF THE RESPONSE
```

It consists of the following components:

| | | |
|---|---|---|
| 1. | A status line | This line is constituted by the following three elements which must be separated by a space:<br>• The version of the protocol used (e.g. *HTTP/1.0*).<br>• The status code (indicates the status of the request being processed).<br>• 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
Date: Sat, 15 Jan 2000 14:37:12 GMT
Server: Microsoft-IIS/2.0
Content-Type: text/HTML
Content-Length: 1245
Last-Modified: Fri, 14 Jan 2000 08:25:13
GMT
```

***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.*

## URL Syntax

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

### Syntax

```
http://<deviceIP:deviceport>/device.cgi/<request-type>?<argument>=<value>[&<argument>=<value>......]
```

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

| URL element | Description |
|---|---|
| *http://* | This is the protocol used to communicate with the client.<br>**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. |
| *<deviceIP:deviceport>* | This identifies the device with which communication is to be performed. It consists of two components:<br>deviceIP: Device IP address<br>deviceport: Device Port Number |
| *device.cgi* | This is a mandatory entity required to specify the CGI directory for all the device-related commands. |
| *<request-type>* | This specifies the type of API request. For the mandatory request types, please refer to the individual API descriptions. |

| URL element | Description |
|---|---|
| *<argument>* | This defines a specific action or command depending on the function to be performed.<br>A mandatory argument for all COSEC API functions is *action*. This argument always takes an action as its value (For eg. *action=get*).<br>For more information on the common HTTP actions used in COSEC APIs, please refer to section *Common Actions.* |
| *<value>* | These are argument values that determine the output. |

### Example

Let us assume that the target device has the IP address 192.168.x.y and the device port number is *80*. The user wants to fetch basic configured parameters for the device. In this case, a sample request would resemble the following:

```
http://192.168.x.y:80/device.cgi/device-basic-config?action=get&format=xml
```

In this case, the query uses an ***action=get*** parameter which is commonly used to retrieve information from the device-side. The URL takes another argument called ***format*** which specifies that the response returned should be in the XML format.

- Special characters ( &, ‘, “, **<**, **>**, #, % and **;**) will not be allowed in arguments or their values. Special character “**&**” will be allowed as a separator between consecutive arguments and “**?**” will be allowed as a separator between the request-type and an argument.
- The request line and headers must all end with <CR><LF> that is carriage return character followed by a line feed character.
- The status line and header must all end with <CR><LF>.
- The empty line must consist of only <CR><LF> and no other white space.

## Common Actions

The following actions are commonly used in COSEC APIs as values for the ‘***action***’ argument:

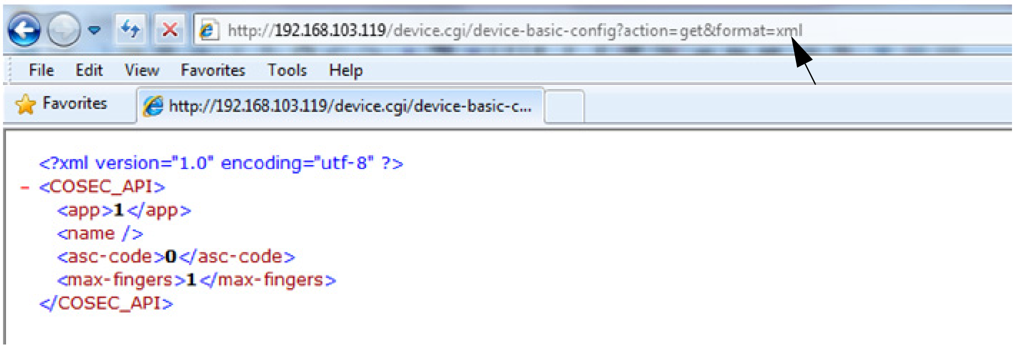
| Action | Use |
|---|---|
| *GET* | To fetch required data from device. |
| *SET* | To set required parameters for a given function. |
| *GETDEFAULT* | This is used to get default the parameters of all/ specified argument. If any argument is specified then default value of that particular argument is returned else default value of complete group is returned. |
| *SETDEFAULT* | This is used to default the parameters. If any argument is specified then default that particular value else default complete group |

| Action | Use |
|---|---|
| *DELETE* | To delete data from device. |
| *ENROLL* | To enroll an entity to a device. |

## Additional Information

- Generally, all the commands will be supported in the GET Method and hence the arguments and valid values will be expected in the URL. Wherever applicable POST method will be specified explicitly. For the POST method, the parameters must be included in the body of the HTTP request.
- To set blank values in a particular field, a blank can follow the “=”. E.g. “argument=&”
- If the format is not specified then by default, the values should be returned in text format.
- For all arguments other than ‘action’, the position in the URL may be changed.

*COSEC APIs use basic authentication and can be tested on any standard Web browser. Enter the request URL in the address field of your browser and press the ‘Enter’ key to send query to the device. Enter the authentication credentials when prompted. The response will be displayed on your browser in the specified format.*



```
http://192.168.103.119/device.cgi/device-basic-config?action=get&format=xml
```

```
  <?xml version="1.0" encoding="utf-8" ?>
- <COSEC_API>
    <app>1</app>
    <name />
    <asc-code>0</asc-code>
    <max-fingers>1</max-fingers>
  </COSEC_API>
```

# Supported APIs

COSEC Devices support the following groups of APIs categorized on the basis of functions to be performed:

- Device Configuration
- User Configuration
- Enrollment
- Events
- Sending Commands to Device

## API Quick Reference

This section enables users to view a quick reference list of all supported Devices APIs discussed in the guide. The following table lists all functions along with their respective HTTP Request URLs and the applicable action values. For further details on supported parameters and values, refer to the respective argument-value tables for individual APIs (See *List of Tables)*.

**Table: API Quick Reference**

| URL | Actions | Functionality |
|---|---|---|
| `http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Basic Device Configuration |
| `http://<deviceIP:deviceport>/device.cgi/function-key?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Function Key Configuration |
| `http://<deviceIP:deviceport>/device.cgi/reader-config?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Reader Configuration |
| `http://<deviceIP:deviceport>/device.cgi/enroll-options?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Enrollment Configuration |
| `http://<deviceIP:deviceport>/device.cgi/access-setting?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Access Settings Configuration |
| `http://<deviceIP:deviceport>/device.cgi/alarm?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Alarm Configuration |
| `http://<deviceIP:deviceport>/device.cgi/date-time?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Date and Time Configuration |
| `http://<deviceIP:deviceport>/device.cgi/door-feature?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Door Features Configuration |
| `http://<deviceIP:deviceport>/device.cgi/system-timer?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | System Timers Configuration |
| `http://<deviceIP:deviceport>/device.cgi/special-function?action=<value>[&<argument>=<value>….]` | get, set, getdefault, setdefault | Special Function Configuration |
| `http://<deviceIP:deviceport>/device.cgi/users?action=<value>[&<argument>=<value>….]` | set, get | Setting/Retrieving User Configuration |
| `http://<deviceIP:deviceport>/device.cgi/userphoto?action=<value>[&<argument>=<value>….]` | get, set, delete | Setting a User Photo |
| `http://<deviceIP:deviceport>/device.cgi/users?action=delete[&<argument>=<value>….]` | delete | Deleting a User |
| `http://<deviceIP:deviceport>/device.cgi/credential?action=set[&<argument>=<value>….]` | set | Setting User Credentials |
| `http://<deviceIP:deviceport>/device.cgi/credential?action=get[&<argument>=<value>….]` | get | Retrieving User Credentials |

**Table: API Quick Reference**

| URL | Actions | Functionality |
|---|---|---|
| `http://<deviceIP:deviceport>/device.cgi/credential?action=delete[&<argument>=<value>….]` | delete | Deleting User Credentials |
| `http://<deviceIP:deviceport>/device.cgi/enrolluser?action=enroll[&<argument>=<value>….]` | enroll | Enrolling a User |
| `http://<deviceIP:deviceport>/device.cgi/enrollspcard?action=enroll[&<argument>=<value>….]` | enroll | Enrolling Special Crads |
| `http://<deviceIP:deviceport>/device.cgi/events?action=getevent[&<argument>=<value>….]` | getevent | Retrieving Events |
| `http://<deviceIP:deviceport>/device.cgi/tcp-events?action=getevent[&<argument>=<value>….]` | getevent | Retrieving Events in TCP Socket |
| `http://<deviceIP:deviceport>/device.cgi/command?action=clearalarm` | clearalarm | Sending Commands -<br>Clear Alarm |
| `http://<deviceIP:deviceport>/device.cgi/command?action=getcount` | getcount | Sending Commands -<br>Get Credential Count for Enrolled Credentials |
| `http://<deviceIP:deviceport>/device.cgi/command?action=acknoledgealarm` | acknoledgealarm | Sending Commands -<br>Acknowledge Alarm |
| `http://<deviceIP:deviceport>/device.cgi/command?action=lockdoor` | lockdoor | Sending Commands -<br>Lock Door |
| `http://<deviceIP:deviceport>/device.cgi/command?action=unlockdoor` | unlockdoor | Sending Commands -<br>Unlock Door |
| `http://<deviceIP:deviceport>/device.cgi/command?action=normalizedoor` | normalizedoor | Sending Commands -<br>Normalize Door |
| `http://<deviceIP:deviceport>/device.cgi/command?action=getusercount` | getusercount | Sending Commands -<br>Getting User Count on Device |
| `http://<deviceIP:deviceport>/device.cgi/command?action=geteventcount` | geteventcount | Sending Commands -<br>Get Current Event Sequence Number |
| `http://<deviceIP:deviceport>/device.cgi/command?action=systemdefault` | systemdefault | Sending Commands - Default the System Configuration |
| `http://<deviceIP:deviceport>/device.cgi/command?action=deletecredential` | deletecredential | Sending Commands - Delete Credentials for All Users |

# Device Configuration

This group of APIs enables users to perform the following types of device configuration:

- Basic Device Configuration
- Function Key Configuration
- Reader Configuration
- Finger Reader Parameter Configuration
- Palm Sensor Parameter Configuration
- Enrollment Configuration
- Access Settings Configuration
- Alarm Configuration
- Date and Time Configuration
- Door Features Configuration
- System Timers Configuration
- Special Function Configuration

## Basic Device Configuration

**Description:** To set or retrieve basic configuration parameters for a device such as application type, name, Additional Security Code and maximum number of finger templates on device.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Device Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| app | 1, 2 | No | To define the application.<br>1 = Advanced Access Control<br>2 = Basic Access Control |
| name | Alphanumeric,<br>Max. 30 characters | No | To identify/configure the device name. |
| asc-code | Numeric, 16 bits,<br>1-65535 range | No | To configure an Additional Security Code (ASC). Should be non-zero. |
| Max-fingers | Single Template/Finger: 0-9<br>where,<br>0 - 1 Finger<br>1 - 2 Fingers<br>2 - 3 Fingers<br>3 - 4 Fingers<br>4 - 5 Fingers<br>5 - 6 Fingers<br>6 - 7 Fingers<br>7 - 8 Fingers<br>8 - 9 Fingers<br>9 - 10 Fingers<br>Dual Template/Finger: 0-4<br>where,<br>0 - 1 Finger<br>1 - 2 Fingers<br>2 - 3 Fingers<br>3 - 4 Fingers<br>4 - 5 Fingers | No | Maximum no. of templates that can be stored per user on this device. |
| format | text, xml | No | specifies the format in which the response is expected. |

*The **Additional Security Code** is a code that can be written on a smart card for adding an additional layer of security check during door access.*

*To get the default values for any parameter, use the **action=getdefault** method. To restore configuration parameters on device to default values, use the **action=setdefault** method.*

## Example

Following are some test cases for your reference:

1. **To get all parameters.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=get
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body:
app=1 name= asc-code=0 max-fingers=1
```

2. **To get device name, when expected value is blank and the response format is in text.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=get&name&app
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body:
app=1 name=
```

3. **To get device name, when the expected value is blank and the response format is XML.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=get&name&app&format=xml
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body:
<COSEC_API>
<name></name>
<app>1</app>
</COSEC_API>
```

4. **To set device name as blank– Valid argument.**

### Sample Request

```
http://<deviceIP:deviceport>/device.cgi/device-basic-config?action=set&name=
```

### Sample Response

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body: Response-Code=0
```

## Function Key Configuration

**Description:** To set or retrieve configuration of Function Keys on the Device keypad. COSEC enables its users to map up to 4 special functions to the arrow keys on a Direct Door keypad. These functions can then be performed at the door by using the keypad shortcuts. Use this API to specify which special functions are to be assigned shortcuts on COSEC devices.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/function-key?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Function Key Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| F1 | 0 = None<br>1 = Official IN<br>2 = Official OUT<br>3 = Short Leave IN<br>4 = Short Leave OUT<br>5 = Regular IN<br>6 = Regular OUT<br>7 = Post Break IN<br>8 = Pre - Break OUT<br>9 = Overtime IN<br>10 = Overtime OUT | No | Assigning special functions to respective function keys. |
| F2 | | | |
| F3 | | | |
| F4 | | | |
| format | text,xml | No | Specifies the format in which the response is expected. |

### Example

1. **To configure function key F1 as official work – IN.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/function-key?action=set&f1=1
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <type>
Content-Length: <length>
Body: Response-Code=0
```

## Reader Configuration

**Description:** To set or retrieve configuration parameters for internal and external readers such as reader type, access mode, entry-exit mode and the tag re-detection delay time.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/reader-config?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Reader Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| reader1 | 0 = None<br>1 = EM Prox Reader<br>2 = HID Prox Reader<br>3 = MiFare Reader<br>4 = HID iCLASS-U Reader<br>5 = HID iCLASS-W Reader | No | To define the internal card reader. |
| reader2 | 0 = None<br>1 = Finger Reader<br>2 = Palm Vein Reader | No | To define the internal biometric reader. |
| reader3 | 0 = None<br>1 = EM Prox Reader<br>2 = HID Prox Reader<br>3 = MiFare U Reader<br>4 = HID iCLASS-U Reader<br>5 = Finger Reader<br>6 = HID iCLASS-W Reader<br>7 = UHF Reader<br>8 = Combo Exit Reader<br>9 = MiFare-W Reader | No | To define the external reader. |
| door-access-mode | 0 = Card<br>1 = Finger<br>2 = Card + PIN<br>3 = PIN + Finger<br>4 = Card + Finger<br>5 = Card + PIN + Finger<br>6 = Any<br>7 = Palm<br>8 = Palm + PIN<br>9 = Card + Palm<br>10 = Card + PIN + Palm<br>11 = Palm + Group (Optional)<br>12 =  Finger then Card<br>13 = Palm then Card | No | To define the access mode applicable for door access. |
| door-entry-exit-mode | 0 = Entry<br>1 = Exit | No | To define the whether the internal reader is to be set on an entry or exit mode. |

**Table: Reader Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| reader-access-mode | 0 = Card<br>1 = Finger<br>4 = Card + Finger<br>6 = Any<br>12 = Finger then Card | No | To define the access mode applicable for the external reader. |
| reader-entry-exit-mode | 0 = Entry<br>1 = Exit | No | To define the whether the external reader is to be set on an entry or exit mode. |
| tag-re-detect-delay | 00 - 3600 seconds | No | To define the tag re-detection delay time. |
| format | text,xml | No | Specifies the format in which the response is expected. |

## Example

1. **To configure internal card reader as an HID Prox reader and internal reader mode as entry**.

### Sample Request

```
http://<deviceIP:deviceport>/device.cgi/reader-config?action=set&reader1=2&door-access-mode=0
```

### Sample Response

```
HTTP Code: 200 OK
Content-Type: <type>
Content-Length: <length>
Body: Response-Code=0
```

## Finger Reader Parameter Configuration

**Description:** To set the finger reader calibration for fingerprint enrollment.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi /finger-parameter?<argument>=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Finger Reader Parameter Configuration - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| security | 0 = Normal<br>1 = Secure<br>2 = More Secure<br>Default = 0 | Yes | To define the security type while enrollment. |
| lighting-cond | 0 = Out door<br>1 = In door<br>Default =1 | No | To define the lighting condition. |
| sensitivity | 0 = Level 1 (Low)<br>1 = Level 2<br>2 = Level 3<br>3 = Level 4<br>4 = Level 5<br>5 = Level 6<br>6 = Level 7<br>7 = Level 8 (High)<br>Default = 7 | No | To define the sensitivity levels from low to high. |
| fast-mode | 0 = Mode 1 (Normal)<br>1 = Mode 2<br>2 = Mode 3<br>3 = Mode 4<br>4 = Mode 5<br>5 = Mode 6 (Fastest)<br>6 = Auto<br>Default = 6 | No | To define the mode to be used during enrollment. |
| image-quality | 0 = Weak<br>1 = Moderate<br>2 = Strong<br>3 = Strongest<br>Default = 1 | No | To define the acceptable image quality for enrollment. |
| format | text,xml | No | Specifies the format in which the response is expected |

## Palm Sensor Parameter Configuration

**Description:** To set the palm sensor calibration for palm enrollment.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi /palm-parameter?<argument>=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Finger Reader Parameter Configuration - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| security | 0 = Normal<br>1 = Highest<br>2 = High<br>3 = Low<br>4 = Lowest<br><br>Default = 2 | Yes | To define the security type while enrollment. |
| palm-matching-timeout | 0 to 9999 sec<br><br>Default = 15 sec | No | To define the palm matching timeout. |
| palm-temp-quality | 0 = Good<br>1 = Moderate<br>2 = Poor<br><br>Default = 1 | No | To define the acceptable image quality for enrollment. |
| format | text,xml | No | Specifies the format in which the response is expected |

## Enrollment Configuration

**Description:** To set or retrieve configuration parameters for enrollment of credentials on a device such as number of credentials allowed, number of templates allowed per finger, enrollment mode etc.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/enroll-options?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Enrollment Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| enroll-on-device | 0 = Inactive<br>1 = Active | No | To enable/disable the feature to enroll through special function |
| enroll-using | 0 = User ID<br>1 = Reference No. | No | To define the option to enroll the credential using the user's Reference No. or User ID, for enrollment through special function.<br>Note: This parameter will not be valid for NGT Direct Door and Vega Controller where enrollment must be performed by User ID. |
| temp-per-finger | 0 = Single Template/ Finger<br>1 = Dual Template/Finger | No | To define the number of templates to be saved per finger. |
| enroll-finger-count | Single Template/Finger: 0-9<br>where,<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers<br>5 = 6 Fingers<br>6 = 7 Fingers<br>7= 8 Fingers<br>8 = 9 Fingers<br>9 = 10 Fingers<br>Dual Template/Finger: 0-4<br>where,<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers | No | No. of fingers allowed to be enrolled in one enrollment cycle.<br>Note: For the **action=set** method, this value should not be greater than the **max-finger** value set in Basic Device Configuration API. |

**Table: Enrollment Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| enroll-palm-count | 0 = 1 Palm<br>1 = 2 Palms<br>2 = 3 Palms<br>3 = 4 Palms<br>4 = 5 Palms<br>5 = 6 Palms<br>6 = 7 Palms<br>7 = 8 Palms<br>8 = 9 Palms<br>9 = 10 Palms | No | No. of palms allowed to be enrolled in one enrollment cycle. |
| enroll-card-count | 0 = 1 Card<br>1 = 2 Cards<br>2 = 3 Cards<br>3 = 4 Cards | No | No. of special function cards allowed to be enrolled in one enrollment cycle. |
| enroll-mode | 0 = Read Only Card<br>1 = Smart Card<br>2 = Finger Print<br>3 = FP then Card<br>4 = Palm Template<br>5 = Palm then Card | No | To define the enrollment mode for enrollment through device. |
| format | text,xml | No | Specifies the format in which the response is expected. |

- *If the **temp-per-finger** mode is changed, then the templates have to be restored to the device explicitly by the third party software, else mismatch will occur in the module.*

- *If **Single Template/Finger** mode is selected on the device and some users are already enrolled according to it and if abruptly the mode is changed to **Dual Template/Finger** then:*

  **i.** *If the maximum finger count was greater than 5 fingers in Single Template/Finger mode, then after changing the mode to the Dual Template/Finger, the finger count will set to 5.*

  **ii.** *If the maximum finger count was less than 5 fingers in Single Template/Finger mode, then after changing the mode to the Dual Template/Finger, the finger count will remain same.*

- *If the mode is changed back to Single Template/Finger, then finger count should not be changed. If users want to increase the finger count they should mention it explicitly.*

# Access Settings Configuration

**Description:** To set or retrieve configuration parameters for enabling basic access control on a device for users.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/access-setting?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Access Settings Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| week-day<0~6> | sun (0) to sat (6)<br>0 = Inactive<br>1 = Active | No | To define the active working days. This parameter is repeated for each day of the week. |
| work-start-hh | 00-23 | No | Define the work start time |
| work-start-mm | 00-59 | No | Define the work start time |
| work-end-hh | 00-23 | No | Define the work stop time |
| work-end-mm | 00-59 | No | Define the work stop time |
| format | text, xml | No | Specifies the format in which the response is expected |

## Example

1. **To get data for all parameters in the text format.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/access-setting?action=get&format=xml
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body:
week-day0=1 week-day1=1 week-day2=1 week-day3=1 week-day4=1 week-day5=1 week-day6=1 work-start-hh=0
work-start-mm=0 work-end-hh=23 work-end-mm=59
```

## Alarm Configuration

**Description:** To set or retrieve configuration parameters for enabling/disabling alarms and related functions on a COSEC device such as Auto Alarm Acknowledgement.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/alarm?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Alarm Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| alarm | 0 = Inactive<br>1 = Active | No | To enable/disable alarm. |
| tamper-alarm | 0 = Inactive<br>1 = Active | No | To enable or disable the feature. |
| auto-alarm-ack | 0 = Inactive<br>1 = Active | No | To enable or disable the Auto Alarm Acknowledgement feature. |
| format | text,xml | No | Specifies the format in which the response is expected. |

## Date and Time Configuration

**Description:** To set or retrieve date and time configurations on a COSEC device. The user can configure the date and time to be displayed on the device, the display format, the time update mode, the NTP server settings as well as the Daylight Savings Time (DST) settings on the selected device.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/date-time?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Date and Time Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| year | 2009 to 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 minutes |
| second | 00 to 59 | No | To set seconds |
| time-format | 0 = 24 hours<br>1 = 12 hours | No | Defines the time format to be displayed on the device display.<br>Note: This is applicable only for the time shown on the device display and not for general date-time which will always be in 24 hours format. |
| update-mode | 0 = Auto<br>1 = Manual | No | Defines whether the update mode is manual or through NTP Server. |
| ntp-server-type | 0 = Predefined<br>1 = User Defined | No | Defines whether the NTP server is a predefined server or user-defined server address. |
| time-zone | 00-74 (Tool supported by Windows), default: GMT (+05:30) Chennai, Kolkata, Mumbai, New Delhi.<br>Refer to *"Table: Universal Time Zone Reference" on page 61* | No | To define the universal time zone. |
| ntp-server | 0 = ntp1.cs.wisc.edu<br>1 = time.windows.com<br>2 = time.nist.gov | No | To define the NTP Address. |
| user-defined-ntp | Alphanumeric, Max. 40 characters. | No | To define the user-defined NTP. |
| dst-enable | 0 = Disable<br>1 = Enable | No | To enable/disable DST. |

**Table: Date and Time Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| fwd-month | 0 = January<br>1 = February<br>2 = March<br>3 = April<br>4 = May<br>5 = June<br>6 = July<br>7 = August<br>8 = September<br>9 = October<br>10 = November<br>11 = December | No | Forward clock day |
| fwd-week | 0 = 1st<br>1 = 2nd<br>2 = 3rd<br>3 = 4th<br>4 = Last | | |
| fwd-day | 0 = Sunday<br>1 = Monday<br>2 = Tuesday<br>3 = Wednesday<br>4 = Thursday<br>5 = Friday<br>6 = Saturday | | |
| fwd-time-hh | 00 - 23 (24 hours format only) | No | Forward clock time instance |
| fwd-time-mm | 00 - 59 | | |
| rev-month | 0 = January<br>1 = February<br>2 = March<br>3 = April<br>4 = May<br>5 = June<br>6 = July<br>7 = August<br>8 = September<br>9 = October<br>10 = November<br>11 = December | No | Reverse clock day |
| rev-week | 0 = 1st<br>1 = 2nd<br>2 = 3rd<br>3 = 4th<br>4 = Last | No | |

**Table: Date and Time Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| rev-day | 0 = Sunday<br>1 = Monday<br>2 = Tuesday<br>3 = Wednesday<br>4 = Thursday<br>5 = Friday<br>6 = Saturday | No | Reverse clock day |
| rev-time-hh | 00 - 23 (24 hours format only) | No | Reverse clock time instance |
| rev-time-mm | 00 - 59 | | |
| duration-hh | 00 - 23 (24 hours format only) | No | Time by which clock should be forwarded or reversed. |
| duration-mm | 00 - 59 | | |
| format | text,xml | No | Specifies the format in which the response is expected. |

- *When user sets the time locally it should be GMT time. And in GET command also the time value to be returned will be GMT time irrespective of the time displaying on the device.*
- *While configuring Daylight Saving Parameters, users are responsible to define the forward and reverse time properly.*

## Door Features Configuration

**Description:** To enable, disable, define or retrieve configuration parameters related to various door features such as auto-relock, ASC, door sense, exit switch, greeting message display, voice guidance etc.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/door-feature?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Door Features Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| allow-exit-when-locked | 0 = Inactive<br>1 = Active | No | To allow exit when door is locked. |
| auto-relock | 0 = Inactive<br>1 = Active | No | To enable/disable the Auto-relock feature. |
| asc-active | 0 = Inactive<br>1 = Active | No | To enable/disable the Additional Security Code (ASC). |
| buzzer-mute | 0 = Unmute<br>1 = Mute | No | To mute/un-mute the buzzer. |
| door-sense-active | 0 = Inactive<br>1 = Active | No | To enable/disable sensing of door states. |
| door-sense | 0 = NO<br>1 = NC | No | To define the normal door state as as normally open (NO) or normally closed (NC). |
| supervised | 0 = Unsupervised<br>1 = Supervised | No | To enable/disable supervised sensing of door states (four-state monitoring of door controllers). |
| exit-switch | 0 = Inactive<br>1 = Active | No | To enable/disable the exit switch. |
| greeting-msg-enable | 0 = Inactive<br>1 = Active | No | To enable/disable the display greeting message. |
| greeting-msg<1~4> | Alphanumeric, Max. 21 ASCII characters | No | To define upto 4 display greeting messages, the start time and the end time for displaying each message. |
| greeting-start-time-hh<1~4> | 00-23 | No | |
| greeting-start-time-mm<1~4> | 00-59 | No | |
| greeting-end-time-hh<1~4> | 00-23 | No | |
| greeting-end-time-mm<1~4> | 00-59 | No | |
| voice-guidance | 0 = Inactive<br>1 = Active | No | To enable/disable Voice Guidance (Only for NGT doors). |

**Table: Door Features Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| format | text,xml | No | Specifies the format in which the response is expected. |

- *When greeting messages are defined in an order then first message will always have precedence over second and second over third and so on. Hence, if two messages defined with overlapped timing range, the first defined message between two will have the priority.*
- *Third party should always take care of setting the time range for different messages.*

## System Timers Configuration

**Description:** To set or retrieve configurations for the following system timers:

| | |
|---|---|
| **Auto Alarm Acknowledgement Timer** | Specifies the time period in seconds after which an unacknowledged alarm will acknowledge itself automatically. |
| **Inter Digit Wait Timer** | Specifies time period in seconds between two key inputs on the device keypad. On the expiry of this timer, the system considers the user input to be complete and is ready for the next input. |
| **Multi Access Wait Timer** | Defines the time in seconds for which the system needs to wait for the second credential input from a user when more than one credential is required to grant access. |
| **Palm Enrollment Time Out Timer** | Defines the time period in seconds within which a palm must be enrolled after generating the enrollment command. |
| **Door Open Pulse Timer** | Defines the time in seconds required for a door to be energized for a valid credential. If the opened door does not return to its closed state before the expiry of this timer, the door will generate a “Door Abnormal Alarm”. |
| **Special Function Timer** | Defines the time in minutes for which the Late-IN and Early-OUT special functions will remain active after being enabled at the door controller. |

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/system-timer?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: System Timers Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| alarm-ack-timer | 10 to 65535 (sec) | No | To define the timer for Auto Alarm Acknowledgement. |
| 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 Palm Enrollment Time out Timer. |
| pulse-time | 1 - 65535 (sec) | No | To define the Door Pulse time |
| sp-function-timer | 1-99 (mins) | No | To define the Special Function Timer. |
| format | text,xml | No | Specifies the format in which the response is expected. |

## Special Function Configuration

**Description:** COSEC enables its users to perform certain pre-defined operations directly from the COSEC device. These are known as special functions. An RFID card can be encoded for a special function and the card-holder can perform this function at the device just by showing this special card.

Use this API to enable, disable, define or retrieve Special Functions configuration on a device.

**Actions:** get, set, getdefault, setdefault

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/special-function?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Special Function Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| Sp-fn-Index | 1 = Offical Work - IN<br>2 = Official Work - OUT<br>3 = Short Leave - IN<br>4 = Short Leave - OUT<br>5 = Regular - IN<br>6 = Regular - OUT<br>7 = Post Break - IN<br>8 = Pre Break - OUT<br>9 = Over Time - IN<br>10 = Over Time - OUT<br>11 = Enroll User<br>12 = Enroll Special Card<br>13 = Delete Credentials<br>14 = Late IN - Start<br>15 = Late IN - Stop<br>16 = Early OUT - Start<br>17 = Early OUT- Stop<br>18 = Door Lock<br>19 = Door Unlock<br>20 = Door Normal<br>21 = Clear Alarm | Yes | The index number of a special function. |
| enable | 0 = Disable<br>1 = Enable | No | To enable/disable special functions on the device. |
| card1 | 64 Bits (20 Numeric Digits approx.) | No | To define the special function card 1. |
| card2 | 64 Bits (20 Numeric Digits approx.) | No | To define the special function card 2. |
| card3 | 64 Bits (20 Numeric Digits approx.) | No | To define the special function card 3. |
| card4 | 64 Bits (20 Numeric Digits approx.) | No | To define the special function card 4. |
| format | text,xml | No | Specifies the format in which the response is expected. |

# User Configuration

The various COSEC devices have capacity to support the following number of users:

| | |
|---|---|
| Direct Door V2 | :2000 |
| NGT Direct Door | :10,000 |
| Wireless Door | :50,000 |
| Path Controller | :2000 |
| PVR Door | :10,000 |
| Vega Controller | :50,000 |

This group of APIs enables users to add or delete users, set user photographs, add or fetch various configurations related to users on or from a device as well as synchronize credentials with device. The following functions can be called:

- Setting/Retrieving User Configuration
- Setting a User Photo
- Deleting a User
- Setting User Credentials
- Retrieving User Credentials
- Deleting User Credentials

## Setting/Retrieving User Configuration

**Description:** To set basic user configuration parameters on a device using the ***action=set*** parameter and retrieve configuration details using ***action=get***.

**Actions:** get, set

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/users?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: User Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-id | Maximum 10 characters | Yes | To set or retrieve the alphanumeric user ID for the selected user.<br>Note: If a ***set*** request is sent against an existing user ID, then configuration for this user will be updated with the new values. |
| user-index | Direct Door V2= 1 - 2,000<br>Path Controller = 1 - 2,000<br>Wireless Door = 1 - 50,000<br>PVR = 1 - 10,000<br>NGT = 1 - 10,000<br>Vega Controller = 1 - 50,000 | No | To identify the index number for the selected user ID (only ***get*** parameter) |
| ref-user-id | Maximum 8 digits | Yes (Not mandatory for the ***get*** action) | To select the numeric user ID on which the specified operation is to be done. |
| name | Alphanumeric. Max. 15 characters | No | To define the user name |
| user-active | 0 = Inactive<br>1 = Active | No | to activate or deactivate a user. |
| vip | 0 = Inactive<br>1 = Active | No | To define a user as VIP.<br>Note: A VIP user is a user with the special privilege to access a particular door. |
| validity-enable | 0 = Inactive<br>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 | |

**Table: User Configuration Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-pin | 1 to 6 Digits | No | To set the user PIN or get the event from user PIN.<br>Note: The user-pin can be set to a blank value. |
| by-pass-finger | 0 = Inactive<br>1 = Active | No | To enable/disable the bypass finger option. |
| by-pass-palm | 0 = Inactive<br>1 = Active | No | To enable/disable the bypass palm option. |
| card1 | 64 Bits (8 bytes) (max value - 18446744073709551615) | No | To set or delete the card value against a user. |
| card2 | 64 Bits (8 bytes) (max value - 18446744073709551615) | No | To set or delete the card value against a user. |
| dob-enable | 0 = Enable<br>1 = Disable | No | To enable/disable the display of a birthday message. |
| dob-dd | 1-31 | No | To set or delete the date of birth for a user. |
| dob-mm | 1-12 | | |
| dob-yyyy | 1990-2037 | | |
| user-group | 0-999 | No | To set the user group number.<br>**Note:** A user can be assigned to any user group ranging from 1 to 999. User group number can be set/update via “Set” command. To remove a user from an assigned user group, user group should be set to 0. |
| format | text, xml | No | Specifies the format in which the response is expected. |

- *For **set** requests only one user’s complete data should be sent at a time. Attempting to set data for multiple users at a time will return an error response. For more examples of error responses, see Error Responses.*
- *To create a new user on device, both **user-id** and **ref-user-id** are mandatory parameters to be provided, and these should be unique for each user.*
- *If a user is already configured in the system and admin wants to update the user with new information/data, only Alphanumeric User ID is sufficient but if the reference user ID is also mentioned then it would be verified whether this belongs to the same user or not.*
- *Whenever an event is generated related to a user, the required user ID field upon calling the event will always show user’s reference user ID. Whereas if “Get” action is sent to call user configuration then it will show alphanumeric user ID.*

## Example

1. **To get user names for user-id = 1**

### Sample Request

```
http://deviceIP:deviceport/device.cgi/users?action=get&user-id=1&format=xml
```

### Sample Response

```
HTTP Code: 200 OK
Content-Type: <xml>
Content-Length: <length>
Body:
<COSEC_API>
<user-id>1</user-id>
<user-index>0</user-index>
<ref-user-id></ref-user-id>
<name></name>
<user-active>0</user-active>
<vip>0</vip>
<validity-enable>0</validity-enable>
<validity-date-dd>1</validity-date-dd>
<validity-date-mm>1</validity-date-mm>
<validity-date-yyyy>2009</validity-date-yyyy>
<user-pin></user-pin>
<by-pass-finger>0</by-pass-finger>
<card1>0</card1>
<card2>0</card2>
</COSEC_API>
```

## Setting a User Photo

**Description:** To set, fetch or delete a photograph against a user's profile on the device using a third party application.

**Actions:** get, set, delete

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/userphoto?action=<value>[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Setting a User Photo - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-id | Maximum 10 characters | Yes | To specify the alphanumeric user ID for the user whose photo is to be set. |
| user-photo | N/A | Yes | To get, set or delete the user photo. This should be done in the data portion of the request / response.(applicable only for VEGA and NGT doors) |
| photo-format | 0 = jpeg<br>1 = jpg<br>2 = png<br>3 = bmp | Yes (only for ***set*** parameter) | To define the format for the photograph. |
| format | text,xml | No | Specifies the format in which the response is expected. |

### Example

Following are some test cases for your reference:

1. **To add an image file in .jpeg format for user-id 1.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/userphoto?action=set&user-id=1&photo-format=0
Data:
Image data
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <code>
Content-Length: <type>
Body: Response-Code=0
```

2. **To fetch the user photo for the same user.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/userphoto?action=get&user-id=1
```

### Sample Response

```
HTTP Code: 200 OK
Content-Type: image/jpeg
Content-Length: 12345
Body:
```

```
<JPEG Image Data>
```

*This is an example only. The actual response will vary depending on product model and configuration.*

## Deleting a User

**Description:** To delete a user from a device. Deleting a user will result in deletion of the credentials of that user along with all the other configurations set on the device.

**Actions:** delete

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/users?action=delete[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Delete User - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-id | Maximum 10 characters | Yes | To specify the alphanumeric user ID for the user to be deleted. |
| format | text,xml | No | Specifies the format in which the response is expected. |

## Setting User Credentials

**Description:** To set a user’s biometric or card credentials on a device.

**Actions:** set

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/credential?action=set[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Setting User Credentials - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| type | 1 = Finger<br>2 = Card<br>3 = Palm | Yes | To define the user credentials type. |
| user-id | Alphanumeric (Max 10 characters) | Yes | To select the user-id for which the credential is to be fetched. |
| card1 | 64 Bits (8 bytes) (max value - 18446744073709551615) | No | It defines the value for card-1 |
| card2 | 64 Bits (8 bytes) (max value - 18446744073709551615) | No | It defines the value for card-2 |
| format | text,xml | No | Specifies the format in which the response is expected. |
| data | - | No | This is the data of respective credential type, which is to be stored at given index number for the respective user id. |

## Retrieving User Credentials

**Description:** To retrieve a user’s credential information from a device.

**Actions:** get

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/credential?action=get[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Retrieving User Credentials - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| type | 1 = Finger<br>2 = Card<br>3 = Palm | Yes | To define the user credentials type. |
| user-id | Alphanumeric (Max. 10 characters | Yes | To select the user-id for which the credential is to be fetched. |
| card1 | 64 Bits (8 bytes) (max value - 18446744073709551615) | | It defines the value for card-1 |
| card2 | 64 Bits (8 bytes) (max value - 18446744073709551615) | | It defines the value for card-2 |
| finger-index | 1 = 1 Finger<br>2 = 2 Fingers<br>3 = 3 Fingers<br>4 = 4 Fingers<br>5 = 5 Fingers<br>6 = 6 Fingers<br>7 = 7 Fingers<br>8 = 8 Fingers<br>9 = 9 Fingers<br>10 = 10 Fingers | No | Identifies the number of finger templates/palm templates to be set or retrieved, on or from the device. The template will be set and retrieved from the data portion of the request and response. |
| palm-index | 1 = 1 Palm<br>2 = 2 Palms<br>3 = 3 Palms<br>4 = 4 Palms<br>5 = 5 Palms<br>6 = 6 Palms<br>7 = 7 Palms<br>8 = 8 Palms<br>9 = 9 Palms<br>10 = 10 Palms | No | |
| format | text,xml | No | Specifies the format in which the response is expected. |
| data | - | No | This is the data of respective credential type, which is to be stored at given index number for the respective user id. |

- *Credential parameters to be applied will depend on the credential type selected.*
- *At a time only finger print or palm can be get/set. Both cannot be set at the same time.*
- *The set command is basically similar to adding and duplication of finger template will not be verified by the device. It is expected to be handled by the 3rd party software.*
- *The method used in this case should be POST method as it consists of raw/ hex data in the data portion of the request and the response.*
- *Finger/palm index fields are not mentioned as mandatory fields because if user selects credential type card then there is no need to specify the finger or palm index, similarly if credential type is finger then palm index in not a mandatory field and vice versa.*

## Deleting User Credentials

**Description:** To delete selected credentials of a user from a device.

**Actions:** delete

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/credential?action=delete[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Deleting User Credentials - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-id | Alphanumeric (Max. 10 characters) | Yes | To delete the credential of a particular user. |
| type | 0 = All<br>1 = Finger<br>2 = Card<br>3 = Palm | Yes | Defines the credential type to be deleted.<br>Note: For the selected type, all credentials will be deleted. |
| format | text,xml | No | Specifies the format in which the response is expected. |

### Example

1. **To delete finger templates of user id 1.**

**Sample Request**

```
http://deviceIP:deviceport/device.cgi/credential?action=delete&user-id=1&type=1
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <type>
Content-Length: <length>
Body: Response-Code=0
```

# Enrollment

The Enrollment APIs can be used to generate an enrollment request for a device. Once the enrollment request is successfully sent on the device, the device will initiate the enrollment process and request credentials to be provided physically, as per the credential type and sequence specified.

Perform the enrollment function on a remote door controller using these enrollment APIs:

- Enrolling a User
- Enrolling Special Cards

## Enrolling a User

**Description:** To command a device to initiate enrollment for a user based on parameters specified.

**Actions:** enroll

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/enrolluser?action=enroll[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Enrolling User - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| type | 0 = Read Only Card<br>1 = Smart Card<br>2 = Finger Print<br>3 = FP Then Card<br>4 = Palm Template<br>5 = Palm Then Card | Yes | Defines the credential to be enrolled. |
| user-id | Maximum 10 characters | Yes | Defines the alphanumeric User ID of the user whose credential is to be enrolled. |
| finger-count | Single Template/Finger: 0-9<br>where,<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers<br>5 = 6 Fingers<br>6 = 7 Fingers<br>7= 8 Fingers<br>8 = 9 Fingers<br>9 = 10 Fingers<br>Dual Template/Finger: 0-4<br>where,<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers | No | To specify the number of fingers to be enrolled. |
| card-count | 0 = 1 Card<br>1 = 2 Cards<br>2 = 3 Cards<br>3 = 4 Cards | No | To specify the number of cards to be enrolled. |

**Table: Enrolling User - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| palm-count | 0 = 1 Palm<br>1 = 2 Palms<br>2 = 3 Palms<br>3 = 4 Palms<br>4 = 5 Palms<br>5 = 6 Palms<br>6 = 7 Palms<br>7 = 8 Palms<br>8 = 9 Palms<br>9 = 10 Palms | No | To specify the number of palms to be enrolled. |
| w-asc | 0 = Inactive<br>1 = Active | No | To enable/disable the Additional Security Code (ASC) to be written on the Smart Card. |
| w-fc | 0 = Inactive<br>1 = Active | No | To enable/disable the Facility Code (FC) to be written on the Smart Card. |
| w-ref-user-id | 0 = Inactive<br>1 = Active | No | To enable/disable the User ID to be written on the Smart Card. |
| w-name | 0 = Inactive<br>1 = Active | No | To enable/disable the User Name to be written on the Smart Card. |
| w-designation | 0 = Inactive<br>1 = Active | No | To enable/disable the designation to be written on the Smart Card. |
| w-branch | 0 = Inactive<br>1 = Active | No | To enable/disable the branch name to be written on the Smart Card. |
| w-department | 0 = Inactive<br>1 = Active | No | To enable/disable the department name to be written on the Smart Card. |
| w-bg | 0 = Inactive<br>1 = Active | No | To enable/disable the blood group to be written on the Smart Card. |
| w-contact | 0 = Inactive<br>1 = Active | No | To enable/disable Emergency Contact information to be written on the Smart Card. |
| w-medical-history | 0 = Inactive<br>1 = Active | No | To enable/disable the medical history to be written on the Smart Card. |
| w-fp-template | 0 = No Templates<br>1 = 1 Finger Template<br>2 = 2 Finger Templates | No | To enable/disable the finger templates to be written on the Smart Card. |
| name | Alphanumeric, 15 Chars, ASCII Code | No | Defines the values for the respective fields to be written on the Smart Card. |
| designation | | | |
| branch | | | |
| department | | | |

**Table: Enrolling User - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| bg | Maximum 4 characters. Valid Values:<br>A+<br>A-<br>B+<br>B-<br>AB+<br>AB-<br>O+<br>O-<br>A1-<br>A1+<br>A1B-<br>A1B+<br>A2-<br>A2+<br>A2B-<br>A2B+<br>B1+ | No | Defines the values for the respective fields to be written on the Smart Card.<br>Note: '**bg**' stands for blood group of the user. |
| contact | Alphanumeric, 15 Chars, ASCII Code | No | |
| medical-history | Alphanumeric, 15 Chars, ASCII Code | No | |
| format | text,xml | No | Specifies the format in which the response is expected. |

- *This is only to send enrollment command, if the credential is to be retrieved then it has to be retrieved explicitly using the get and set credential command.*

- *By default, if count is not specified for enroll command then consider it as one and perform the enroll operation.*

- *This enrollment has no links to the parameter configured on the device for "enroll through special function".*

## Example

1. **To start enrollment of two fingers for user id 45.**

**Sample Request**

```
http://deviceIP:deviceport/device.cgi/enrolluser?action=enroll&user-id=45&type=2&finger-count=1
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <type>
Content-Length: <length>
Body: Response-Code=0
```

## Enrolling Special Cards

**Description:** A Special Card is an RFID card which can be encoded for a special function. This API enables the user to perform enrollment of special cards on the selected device based on specified parameters such as special function ID and number of cards to be enrolled as special cards.

**Actions:** enroll

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/enrollspcard?action=enroll[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Enroll Special Cards - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| sp-fn-id | All configured Special Functions (special function ID) | Yes | Defines the special function identification number. |
| card-count | 0 = 1 Card<br>1 = 2 Cards<br>2 = 3 Cards<br>3 = 4 Cards | No | To specify the number of cards to be enrolled. |
| format | text,xml | No | Specifies the format in which the response is expected. |

# Events

Any action that occurs or is performed using a live COSEC device is referred on the COSEC system as an Event. A client application can directly request event logs to be fetched from a specific device or be fed with live events data via the device listening port. The functions available in this API group are as follows:

- Retrieving Events
- Retrieving Events in the TCP Socket

## Retrieving Events

**Description:** To request all or specified events from a device.

**Actions:** getevent

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/events?action=getevent[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Retrieving Events - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| roll-over-count | 0 to 65535 | Yes | This identifies the first event that is to be sent to the 3rd party from a set of events sent in this response. If the “no-of-events” field value is 1, then this will be the only event sent to the server. |
| seq-number | Refer to “Table: Value Range for Event Sequence Numbers” for the valid values on different devices. | Yes | |
| no-of-events | 1 to 5 (for Direct Door V2 and Path Controller)<br>1 to 100 ( for all other Direct Doors) | No | Specifies the number of events to be fetched. |
| format | text,xml | No | Specifies the format in which the response is expected. |

**Table: Value Range for Event Sequence Numbers**

| Door | Event Sequence Number |
|---|---|
| V2 | 1 to 50,000 |
| CDC | 1 to 50,000 |
| Wireless | 1 to 5,00,000 |
| NGT | 1 to 1,00,000 |
| PVR | 1 to 1,00,000 |
| Vega Controller | 1 to 5,00,000 |

- *For different kind of events, different fields are required, to understand the functionality of an event, which are denoted as detail fields.*
- *The details field in the response depends on the type of device.*

### Example

1. **To request specific events with roll over count = 0 and sequence number = 1. No. of events requested is 3, for an NGT door.**

**Sample Request**

```
http://deviceIP:deviceport/device.cgi/events?action=getevent&roll-over-count=0&seq-number=1&no-of-events=3
```

## Sample Response

```
HTTP Code: 200 OK
Content-Type: xml
Content-Length: 12345
Body:
<COSEC_API>
<Events>
<roll-over-count>0</roll-over-count>
<seq-No>1</seq-No>
<date>16/4/2014</date>
<time>14:56:20</time>
<event-id>457</event-id>
<detail-1>0</detail-1>
<detail-2>0</detail-2>
<detail-3>6</detail-3>
<detail-4>0</detail-4>
<detail-5>0</detail-5>
</Events>
<Events>
<roll-over-count>0</roll-over-count>
<seq-No>2</seq-No>
<date>16/4/2014</date>
<time>14:56:20</time>
<event-id>453</event-id>
<detail-1>0</detail-1>
<detail-2>0</detail-2>
<detail-3>0</detail-3>
<detail-4>0</detail-4>
<detail-5>0</detail-5>
</Events>
<Events>
<roll-over-count>0</roll-over-count>
<seq-No>3</seq-No>
<date>16/4/2014</date>
<time>14:57:28</time>
<event-id>453</event-id>
<detail-1>0</detail-1>
<detail-2>0</detail-2>
<detail-3>0</detail-3>
<detail-4>0</detail-4>
<detail-5>0</detail-5>
</Events>
</COSEC_API>
```

For example if an enrollment event is called in which three fingers have been enrolled with the dual template per finger then the detail fields will be as follows:

**For first finger:**

- Event-ID: 405 (code for enrollment event)
- Detail-1: user-id
- Detail-2: 9 (code for finger credential)
- Detail-3: **12**
- Detail-4: 0
- Detail-5: 0

**For second finger:**

- Event-ID: 405 (code for enrollment event)
- Detail-1: user-id
- Detail-2: 9 (code for finger credential)
- Detail-3: **24**
- Detail-4: 0
- Detail-5: 0

**For third finger:**

- Event-ID: 405 (code for enrollment event)
- Detail-1: user-id
- Detail-2: 9 (code for finger credential)
- Detail-3:**36**
- Detail-4: 0
- Detail-5: 0

If the template per finger mode was selected as single template per finger then the respective values for detail 3 will be 11, 22 and 33, where LSB denotes the template index.

## Retrieving Events in the TCP Socket

**Description:** To receive all or specific events through the TCP listening port of the device.

**Actions:** getevent

**Syntax:** `http://<deviceIP:deviceport>/device.cgi/tcp-events?action=getevent[&<argument>=<value>….]`

**Parameters:** All arguments for this query and their corresponding valid values are listed below:

**Table: Retrieving Events in the TCP Socket - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| ipaddress | IP address and port number validations are same as for network configuration settings. | Yes | Defines the IP Address and the listening port on which the events are to be sent. |
| port | | | |
| roll-over-count | 0 to 65535 | Yes | It is used to specify the exact sequence number of an event stored at any port. |
| seq-number | Refer to “Table: Value Range for Event Sequence Numbers” for the valid values on different devices. | Yes | It is used to specify the sequence number of any event. The maximum value for this can be from 1 to the event log capacity of that device. |
| response-time | 3 - 15 seconds | No | To specify the response time to wait for a confirmation of established network. |
| interface | 0 = Ethernet<br>1 = Wi-Fi<br>2 = Mobile Broadband | No | Specifies the interface.<br>Note: If no interface is defined, **Ethernet** will be tried by default. |
| format | text,xml | No | Specifies the format in which the response is expected. |

*Due to memory constraints, this API is not supported on Direct Door V2.*

### Example

1. **To request to send the events continuously on the TCP port from event seq 1 and roll over count 0 on IP address 192.168.102.42 and tcp listening port 80.**

**Sample Request**

```
http://deviceIP:deviceport/device.cgi/tcp-events?action=getevent&ipaddress=192.168.102.42&port=80&roll-over-
count=0&seq-number=1
```

### Sample Response

```
HTTP Code: 200 OK
Content-Type: <type>
Content-Length: <length>
Body: Response-Code=0
```

- *The default TCP protocol acknowledgement should be used to send the next event. If in case any event is missed in between, then it is the responsibility of the 3rd party to re-request for that event. This shouldn't be done via TCP port but missed events can be re-requested through HTTP API.*

- *If during the event transferring if reboot occurs then the prior command (to send events) will no longer be valid and client must re-request events. In such a case, the events which have already been sent, will be overwritten by the same.*

- *The user ID against which an event is stored must be the Reference ID for a user. This being numeric (max. 8 digits), will enable efficient utilization of storage space on devices, especially those having high event logging capacity (upto 5,00,000 events).*

# Sending Commands to Device

It is possible to send CGI commands to a device in order to perform certain functions.

The generic URL for these commands: `http://<deviceIP:deviceport>/device.cgi/command?action=<value>`

**Table: List of Commands to Device**

| S.No. | Command to Device | Action | Description |
|---|---|---|---|
| 1 | Clear Alarm | clearalarm | To command the device to clear an alarm. |
| 2 | Get Credential Count for Enrolled Credentials | getcount | To get the count of already enrolled templates and credentials for a user on the selected device. |
| 3 | Acknowledge Alarm | acknoledgealarm | To command the device to acknowledge an alarm without clearing it. |
| 4 | Lock Door | lockdoor | To command the door to return to a locked state. |
| 5 | Unlock Door | unlockdoor | To command the door to return to an unlocked state. |
| 6 | Normalize Door | normalizedoor | To command the door to return to a normal state. |
| 7 | Get User Count on Device | getusercount | To obtain the total number of users added on a device. |
| 8 | Get Current Event Sequence Number | geteventcount | To get the current event sequence number and roll over count in a device. |
| 9 | Default the System Configuration | systemdefault | To set all the configurations on the device to default status. |
| 10 | Delete Credentials for All Users | deletecredential | To delete all biometric credentials of users from device. |

### For action=getcount

For valid values of this method, refer to the following argument-value table.

**Table: Get Credential Count Command - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| user-id | 1 to max. User ID in the door (2 bytes) | Yes | Defines the numeric ID of the user whose data is to be fetched. |
| card-count | 0 = 1 Card<br>1 = 2 Cards<br>2 = 3 Cards<br>3 = 4 Cards | No | To get the number of cards enrolled. |

**Table: Get Credential Count Command - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| finger-count | Single Template/Finger: 0-9<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers<br>5 = 6 Fingers<br>6 = 7 Fingers<br>7 = 8 Fingers<br>8 = 9 Fingers<br>9= 10 Fingers<br>Dual Template/Finger: 0-4<br>0 = 1 Finger<br>1 = 2 Fingers<br>2 = 3 Fingers<br>3 = 4 Fingers<br>4 = 5 Fingers | No | To get the number of fingers enrolled. |
| palm-count | 0 = 1 Palm<br>1 = 2 Palms<br>2 = 3 Palms<br>3 = 4 Palms<br>4 = 5 Palms<br>5 = 6 Palms<br>6 = 7 Palms<br>7 = 8 Palms<br>8 = 9 Palms<br>9 = 10 Palms | No | To get the number of palms enrolled. |
| format | text,xml | No | Specifies the format in which the response is expected. |

- *If no parameter is requested then all the count values will be returned by default (of supported credential types e.g. for PVR door, only card and palm template count will be returned).*
- *Palm template count and finger template counts depend on the device type i.e. Palm template count is only applicable for PVR doors and FP template counts are applicable for other devices. The specified credential should be applicable for the device on which the command is sent.*

### For action=deletecredential

For valid values of this method, refer to the following argument-value table.

**Table: Deleting Credentials for All Users - Parameters**

| Argument | Valid Values | Mandatory | Description |
|---|---|---|---|
| type | 0 = All<br>1 = Finger<br>2 = Palm | Yes | To specify the type of credential to be deleted. |

## Example

Following are some test cases for your reference:

1. **To get the current rollover count and sequence number of events in the device.**

**Sample Request**

```
http://<deviceIP:deviceport>/device.cgi/command?action=geteventcount&format=xml
```

**Sample Response**

```
HTTP Code: 200 OK
Content-Type: <xml>
Body:
<COSEC_API>
<Roll-over-count>1</roll-over-count>
<seq-number>1</seq-number>
</COSEC_API >
```

# Error Responses

These are some possible error response types obtained from incorrect API requests.

- **Argument is mentioned in request but valid value is not assigned.**

**Sample Response**

```
HTTP code: <code>
Content-type: <type>
Body:
Request failed: Incomplete command “<argument>=”
```

- **Invalid value is assigned to argument in request.**

**Sample Response**

```
HTTP code: <code>
Content-type: <type>
Body:
Request failed: Invalid command “<argument>=<invalid value>”
```

- **Syntax of request is incorrect or any unexpected arguments are received.**

**Sample Response**

```
HTTP code: <code>
Content-type: <type>
Body:
Request failed: Invalid syntax “<entire request>”
```

- **Mandatory fields are not mentioned in request.**

**Sample Response**

```
HTTP code: <code>
Content-type: <type>
Body:
Request failed: Incomplete command “<entire request>”
```

- **Syntax of request is valid but no data found.**

**Sample Response**

```
HTTP code: <code>
Content-type: <type>
Body:
Request failed: No record found “<argument>=<value>”
```

# API Response Codes

These numerical codes will be returned with an API response. These response codes shall indicate the result of a particular request made by the client. For e.g. the response code ‘0’ will indicate that the requested action was performed successfully. Refer to the given table for a list of response codes and their meanings.

**Table: API Response Codes**

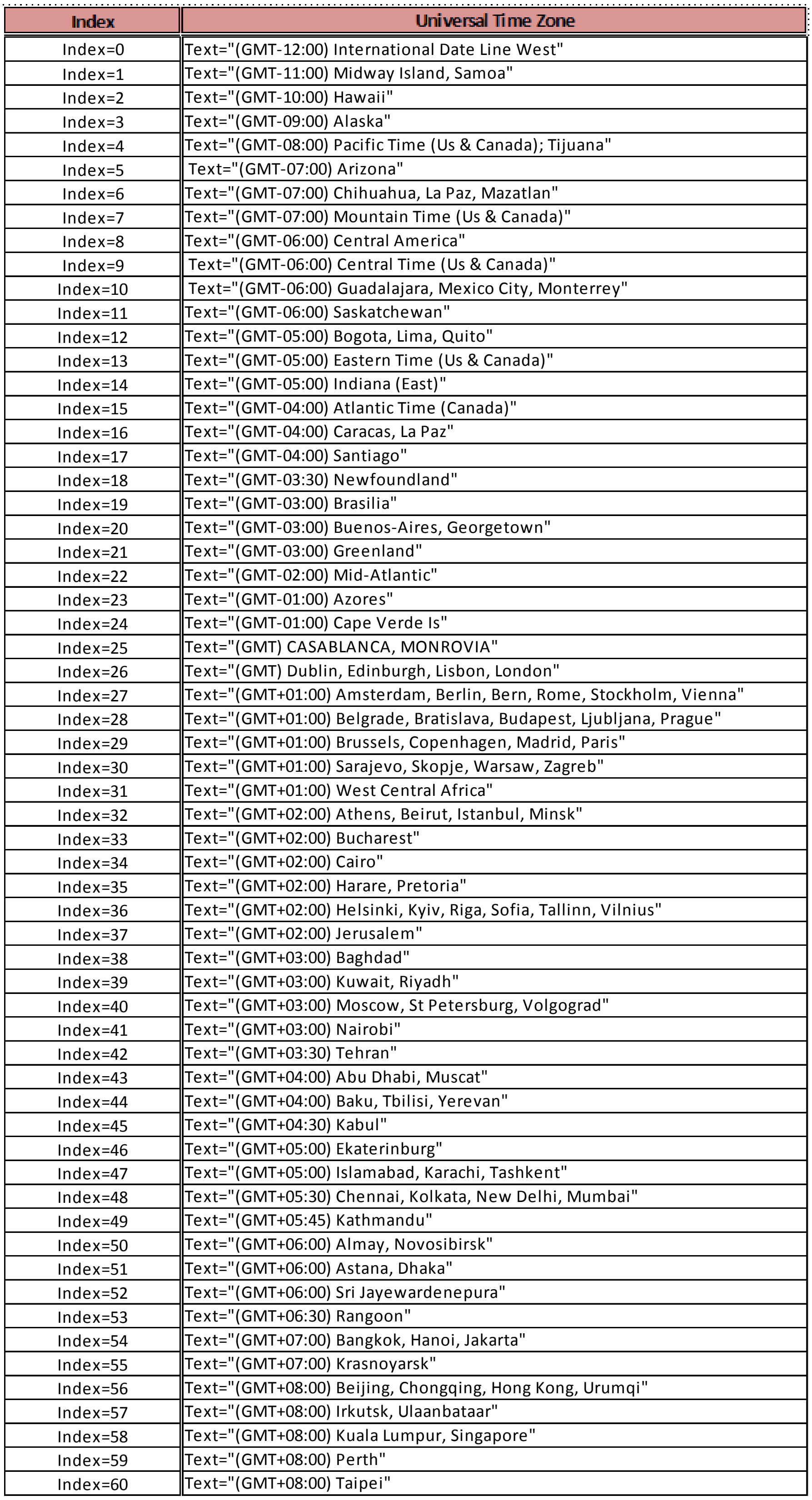
| Response Code | Description | Test Condition |
|---|---|---|
| 0 | Successful | - |
| 1 | Failed - Invalid Login Credentials | On every Authentication/Verification while logging In |
| 2 | Date and time – manual set failed | If unable to set the RTC for date and time API |
| 3 | Invalid Date/Time | In User API, if validity-date or date of birth is set wrong.<br>If the starting time and end time of a shift is configured as same. |
| 4 | Maximum users are already configured. | On every set command for user API |
| 5 | Image – size is too big | On every set command for user API |
| 6 | Image – format not supported | On every set command for user API |
| 7 | Card 1 and card 2 are identical | On every set command for user API and set credential API |
| 8 | Card ID exists | On every set command for user API and set credential API, Set Special Function API |
| 9 | Finger print template/ Palm template already exists | Set credential API |
| 10 | No Record Found | Event sequence number and roll over count not found, user id not found in Set User API |
| 11 | Template size/ format mismatch | If the expected template size is not as per the required size, format or any checksum error etc. in Set credential API |
| 12 | FP Memory full | In Set credential API, if the max FP template is set in the module. |
| 13 | User id not found | In enroll user command if user id is not available in the device and in User Configuration API, to update a user if provided reference user ID doesn’t belong to that user verified with alphanumeric user ID. |
| 14 | Credential limit reached | In enroll user command, if max no. of credentials is already enrolled. |
| 15 | Reader mismatch/ Reader not configured | The enroll request is for smart card and the device has proximity reader or if enroll request has palm template but door has finger reader and similar cases. |
| 16 | Device Busy | All cases of enrollment when the device is unable to process a request as it is in a different menu state |
| 17 | Internal process error | Internal error like configuration, firmware or event or calibration failure occur |
| 18 | PIN already exists | Set User API: PIN is already assigned to another user |
| 19 | Biometric credential not found | In enroll user smart card, write FP is enabled, but FP is not enrolled, Get FP/Palm template command is sent but template is not present. |
| 20 | Memory Card Not Found | In case memory card is not connected, and a command related to getting an image (user photo) is sent. |

**Table: API Response Codes**

| Response Code | Description | Test Condition |
|---|---|---|
| 21 | Reference User ID exists | When an already existing User ID is entered against a user having unique User ID. |
| 22 | Wrong Selection | For enrolling user, if writing FP template on smart card is enabled, but no fingerprint is enrolled.<br>When palm/finger/card count exceeds the maximum number of available places. |

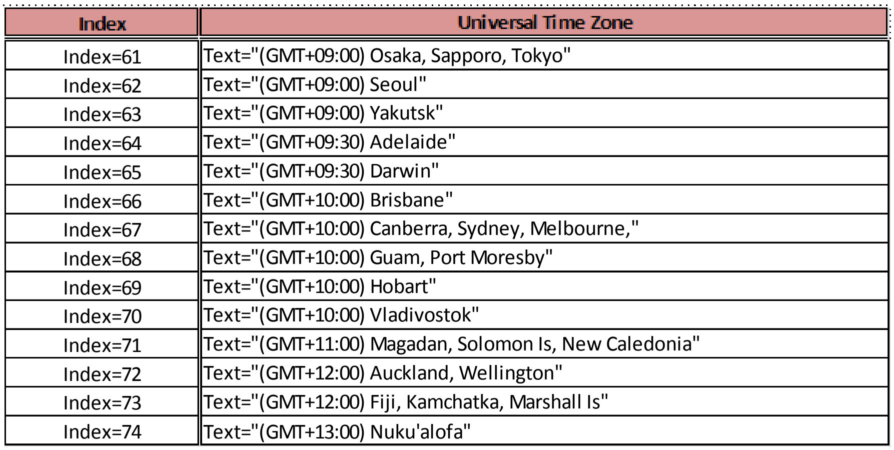
# Appendix

### 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 |
| 102 | User Allowed – with Duress |
| 103 | User Allowed – Anti-Pass Back-soft |
| 104 | User Allowed - Dead-man Zone |
| 105 | User Allowed – Door Not open |
| 106 | User Allowed – Smart Secure Access |
| 107 | User Allowed – Smart card based route access - soft |
| 108 | User Allowed – Panel route access - soft |
| 109 | User Allowed – two person rule - primary user |
| 110 | User Allowed – two person rule - secondary user |
| 151 | User Denied – User Invalid |
| 152 | User Denied – Occupancy Control |
| 153 | User Denied – 2-Person Rule |
| 154 | User Denied – Time Out |
| 155 | User Denied – Visitor Escort Rule |
| 156 | User Denied – Anti-Pass Back |
| 157 | User Denied – Disabled User |
| 158 | User Denied – Blocked User |
| 159 | User Denied – First IN User |
| 160 | User Denied – DND Enabled |
| 161 | User denied – Control zone |
| 162 | User Denied – Door Lock |

**Table: List of Events**

| Event ID | Event Description |
|---|---|
| 163 | User Denied – Invalid Access Group |
| 164 | User Denied – Validity date expired |
| 165 | User Denied – Invalid Route Access |
| 166 | User Denied – Invalid Shift Access |
| 201 | Door Status changed |
| 202 | Dead-man timer changed |
| 203 | DND status changed |
| 204 | Aux input status changed |
| 205 | Aux output status changed |
| 206 | Door sense input status |
| 207 | Door Controller Communication status |
| 301 | Dead-man timer expired Alarm– User IN |
| 302 | Duress detection |
| 303 | Panic Alarm |
| 304 | FP Memory Full – Alarm |
| 305 | Door Held open too long |
| 306 | Door Abnormal |
| 307 | Door force open |
| 308 | Door Controller Offline |
| 309 | Door Controller -Fault |
| 310 | Tamper Alarm |
| 311 | Master Controller Mains fail Alarm |
| 312 | Master Controller Battery fail |
| 313 | Master Alarm – MC Alarm input |
| 314 | RTC |
| 315 | Event Buffer Full |
| 351 | Alarm acknowledged |
| 352 | Alarm cleared |
| 353 | Alarm Re-issued |
| 401 | User Block/Restore |
| 402 | Login to ACS |
| 403 | Message transaction confirmation to ACMS |
| 404 | Guard Tour-status |
| 405 | Enrolment |
| 406 | Master Alarm sense input status |
| 407 | Master Aux Output status |

### Table: List of Events

| Event ID | Event Description |
|---|---|
| 408 | Input Output Group Link status |
| 409 | Credentials Deleted |
| 410 | Time Triggered Function |
| 411 | Time Stamping Function |
| 412 | Guard tag |
| 413 | Camera Event for time stamp |
| 451 | Configuration Change |
| 452 | Roll over of events |
| 453 | Master Controller Power ON |
| 454 | Configuration Defaulted |
| 455 | Soft Override |
| 456 | Backup and Update |
| 457 | Default System |
| 458 | Sensor Calibration |

*Some of the events listed are applicable only on Panels/Panel Doors and not on Direct Doors. Refer the respective event tables to see the applicable doors for each event.*

### Table: Size of Event Fields

| Door | Field 1 | Field 2 | Field 3 | Field 4 | Field 5 | Event Log Capacity |
|---|---|---|---|---|---|---|
| Direct Door V2 | 4 bytes | 2 bytes | 2 bytes | N.A. | N.A. | 50,000 events |
| Path Controller | 4 bytes | 2 bytes | 2 bytes | N.A. | N.A. | 50,000 events |
| Wireless Door | 4 bytes | 2 bytes | 2 bytes | 4 bytes | 4 bytes | 5,00,000 events |
| NGT Direct Door | 4 bytes | 2 bytes | 2 bytes | 4 bytes | 4 bytes | 1,00,000 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 |

### Table: User Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Event ID | (Field 1)<br>User ID | (Field 2)<br>Special Code | (Field 3)<br>Entry/Exit | (Field 4) | (Field 5) | Direct Door V2 | Path Controller | Wireless Door | NGT Door | PVR Door | Vega Controller |
| User Allowed Events | | | | | | | | | | | |

## Table: User Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Event ID** | **(Field 1)**<br>**User ID** | **(Field 2)**<br>**Special Code** | **(Field 3)**<br>**Entry/Exit** | **(Field 4)** | **(Field 5)** | **Direct Door V2** | **Path Controller** | **Wireless Door** | **NGT Door** | **PVR Door** | **Vega Controller** |
| 101 | Xxxx<br>(user ID=0 for REX input) | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 102 | Xxxx | Special Function code | Detail | 0 | 0 | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| 103 | Xxxx | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 104 | Xxxx | Special Function code | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 105 | Xxxx | Special Function code | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 106 | Xxxx | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 107 | Xxxx | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 108 | Xxxx | Special Function code | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 109 | Xxxx | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 110 | Xxxx | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| User Denied Events | | | | | | | | | | | |
| 151 | (User ID = 0 if not identified) | Special Function code | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 152 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 153 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 154 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 155 | Xxxx | 0 | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 156 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 157 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 158 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 159 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 160 | Xxxx | 0 | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 161 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 162 | Xxxx | 0 | Detail | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 163 | Xxxx | 0 | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 164 | Xxxx | 0 | Detail | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## Table: User Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Event ID | (Field 1)<br>User ID | (Field 2)<br>Special Code | (Field 3)<br>Entry/Exit | (Field 4) | (Field 5) | Direct Door V2 | Path Controller | Wireless Door | NGT Door | PVR Door | Vega Controller |
| 165 | Xxxx | 0=Door Not in Sequence<br>1=Door Not in Route | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | | 2=Door Not in Sequence for Smart card based Route<br>3=Door Not in Smart card based Route<br>4=Credential Invalid for Smart card based Route Access | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 166 | Xxxx | 0=Outside working hours<br>1=Holiday | Detail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | | 2=Week off<br>3=Field Break<br>4=Rest Day | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## 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 | Clock - IN Marking in T&A | 5 | ✓ | ✗ |
| 6 | Clock - OUT Marking in T&A | 6 | ✓ | ✗ |
| 7 | Post Lunch-IN Marking in T&A | 7 | ✓ | ✗ |

**Table: Special Function Codes Reference**

| S.No. | Special Function Name | Special Function Code | Applicable for Allowed Events | Applicable for Denied Events |
|---|---|---|---|---|
| 8 | Pre Lunch -OUT 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 | ✓ | ✗ |
| 13 | Access in Degrade Mode Marking | 99 | ✓ | ✓ |
| 14 | Smart Identification | 98 | ✗ | ✓ |
| 15 | e-Canteen | 97 | ✗ | ✓ |

**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 | | | | | | | Group | Palm | Finger | Card | PIN | RFU | | RFU | 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 8**

| Credential | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Value |
|---|---|---|---|---|---|---|
| PIN | 0 | 0 | 0 | 0 | 1 | 1 |
| Card | 0 | 0 | 0 | 1 | 0 | 2 |
| Card + PIN | 0 | 0 | 0 | 1 | 1 | 3 |
| Finger | 0 | 0 | 1 | 0 | 0 | 4 |

### Table: Information of Bit 4 and Bit 8

| Credential | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Value |
|---|---|---|---|---|---|---|
| Finger + PIN | 0 | 0 | 1 | 0 | 1 | 5 |
| Finger + Card | 0 | 0 | 1 | 1 | 0 | 6 |
| Finger + Card + PIN | 0 | 0 | 1 | 1 | 1 | 7 |
| Finger + Card | 0 | 0 | 1 | 1 | 0 | 6 |
| Finger + Card + PIN | 0 | 0 | 1 | 1 | 1 | 7 |
| Finger + Card | 0 | 0 | 1 | 1 | 0 | 6 |
| Finger + Card + PIN | 0 | 0 | 1 | 1 | 1 | 7 |
| Palm | 0 | 1 | 0 | 0 | 0 | 8 |
| PIN + Palm | 0 | 1 | 0 | 0 | 1 | 9 |
| Card + Palm | 0 | 1 | 0 | 1 | 0 | 10 |
| PIN + Card + Palm | 0 | 1 | 0 | 1 | 1 | 11 |
| Group + Palm | 1 | 1 | 0 | 0 | 0 | 24 |

### Table: Door Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Event ID | (Field 1)<br>Status | (Field 2) | (Field 3) | (Field 4) | (Field 5) | Direct Door V2 | Path Controller | Wireless Door | NGT Door | PVR Door | Vega Controller |
| 201 | 1= Normal<br>2= Locked<br>3= Unlocked | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 202 | 4= Activated<br>5= Deactivated | 0 | 0 | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 203 | 4= Activated<br>5= Deactivated | 0 | 0 | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 204 | 4= Activated<br>1= Normal<br>6= Fault (open)<br>7= Fault (short)<br>11= Disabled | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 205 | 4= Activated<br>1= Normal<br>11= Disabled | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 206 | 1= Normal<br>6= Fault (open)<br>7= Fault (short)<br>11= Disabled | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 207 | 0 | 0 | 1= ON Line<br>0= OFF Line | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

## Table: Alarm Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Event ID** | **(Field 1)** | **(Field 2)** | **(Field 3)** | **(Field 4)** | **(Field 5)** | **Direct Door V2** | **Path Controller** | **Wireless Door** | **NGT Door** | **PVR Door** | **Vega Controller** |
| 301 | User ID Xxxx | 1 = Critical | Alarm Seque-nce Number | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 302 | User ID Xxxx | 1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 303 | User ID Xxxx | 1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 304 | 1= Internal<br>2= External | 3 = Minor | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 305 | 0 | 3 = Minor | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 306 | 0 | 2 = Major | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 307 | 0 | 1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 308 | 0 | 2 = Major | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 309 | 0 | 2 = Major | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 310 | 0 | 1 = Critical | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 311 | 0 | 2 = Major | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 312 | 0 | 1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 313 | 0 | 1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 314 | 1= Power ON/ OFF Detected (time not in sync)<br>2= low battery detected<br>3= RTC Not Detected | 2 = Major<br>1 = Critical | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 315 | 0 | 2 = Major<br>1 = Critical | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 351 | 0 | 4 = SysInterlock<br>5 = User_Jeeves<br>6 = User_ACMS<br>9 = Auto | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 352 | 0 | 4 = SysInterlock<br>5 = User_Jeeves<br>6 = User_ACMS<br>7= Special Function | Same as above | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 353 | 0 | 0 | Same as above | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

## Table: System Events

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Event ID** | **(Field 1)** | **(Field 2)** | **(Field 3)** | **(Field 4)** | **(Field 5)** | **Direct Door V2** | **Path Controller** | **Wireless Door** | **NGT Door** | **PVR Door** | **Vega Controller** |
| 401 | User ID: xxxx | 0= Unused (Restore User)<br>1=Absentee Rule<br>2=Unauthorized access<br>3=Usage count<br>4=Invalid PIN | 1= Blocked<br>0= Restored | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 402 | 0 | 5= SA<br>6= SE<br>7= Operator | 1=Success<br>0=Fail | 0 | 0 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 403 | Transaction ID: Xxxx | 0 | 1=Success<br>0=Fail | 0 | 0 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 404 | Guard Tour no. Xxxx + cycle no. | 0 | 1=Success<br>0=Fail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 405 | ID: Xxxx | 8 = User Card<br>9 = User Finger<br>10 = Special Cards<br>14 = Palm | 1= Card/FP/ Palm-1<br>2= Card/FP/ Palm-2<br>3 = Card-3<br>4 = Card-4 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 406 | 0 | 0 | 1=Normal<br>2=Fault (Open)<br>3= Fault(Short)<br>4= Activated | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 407 | 0 | 0 | 1=Normal<br>4=Activated | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 408 | I/O Link ID | 11 = Pulse<br>12 = Interlock<br>13 = Latch<br>15 = Toggle (only with activated event) | 1=Normal<br>4=Activated | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 409 | ID: Xxxx | 8 = User Cards<br>9 = User Fingers<br>14 = Palm | 5= Web Jeeves<br>6= ACMS<br>7= Special Function | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 410 | Time Triggered Function Id | 0 | 1=Normal/ Deactivated<br>4=Activated | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 411 | Time Stamping Function ID | 0 | 1=Normal/ Deactivated<br>4=Activated | 0 | 0 | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| 412 | Guard tour no. +cycle no. | Door Controller sequence no. | 1=Success<br>0=Fail | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 413 | event sequence number | roll over count | 1=Success<br>0=Fail | 0 | 0 | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| 451 | Configur-ation Table ID xxx | Index start | Index end | 0 | 0 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |

**Table: System Events**

| Event Details | | | | | | Applicable Devices | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Event ID | (Field 1) | (Field 2) | (Field 3) | (Field 4) | (Field 5) | Direct Door V2 | Path Controller | Wireless Door | NGT Door | PVR Door | Vega Controller |
| 452 | Roll over number 00 to 99 | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 453 | 0 | 0 | 0 | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 454 | Configuration Table ID xxx | Index start | Index end | 0 | 0 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 455 | Time Period = xxx (configured value)<br>(this field is used only with Overridden events)<br>Resume events will have blank | 1= 2-person Rule<br>2= Access Policies<br>3= Alarms<br>4= Anti-pass back<br>5= First In User<br>6= Mantrap<br>7= Occupancy control<br>8= Visitor Escort Rule | 1= Overridden<br>0= Resumed | 0 | 0 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 456 | 1=Backup<br>2=Update | 1=Configuration<br>2=Event<br>3=Firmware | 0 = Fail<br>1 = Success<br>2 = CRC Check Fail | 0 | 0 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 457 | 0 | 0 | 6 = from ACMS<br>8 = from Hardware | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 458 | 0 | 0 = Internal Finger Reader<br>1 = External Finger Reader | 0 = Fail<br>1 = Success<br>2 = Not Supported | 0 | 0 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |

MATRIX
SECURITY SOLUTIONS

**MATRIX COMSEC PVT. LTD.**

**Corporate Office:**
394-GIDC, Makarpura, Vadodara - 390010, India.
Ph.:+91 265 2630555, Fax: +91 265 2636598
E-mail: Info@MatrixComSec.com

**Manufacturing Unit:**
15 & 19 GIDC, Waghodia - 391760, Dist. Vadodara, India.
Ph.: +91 2668 263172/73

**Customer Care:**
Ph.: +91 265 2630555
E-mail: Customer.Care@MatrixComSec.com, Support@MatrixComSec.com

www.MatrixComSec.com