

# DAHUA ACCESS CONTROL PRODUCTS INTEGRATION INSTRUCTION



Empower



## Revision Record

Version	Revision Content	Release Time	Updated by
V1.0	Creation	2021.09.28	Claire Xu



## Contents

- Wiegand Integration ..... 1
- OSDP Integration..... 1
- ONVIF Integration ..... 2
- SDK Integration..... 3
- CGI Integration ..... 4
  - ❑ Enroll a new user..... 4
  - ❑ Enroll face template ..... 6
  - ❑ Subscribe to real-time events ..... 7
  - ❑ Remote open/close door ..... 9
  - ❑ Get offline access records from device ..... 10
  - ❑ Get real-time video stream from device ..... 13





## Wiegand Integration

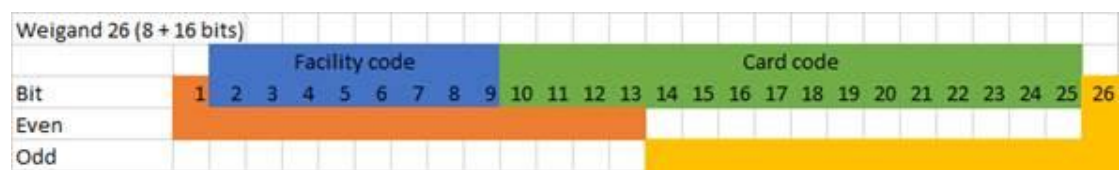
- **Supported models:**

ASI3XXX, ASI6XXX, ASI7XXX, ASC1XXX, ASC2XXX series.

Wiegand 34 by default, customizable wiegand format range: 24-66 bits

For customization, please send us:

1. The data format of the third-party device following below example of wiegand 26.



2. The card number need to be read.

## OSDP Integration

- **Supported models:**

ASC 2XXX series except ASC2204C-H

- **Main functions:**

OSDP V2 features, read physical card number



## ONVIF Integration

- **Supported models:**

ASI3XXX, ASI6XXX, ASI7XXX series.

- **Supported Functions:**

Currently, Dahua support ONVIF Profile A and C

For door control and event management, support below functions:

- Site information and configuration
- Event and alarm management
- Door access control

Profile C is for products used in an electronic access control system. Profile C conformant devices and clients support site information, door access control, and event and alarm management.



## SDK Integration

- **Supported Models:**

All Dahua Access control products

- **Main functions:**

- Enroll a new user
- Enroll face template (only works for products which can support face recognition, that is, ASI3XXX, ASI6XXXX, ASI7XXX series)
- Subscribe to real-time events
- Remote open/close door
- Get offline access records from device

### SDK Download

**Download link:**

<https://www.dahuasecurity.com/support/downloadCenter/software?id=2&child=3>

You can refer to the demo:

General\_NetSDK\_Eng\_Win64\_IS\_V3.055.0000000.0.R.210524\demo\MfcDemo\09.AccessContro  
l1s



## CGI Integration

- **Supported models:**  
ASI3XXX, ASI6XXX, ASI7XXX series.

### Enroll a new user

#### Request

<b>Template</b>	http://<server>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard			
<b>Method</b>	GET			
<b>Parameter Format</b>	key=value format in URL			
<b>Parameter</b>	Type	Required	Description	Example
<b>name</b>	string	Yes	The access control card and fingerprint record name of the user is fixed as "AccessControlCard".	"AccessContro ICard"
<b>CardName</b>	string	No	Card name, up to 32 characters	ZhangSan
<b>CardNo</b>	string	Yes	Card No.	12345
<b>UserID</b>	string	Yes	User ID	10000
<b>CardStatus</b>	integer	No	Card status: 0: Normal. 1<<0: Reported for loss. 1<<1: Canceled. 1<<2: Frozen. 1<<3: Arrearage. 1<<4: Overdue. 1<<5: Pre-arrearage (The door still can be unlocked with a voice prompt).	0
<b>CardType</b>	integer	No	Card type (only for unlocking by card): 0: Ordinary card. 1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card. 0xff is mother card.	0
<b>Password</b>	string	No	Card + password: The password when unlocking with card + password (The video intercom device does not support this field)	123456
<b>Doors</b>	array<integer>	No	Door permission: An array, with the value corresponding to	





			the subscript of AccessControl. (The video intercom device does not support this field)	
<b>TimeSections</b>	array<integer>	No	Periods corresponding to door permission: An array, in which each element corresponds to the door in Doors; period index. (The video intercom device does not support this field)	
<b>VTOPosition</b>	string	No	Door number linked with indoor monitor (The video intercom device does not support this field)	01018001
<b>ValidDateStart</b>	string	No	Start time of validity period: The format is yyyyMMdd hhmss, such as "20151022 093811".	20151022%20093811
<b>ValidDateEnd</b>	string	No	End time of validity period: The format is yyyyMMdd hhmss, such as "20151022 093811".	20151222%20093811
<b>IsValid</b>	bool	No	Valid or not: When the validity period expires or the maximum number of usage is reached, this field is automatically changed to false. (The video intercom device does not support this field)	

### Example

```
http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard&CardName=ZhangSan&CardNo=12345&Use rID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition =01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811
```

## Response

<b>Parameter Format</b>	key=value format in body			
<b>Parameter</b>	Type	Required	Description	Example
<b>RecNo</b>	integer	Yes	Record number	12345
<b>Example</b>				
RecNo=12345				



**Enroll face template**

**Request**

<b>Template</b>	http://<server>/cgi-bin/FaceInfoManager.cgi?action=add			
<b>Method</b>	POST			
<b>Parameter Format</b>	JSON format in body			
<b>Parameter</b>	Type	Required	Description	Example
<b>UserID</b>	string	Yes	User ID	10000
<b>Info</b>	object	Yes	User information	
<b>+UserName</b>	string	No	Person name	ZhangSan
<b>+FaceData</b>	array<string>	No	Face template data, the string after Base64. The array length is limited to 20, and the length of a single element is limited to 2k (after Base64, including \0).	[ "xxxx", "xxxx", ... ]
<b>+PhotoData</b>	array<string>	No	Face image, the string after Base64. The array length is limited to 5, and the length of a single element is limited to 200K. Note: Either FaceData or PhotoData should exist in the request.	[ "yyyy", "yyyy", ... ]

**Example**

```
POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=add
Content-Type: application/json
Content-Length: <length>
{
  "UserID": "102",
  "Info": {
    "UserName": "ZhangSan",
    "FaceData": [ "xxxx", "xxxx", ... ],
    "PhotoData": [ "yyyy", "yyyy", ... ]
  }
}
```

**Response**

<b>Parameter Format</b>	OK at body			
<b>Parameter</b>	Type	Required	Description	Example
<b>Example</b>	OK			



● **Note:**

```
{
  "UserID": "102",
  "Info": {
    "UserName": "ZhangSan",
    "FaceData": [ "xxx", "xxx", ... ], // Face eigenvalue data need to be processed by Dahua algorithm
    "PhotoData": [ "yyy", "yyy", ... ] // data:image/jpeg;base64,/9j(The red part need to be removed after base64 processed)
  }
}
```

● **Face picture requirements:**

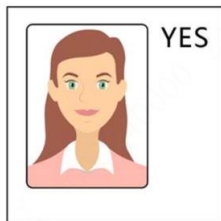
The picture format is jpg, the size is less than 100KB, the pixel range is between 150×300≤resolution≤600×1200, it is recommended that the picture pixel is about 500×500, and the picture name should be the same as user ID.

There can only be one face in the picture, the proportion of the face in the picture does not exceed 2/3 of the entire picture, and the ratio of width to height of the picture does not exceed 1:2.

Keep the face clean, and the beard should not change too much. The face needs to be with open eyes and natural expressions. It is recommended that the forehead should be exposed, and the hair should not cover the face.

Use regular light colors (not white, yellow, backlight, etc.), without covering the face (such as wearing hats, masks, glasses, etc.), and without retouching.

**Right Example:**



**Wrong Example: Side face/Full body/Look up:**



☐ **Subscribe to real-time events**

<b>Syntax</b>	http://<server>/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[<eventCode>,<eventCode>...][&channel=<ChannelNo>][&heartbeat=<Heartbeat>]
<b>Method</b>	GET
<b>Description</b>	Subscription snapshots: You can specify the snapshots when the subscribed <eventCode> event occurs.



<b>Example</b>	http://192.168.1.108/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[AccessControl]&heartbeat=5
<b>Success Return</b>	<pre>--&lt;boundary&gt;\r\n Content-Type: text/plain\r\n Content-Length: &lt;data length&gt;\r\n Events[0].Code=TrafficJunction Events[0].CountInGroup=1 Events[0].IndexInGroup=1 Events[0].Lane=1 Events[0].Data.PTS= 42949485818.0 Events[0].TrafficCar.PlateNumber=Z A12345 Events[0].TrafficCar.DeviceAddress=Hangzhou ..... Events[1].Code=TrafficJunction ..... --&lt;boundary&gt; Content-Type: image/jpeg Content-Length:&lt;image size&gt; &lt;JPEG image data&gt; --&lt;boundary&gt; Content-Type: text/plain Content-Length:&lt;data length&gt; Heartbeat --&lt;boundary&gt;</pre>
<b>Comment</b>	<p>ChannelNo: Video channel number, integer, starting from 1. The default value 1 is used if not specified.</p> <p>Heartbeat: Heartbeat interval, integer, in seconds; the range is [1–60]. If URL comes with this parameter, and the value is 5, it means that the device will send a heartbeat message to the client every five seconds, and the heartbeat message is a string "Heartbeat".</p> <p>eventCode: Event code, which can be the following events:</p> <p>AccessControl: Access control event.</p> <p>CitizenPictureCompare: Person and ID card comparison event.</p> <p>In addition, the value of eventCode can be All, which means all event types.</p> <p>Parameters in the response:</p> <p>GroupID: Event group, integer, namely the snapshot event group ID.</p> <p>CountInGroup: Number of events in the event group, integer.</p> <p>IndexInGroup: The serial number of this event in the event group, integer. For example, if CountInGroup is 3 and IndexInGroup is 1, it means that this event group has three events and snapshots, and this is the first event and snapshot.</p>



## Remote open/close door

### UNLOCK

#### Request

<b>Template</b>	http://<server>/cgi-bin/accessControl.cgi?action=openDoor&channel=<ChannelNo>[&UserID=<UserID>&Type=<Type>]			
<b>Method</b>	GET			
<b>Parameter Format</b>	key=value format in URL			
<b>Parameter</b>	Type	Required	Description	Example
<b>channel</b>	integer	Yes	Channel number, access control number, starting from 1	1
<b>UserID</b>	integer	No	User ID	101
<b>Type</b>	string	No	Unlocking method, "Remote" by default	Remote
<b>Example</b>	http://192.168.1.108/cgi-bin/accessControl.cgi?action=openDoor&channel=1&UserID=101&Type=Remote			

#### Response

<b>Parameter Format</b>	OK at body			
<b>Parameter</b>	Type	Required	Description	Example
<b>Example</b>	OK			

### LOCK

#### Request

<b>Template</b>	http://<server>/cgi-bin/accessControl.cgi?action=closeDoor&channel=<ChannelNo>[&UserID=<UserID>&Type=<Type>]			
<b>Method</b>	GET			
<b>Parameter Format</b>	key=value format in URL			
<b>Parameter</b>	Type	Required	Description	Example
<b>channel</b>	integer	Yes	Channel number, access control number, starting from 1	1
<b>UserID</b>	integer	No	Remote user ID	101



<b>Type</b>	string	No	Door type, "Remote" by default	Remote
<b>Example</b>				
http://192.168.1.108/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&UserID=101&Type=Remote				

### Response

<b>Parameter</b>	OK at body			
<b>Format</b>				
<b>Parameter</b>	Type	Required	Description	Example
<b>Example</b>				
OK				

## ❑ Get offline access records from device

### Request

<b>Template</b>	http://<server>/cgi-bin/recordFinder.cgi?action=find&name=<RecordName>			
<b>Method</b>	GET			
<b>Parameter</b>	key=value format in URL			
<b>Format</b>				
<b>Parameter</b>	Type	Required	Description	Example
<b>name</b>	string	Yes	The access control card and fingerprint record name of the user is fixed as "AccessControlCardRec".	"AccessControlCardRec"
<b>count</b>	integer	No	Maximum number returned, 1024 by default	100
<b>StartTime</b>	string	No	Start time of record creation	123456700
<b>EndTime</b>	string	No	End time of record creation	123456800
<b>condition</b>	object	No	Search conditions	—
<b>+CardNo</b>	string	No	Card No.	123456
<b>Example</b>				
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=123456700&EndTime=123456800&condition.CardNo=12001&count=100				

### Response

<b>Parameter</b>	key=value format in body			
<b>Format</b>				
<b>Parameter</b>	Type	Required	Description	Example
<b>totalCount</b>	integer	No	Total number of records found	1,000
<b>Found</b>	integer	No	Number of records returned	100
<b>Records</b>	array<object>	Yes	Records returned	—



# DAHUA ACCESS CONTROL PRODUCTS INTEGRATION INSTRUCTION

Smarter Together

<b>+RecNo</b>	integer	Yes	Record number	12345
<b>+CreateTime</b>	integer	No	Card swiping time, UTC time	123456789
<b>+CardNo</b>	integer	Yes	Card No.	12001
<b>+CardName</b>	string	No	Card name	ZhangSan
<b>+CardType</b>	integer	No	Card type (only for unlocking by card): 0: Ordinary card. 1: VIP card. 2: Guest card. 3: Patrol card. 4: Blocklist card. 5: Duress card. 0xff is mother card.	0
<b>+Password</b>	string	No	Password	123456
<b>+UserID</b>	string	Yes	User ID	ZhangSan
<b>+Type</b>	string	No	Event type: Enumchar[32]{ "Entry": Entry; "Exit": Exit }	Exit
<b>+Status</b>	integer	No	Card swiping result: Enumint{ 0: Failed; 1: Succeeded }	1
<b>+Method</b>	integer	Yes	Unlocking method: 0: By password. 1: By swiping card. 2: Use password after swiping card. 3: Swipe card after using password. 6: By fingerprint. 15: By (local) face recognition.	1
<b>+Door</b>	integer	No	Door number (The video intercom device does not support this field)	5
<b>+ReaderID</b>	string	No	Card reader ID (The video intercom device does not support this field)	—
<b>+ErrorCode</b>	integer	No	Unlocking failure error code, which is valid when the Status is 0.	—
<b>+URL</b>	string	No	Image URL, with length of up to 127. (The video intercom device does not support this field)	—
<b>+RecordURL</b>	string	No	Swiping card recording URL (The video intercom device does not support this field)	—
<b>+IsOverTemperature</b>	bool	No	Whether it is over temperature.	true
<b>+TemperatureUnit</b>	integer	No	Temperature unit (0: Celsius, 1: Fahrenheit, 2: Kelvin)	0
<b>+CurrentTemperature</b>	float	No	Body temperature	36.8
<b>+CitizenIDResult</b>	bool	No	If the similarity is larger than or equal to	true



			the threshold, the person and ID card comparison is successful.	
<b>+CitizenIDName</b>	string	No	Resident name	Zhang San
<b>+CitizenIDNo</b>	string	No	ID card number, 18 digits	3420000000 00000000
<b>+CitizenIDSex</b>	integer	No	Gender enumint8{0: Unknown 1: Male 2: Female 9: Unspecified}	1
<b>+CitizenIDEthnicity</b>	integer	No	Ethnic (Refer to the definition of the CitizenIDCard event)	1
<b>+CitizenIDBirth</b>	string	No	Date of birth	1980-01-01
<b>+CitizenIDAddress</b>	string	No	Address	No.1199 Bin'an Road
<b>+CitizenIDAuthority</b>	string	No	Issuing authority	Hangzhou Public Security Bureau
<b>+CitizenIDStart</b>	string	No	Start date of validity period	1996-01-01
<b>+CitizenIDEnd</b>	string	No	End date of validity period. "Endless" means long-term validity.	2006-01-01
<b>Example</b>				
<pre>totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].CardNo=12001 records[0].CardName=ZhangSan records[0].UserID=ZhangSan records[0].Type=Entry records[0].Method=1 records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].CardNo=12001 records[1].CardName=ZhangSan records[1].UserID=ZhangSan records[1].Type=Exit records[1].Method=1</pre>				

## Procedure

**Batch extraction of the offline unlocking records is complicated. Procedures:**

**Use the command of getting records as follows:**

```
http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=123456700&EndTime=153456800&condition.CardNo=12001&count=100
```

**The total number of records obtained totalCount, the number of records returned found, and**





**the CreateTime field value corresponding to the last record:**

```
totalCount=1000
found=100
....
records[99].RecNo=12345
records[99].CreateTime=140556698
records[99].CardNo=12001
records[99].CardName=ZhangSan
records[99].UserID=ZhangSan
```

**If totalCount > found, you need to enter the CreateTime field value corresponding to the last record obtained in the previous step in the StartTime field and update the EndTime field value accordingly, and then continue to use the getting command:**

```
http://192.168.1.108/cgi-
bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=140556698&EndTime=140586698&condition.CardNo=12001&count=100
```

**Get the next period.**

**Repeat the getting operations.**

**Until the total number of records returned is equal to the value of totalCount.**

**Note:**

In terms of this method, multiple records exist for the same timestamp, and there might be overlapping records in two returns. Therefore, you need to perform the de-duplication operation (to remove duplicate records) every time after the client gets data.

## Get real-time video stream from device

### Request

<b>URL</b>	rtsp://<server>:[port]/cam/realmonitor			
<b>Method</b>	DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ...			
<b>URL Params ( key=value format in URL )</b>				
<b>Parameter</b>	Type	Require d	Description	Example
<b>Channel</b>	int	Yes	Video channel number, starting from 1. The default value 1 is used if not specified.	1
<b>Subtype</b>	int	Yes	Stream type: Main stream and sub stream. The number of sub streams can be obtained by the command in "Getting Maximum Number of Sub Streams". Some optional values are as follows.	0



			The default value 0 is used if not specified. 0: Main stream. 1: Sub stream 1. 2: Sub stream 2.	
<b>URL Example</b>				
rtsp://192.168.1.108:554/cam/realmonitor?channel=1&subtype=1				