

Version Information

version	Change Description	Change person	date
1.0.0	Add the first edition of Interface Description	liyunfei	2019/9/10
1.0.1	Update content, add a core business process diagrams	liyunfei	2018/9/12
1.0.2	Change to English document	liyunfei\zhangheng	2020/01/09

Schedule 1:

Explanations on transaction ID		
Definition	Type is unified as unsigned long long type (u64) .	
High 32-bit represents request source, client side generation	No. 61-64bit	4bit client side type (range 0-15) : 0-others 1-NVR main program 2-GUI 3-network client side 4-CGI client side 5-trade external hanging
	No. 33-60bit	28bit client side sole identification: when the request is initiated each time, the client side itself generates a non-repetitive identification code for distinguishing multiple client sides of same type.
Low 32-bit represents transaction, main program human face module distribution	No. 25-32bit	8bit transaction type (range 0-255) : 0-others 1-video stream analysis 2-picture stream analysis 3-single picture analysis 4-feature calculation
	No. 1-24bit	24bit (frame/picture/request) No.: cyclic increase

Content

1	Overview.....	2
	1.1.1 Public request header.....	3
	1.1.2 Public reponse header.....	3
	Interface use instructions and examples.....	3
	2.7.47 Acquire parameters of face recognition library in batch.....	3
	2.7.46 Add、Set、delete face recognition library.....	4
	2.7.75 Face library deletion progress.....	7
	2.7.49 Export face library, request of starting export.....	8
	2.7.50 Import/export face library progress.....	8
	2.7.60 Export face picture.....	9
	2.7.48 Import face library.....	10
	2.7.52 Synchronize face library to frontend.....	11
	2.7.51 Status of synchronizing face library to frontend.....	12
	2.7.69 Remodel face library.....	14
	2.7.55 Query face information.....	16
	2.7.56 Query by picture and analyze the face in group photo.....	21
	2.7.57 Query analyzed face.....	21
	2.7.54 Add face picture, import picture firstly.....	23
	2.7.53 Add face picture, import picture firstly.....	26
	2.7.44 Acquire enabling status of face configuration channel.....	27
	2.7.61 Release cache data of affair.....	28
	2.7.56 Query by picture and analyze the face in group photo.....	29
	2.7.57 Query analyzed face.....	29
	2.7.59 Search face information by picture.....	30
	2.7.66 Acquire NVR performance.....	33
	2.10.1 Make request for file uploading.....	35
2	Interface call flow chart.....	35
	3.1 Find by image.....	36
	3.2 Face basemap added.....	38
3	Response status code.....	38
	4.1 Business response status code.....	38
	4.2 Response body.....	38

1 Overview

This document is a description of the application interface related to the face database. It mainly implements operations such as adding, deleting, modifying,

and checking the face database, and adding, deleting, modifying, and checking the face basemap..

This document is suitable for developers who use the web-API interface (CGI protocol) to connect with Tiandy's intelligent face-related products. The docking vendors can find and use related interfaces to implement different functions according to their needs. For the specific usage of the interface, refer to the interface description below..

Encoding : UTF-8

Request agreement : HTTP

Request method : GET/PUT/POST/DELETE

1.1.1 Public request header

Public request header (excluding other HTTP request headers)

Header	Description
Content-Type	application/xml;charset=UTF-8
Accept	application/xml;charset=UTF-8

1.1.2 Public response header

Public response headers (excluding other HTTP request headers)

Header	Description
Content-Type	application/xml;charset=UTF-8
Accept	application/xml;charset=UTF-8

Interface use instructions and examples

2.7.47 Acquire parameters of face recognition library in batch

url: /CGI/Smart/FaceLib/Manage	
GET	
Description	Acquire parameters of face recognition library in batch
Query	None

Inbound Data	None
Success Return	<FaceLibInfo>
Explanations on protocol: Acquire parameters of face recognition library in batch Explanations on key parameters: <facePicModelNum> Picture amount of established models <facePicNum> Total number of base map <key> key value of face library <name> Face library name, 64 bits at most <descripInfo> Description information, 64 bits at most <lock> Whether locked; 1: Locked; 0: Unlocked	

FaceLibParas XML Block

```

<?xml version="1.0" encoding="UTF-8"?>
<faceLibInfo>
<facePicModelNum><!-- req, xs: integer --></facePicModelNum>
<facePicNum><!-- req, xs: integer --></facePicNum>
<faceLibParasList>
<faceLibParas>
<key><!-- req, xs: integer --></key>
<name><!-- req, xs:string --></name>
<descripInfo><!-- dep, xs:string --></descripInfo>
<lock><!-- req, xs: integer --></lock>
</faceLibParas>
//...Repeat <faceLibParas > Structure
</faceLibParasList>

```

Test cases

GET /CGI/Smart/FaceLib/Manage

Request XML: none

Response XML: <FaceLibParas>

```

<?xml version="1.0" encoding="UTF-8"?>
<faceLibInfo>
<facePicModelNum>100</facePicModelNum>
<facePicNum>500</facePicNum>
<faceLibParasList>
<faceLibParas>
<key>2</key>
<name> R&D library</name>
<descripInfo> R&D staff library</descripInfo>
<lock>1</lock>
</faceLibParas>
...
</faceLibParasList>
</faceLibInfo>

```

2.7.46 Add、Set、delete face recognition library

url:	
/CGI/Smart/FaceLib/<Key>/Manage	
POST	
Description	Add face recognition library
Query	None

Inbound Data	<FaceLibParas>
Success Return	<FaceLibResult>
PUT	
Set face recognition library	Set face recognition library
Query	None
Inbound Data	<FaceLibParas>
Success Return	<ResponseStatus>
DELETE	
Description	Delete face recognition library
Query	None
Inbound Data	None
Success Return	<ResponseStatus>
Explanations on protocol: Acquire, set, add and delete parameters of face recognition library and check if library has lock and password. Introduction to key parameters: URL <Key> key value of face library; key=0 means adding of face library; key>0 means setting of face library parameters; long: Long type Request xml <name> Face library name, 64 bits at most <descripInfo> Description information, 64 bits at most <threshold> Face library dependability, >=0 <lock> Whether library is locked; 1: Locked; 0: Unlocked <access> Random information, for calibration use; 32 bits <password> Lock password, encrypted, 16 bits at most Response xml <key> key value of face library, responded when adding value. -1 means failure	

FaceLibParas XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<faceLibParas>
  <name><!-- req, xs:string --></name>
  <descripInfo><!-- dep, xs:string --></descripInfo>
  <threshold><!-- dep, xs: integer --></threshold>
  <picstreamEnable><!--req, xs:boolean--></picstreamEnable>
  <lock><!-- req, xs: integer --></lock>
  <access><!-- req, xs:string --></access>
  <password><!-- req, xs:string --></password>
</faceLibParas>
```

ResponseResult XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<faceLibResult>
  <key><!-- dep, xs: integer --></key>
</faceLibResult>
```

Test cases

POST /CGI/Smart/FaceLib/0/Manage

Request XML: <FaceLibParas>

```
<?xml version="1.0" encoding="UTF-8"?>
<faceLibParas>
<key>2</key>
<name> R&D library</name>
<descripInfo> R&D staff library</descripInfo>
<threshold>80</threshold>
<picstreamEnable>>false</picstreamEnable>
<lock>1</lock>
<access>94AAABB419A9820DC171B43240CEE41</access>
<password>T6g05arqzu4=</password>
</faceLibParas>
```

Response XML: <FaceLibResult>

```
<?xml version="1.0" encoding="UTF-8"?>
<responseResult>
<key>456</key>
</responseResult>
```

PUT /CGI/Smart/FaceLib/123/Manage

Request XML: <FaceLibParas>

```
<?xml version="1.0" encoding="UTF-8"?>
<faceLibParas>
<key>2</key>zr
<name> R&D library</name>
<descripInfo> R&D staff library</descripInfo>
<threshold>70</threshold>
<picstreamEnable>ture</picstreamEnable>
<lock>1</lock>
<access>94AAABB419A9820DC171B43240CEE41</access>
<password>T6g05arqzu4=</password>
</faceLibParas>
```

Response XML: <ResponseStatus>

DELETE /CGI/Smart/FaceLib/123/Manage

Request XML L: none

Response XML: <ResponseStatus>

```
<!-- Face library added -->
<!--POST /CGI/Smart/FaceLib/<Key>/Manage-->
<!--<Key>use 0-->
<!--request-->
<faceLibParas>
    <name><!-- Library name --></name>
    <descripInfo><!-- Library description --></descripInfo>
    <threshold><!-- Face library confidence --></threshold>
    <picstreamEnable><!-- Upload identification information --></picstreamEnable>
</faceLibParas>
<!--response-->
<faceLibResult>
```

```

        <key>14</key>
    </faceLibResult>

<!-- Face library editing -->
<!--Put /CGI/Smart/FaceLib/<Key>/Manage-->
<!--<Key> Use the corresponding library Id -->
<!--request-->
<faceLibParas>
    <name><!-- Library name --></name>
    <descripInfo><!-- Library description --></descripInfo>
    <threshold><!-- Face library confidence --></threshold>
    <picstreamEnable><!-- Upload identification information --></picstreamEnable>
</faceLibParas>

<!-- Face library deletion -->
<!--Delete /CGI/Smart/FaceLib/<Key>/Manage-->
<!--<Key> Use the corresponding library Id -->
<!--response Universal-->
<!-- Callable   /CGI/Smart/FaceLib/<Key>/Del/Progress View progress -->

```

2.7.75 Face library deletion progress

url: /CGI/Smart/FaceLib/<key>/Del/Progress	
GET	
Description	Face library deletion progress
Query	None
Inbound Data	None
Success Return	<Progress>
Explanations on protocol: Deletion progress of face library Explanations of parameters: URL: <key>: Face library key value Reply xml: <state> State; 0-Undeleted; 1-Delete in progress; 2-Deleted; 3-Deletion fails <pro> Progress, 0-100	

Progress XML Block

```

<?xml version="1.0" encoding="UTF-8"?>
<progress version="1.0">
<state><!--req,sx:integer--></state>
<pro><!--req,sx:integer--></pro>

```

</progress>

Test cases

GET /CGI/Smart/FaceLib/<key>/Del/Progress

Response XML: <Progress>

```
<?xml version="1.0" encoding="UTF-8"?>
<progress version="1.0">
<state>1</state>
<pro>50</pro>
</progress>
```

2.7.49 Export face library, request of starting export

url:

/CGI/Smart/StartExport/FaceLib/<Key>/SessionId/<ID>/File/<FileName>/Access/<Access>/Password/<Password>

PUT

Description	Export face library, request of starting export
-------------	---

Query	None
-------	------

Inbound Data	None
--------------	------

Success Return	<ResponseStatus>
----------------	------------------

Explanations on protocol:

Export face library

Explanations of parameters:

<Key> key value of face library

<sessionId> client side and equipment interaction ID, refer to Schedule 1

<FileName> Export filename, 128 bits

<Access> means random information, for calibration use, 32 bits

<PassWord> Export password, encrypted transmission, 16 bits

Test cases

PUT

/CGI/Smart/StartExport/FaceLib/123/SessionId/456/File/0.jpg/Access/94AAABB419A9820DC171B43240CEE41/Password/T6g05arqzu4=

Request XML: none

Response XML: <ResponseStatus>

<!--Face library starts to generate box-->

<!--Put /CGI/Smart/StartExport/FaceLib/<Key>/SessionId/<ID>/File/<FileName>-->

<!--<Key> Use the corresponding library Id, <ID> Depends on the current time to obtain a random decimal number between 0x4000 ~ 0x4fff, <FileName>box filename-->

<!--response Universal-->

<!-- Get progress agreement after /CGI/Smart/FaceLib/SessionId/<ID>/Progress, When the progress is 100, call again /CGI/Smart/Export/LocalFile/<FileName> Form download file -->

2.7.50 Import/export face library progress

url:

/CGI/Smart/FaceLib/SessionId/<ID>/Progress	
GET	
Description	Import/export face library progress
Query	None
Inbound Data	None
Success Return	<Progress>
Explanations on protocol: Progress of importing/exporting face library Explanations of parameters: URL: <sessionId> client side and equipment interaction ID, refer to Schedule 1 Reply xml: <state> State; 0-Not imported/exported; 1-Import/export in progress; 2-Import/export succeeds; 3-Export/import fails <pro> Progress, 0-100	

Progress XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<progress version="1.0">
<state><!--req,sx:integer--></state>
<pro><!--req,sx:integer--></pro>
</progress>
```

Test cases

GET /CGI/Smart/FaceLib /SessionId/123/Progress

Response XML: <Progress>

```
<?xml version="1.0" encoding="UTF-8"?>
<progress version="1.0">
<state>1</state>
<pro>50</pro>
</progress>
```

```
<!-- Face library generation box progress / Face library box package add basemap progress -->
<!--Get /CGI/Smart/FaceLib/SessionId/<ID>/Progress-->
<!--<ID> Depends on the current time to obtain a random decimal number between 0x4000 ~
0x4fff -->
<!--response-->
<progress>
    <stat><!--0-Not imported / exported, 1-Imported / exported, 2-Imported / exported
successfully, 3-Imported / exported failed --></stat>
    <pro><!-- Progress value, 0-100--></pro>
</progress>
```

2.7.60 Export face picture

url:

/CGI/Smart/Export/FacePic/<Key>/SessionId/<ID>/File/<FileName>

GET	
Description	Export face picture
Query	None
Inbound Data	None
Success Return	File content
Explanations on protocol: Export face picture Explanations of parameters: <key> Face key value <sessionId> client side and equipment interaction ID, refer to Schedule 1 <fileName> File name	

Test cases

GET /CGI/Smart/Export/FacePic/123/SessionId/456/File/0.jpg

Request XML: none

Response XML: File contents

```
<!-- Face library box file export -->
<!--Post /CGI/Smart/Export/LocalFile/<FileName>-->
<!--Post /CGI/Smart/Export/LocalFile/<FileName>/Access/<Access>/Password/<Password>
encryption -->
<!--<FileName>box filename,
/CGI/Smart/StartExport/FaceLib/<Key>/SessionId/<ID>/File/<FileName> Consistent with
protocol file name -->
<!--<Access> Secret key, <Password> password -->
<!-- Form download file -->
```

2.7.48 Import face library

url: /CGI/Smart/Import/FaceLib/<Key>/SessionId/<ID>/Access/<Access>/Password/<Password>	
PUT	
Description	Import face library
Query	None
Inbound Data	File content
Success Return	<ResponseStatus>
Explanations on protocol: Import of face library Explanations of parameters: <Key> key value of face library <sessionId> client side and equipment interaction ID, refer to Schedule 1 <Access> means random information, for calibration use, 32 bits <PassWord> Import password, encrypted transmission, 16 bits	

Test cases

PUT

/CGI/Smart/Import/FaceLib/123/SessionId/456/Access/94AAABB419A9820DC171B43240CEE41/Password/T6g05arqzu4=

Request XML: none

Response XML: <ResponseStatus>

<!-- Face library box file import -->
<!--Post /CGI/Smart/Import/FaceLib/<Key>/SessionId/<ID>-->
<!--Post
/CGI/Smart/Import/FaceLib/<Key>/SessionId/<ID>/Access/<Access>/Password/<Password>
encryption -->
<!--<Key> Use the corresponding library Id, <ID> Depends on the current time to obtain a
random decimal number between 0x4000 ~ 0x4fff -->
<!--<Access> Secret key, <Password> password -->
<!--response Universal-->
<!-- Callable /CGI/Smart/FaceLib/SessionId/<ID>/Progress View import progress-->

2.7.52 Synchronize face library to frontend

url: /CGI/Smart/FaceLib/<key>/SyncToIpc/Cmd	
PUT	
Description	Synchronize face library to frontend
Query	None
Inbound Data	<SyncCmd>
Success Return	<ResponseStatus>
Explanations on protocol: Synchronize face library to frontend and acquire status. Explanations of parameters: <channel> Channel <cmd> Command; start: 20; stop: 21; delete: 22	

SyncPara XML Block

```
<?xml version="2.0" encoding="UTF-8"?>  
<syncCmd version="2.0">  
  <elementList>  
    <element>  
      <channel><!--req,sx:integer--></channel>  
      <cmd><!--dep,sx: integer--></cmd>  
    </element>  
    ...  
  </elementList>  
</syncCmd>
```

Test cases

PUT /CGI/Smart/FaceLib/xxxx/SyncToIpc/Cmd

Request XML: <SyncCmd>

Response XML: <ResponseStatus>

```

<?xml version="2.0" encoding="UTF-8"?>
<syncCmd version="2.0">
  <elementList>
    <element>
      <channel>1</channel>
      <cmd>20</cmd>
    </element>
    ...
  </elementList >
</syncCmd>

```

```

<!-- Copy the face library to the front end -->
<!--Put /CGI/Smart/FaceLib/<Key>/SyncToIpc/Cmd-->
<!--<Key> Library id->
<!--request-->
<syncCmd>
  <elementList>
    <element>
      <channel><!-- Front channel --></channel>
      <cmd><!-- Command, start 20, stop 21, delete 22--></cmd>
    </element>
  </elementList>
</syncCmd>

```

2.7.51 Status of synchronizing face library to frontend

url: /CGI/Smart/FaceLib/<key>/SyncToIpc/State	
GET	
Description	Status of synchronizing face library to frontend
Query	None
Inbound Data	None
Success Return	<SyncState>
Explanations on protocol: Acquire status of synchronizing face library to frontend Explanations of parameters: <channel> Channel <state> Status; 0: Unsynchronized; 1: Synchronization in progress; 2: Synchronization succeeds; 3: Synchronization fails; 4: To be synchronized <progress> Progress percentage <successdCnt> Success count <failedCnt> Failure count	

SyncPara XML Block

```
<?xml version="2.0" encoding="UTF-8"?>
<syncState version="2.0">
  <elementList>
    <element>
      <channel><!--req,sx:integer--></channel>
      <state><!--dep,sx: integer--></state>
      <progress><!--dep,sx: integer--></progress>
      <successdCnt><!--dep,sx: integer--></successdCnt>
      <failedCnt><!--dep,sx: integer--></failedCnt>
    </element>
    ...
  </elementList>
</syncState>
```

Test cases

GET /CGI/Smart/FaceLib/xxxx/SyncToIpc/State

Request XML: none

Response XML: <SyncState>

```
<?xml version="2.0" encoding="UTF-8"?>
<syncState version="2.0">
  <elementList>
    <element>
      <channel>1</channel>
      <state>0</state>
      <progress>50</progress>
      <successdCnt>5</successdCnt>
      <failedCnt>0</failedCnt>
    </element>
    ...
  </elementList>
</syncState>
```

<!-- Copy to the front face database state -->

<!--Get /CGI/Smart/FaceLib/<Key>/SyncToIpc/State-->

<!--<Key> Library id -->

<!--response-->

<elementList>

 <element>

 <channel><!-- Front channel--></channel>

 <state><!--0 is not synchronized, 1 is in sync, 2 is successfully synchronized, 3 is failed to synchronize, 4 is to be synchronized --></state>

 <progress><!-- Progress percentage --></progress>

 <successdCnt><!-- Number of successful synchronizations --></successdCnt>

 <failedCnt><!-- The number of synchronization failures --></failedCnt>

 </element>

</elementList>

2.7.69 Remodel face library

url: /CGI/Smart/Face/Reset/Model	
PUT	
Description	Remodel face library
Query	None
Inbound Data	<ResetFaceModel>
Success Return	<ResponseStatus>
Explanations on protocol: Remodel face library, reserve protocol parameters; send 1 to do remodeling completely; send 0 for libKey and faceKey Explanations on key parameters: <libKey> Face library key, 0: Invalid <faceKey> Face library key; 0: Invalid <resetOpt> Remodeling type; 1: Remodel all; 2: Designated faceKey; 3. Designated libKey	

ResetFaceModel XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<resetFaceModel>
<libKey><!-- req, xs: long long --></libKey>
<faceKey><!-- req, xs: long long --></faceKey>
<resetOpt><!-- req, xs:integer --></resetOpt>
</resetFaceModel>
```

Test cases

PUT /CGI/Smart/Face/Reset/Model

Request XML: <ResetFaceModel>

Response XML: <ResponseStatus>

```
<?xml version="1.0" encoding="UTF-8"?>
<resetFaceModel>
<libKey>0</libKey>
<faceKey>0</faceKey>
<resetOpt>1</resetOpt>
</resetFaceModel>
```

<!-- Face Modeling -->

<!--Put /CGI/Smart/Face/Reset/Model-->

<!--request-->

<resetFaceModel>

<libKey><!-- Face library id --></libKey>

<faceKey><!-- Face map id--></faceKey>

<resetOpt><!--1 rebuild all, 2 specify faceKey, 3 specify libKey --></resetOpt>

</resetFaceModel>

2.3.47 Get and set the authentication enable status

url: /CGI/System/Identify/Enable/State	
GET	
Description	Get the authentication enable status
Query	None
Inbound Data	None
Success Return	<enableState>
PUT	
Description	Set the authentication enable status
Query	None
Inbound Data	<enableState>
Success Return	<ResponseStatus>
Explanations on protocol: Whether enabling authentication	
Explanations of parameters: <identifyEnable> true: Enable authentication; false: Disable authentication	

EnableState XML Block

```
<?xml version="2.0" encoding="UTF-8"?>
<enableState>
  <identifyEnable><!--req,xs:Boolean"true,false"--></identifyEnable>
</enableState>
```

Test cases

GET /CGI/System/Identify/Enable/State

Request XML: none

Response XML: <EnableState>

PUT /CGI/System/Identify/Enable/State

Request XML: <EnableState>

```
<enableState>
  <identifyEnable>true</identifyEnable>
</enableState>
```

Response XML: <ResponseStatus>

<!-- Face database enabled authentication status -->

<!--Get /CGI/System/Identify/Enable/State-->

<!--response-->

<enableState>

<identifyEnable><!-- true enables authentication, false disables authentication
--></identifyEnable>

</enableState>

2.7.55 Query face information

url: /CGI/Smart/FacePic/Query	
POST	
Description	Query face information
Query	None
Inbound Data	<FacePicQueryCondition>
Success Return	<FacePicQueryResult>
<p>Explanations on protocol: Query face data</p> <p>Explanations of parameters: Query xml: <pageNo> Page, pageNo >=0 <pageSize> Item per page, pageSize>=0 <libKey> key value of face library, LibKey>=0, No retrieval -1 <model> Modeling status; 0-Neglected; 1-Modeling succeeds; 2-Modeling fails; 3-No modeling <name> Name, 64 bits at most, support query of wildcard characters, such as Zhang *, Li Si *, Wang * Wu, * Liu, no retrieval "####" <sex> Gender; 0-Neglected; 1-Male; 2-Female; no retrieval -1 <birthDayStart> Start time of birthday, 2017-10-19, 16 bits at most <birthDayEnd> End time of birthday, 2017-10-19, 16 bits at most <nation> Nationality, as per national standard code, no retrieval-1 <province> Native place, province, as per code of national administrative division; 0 means unknown; no retrieval-1 <city> Native place, city, as per code of national administrative division; 0 means unknown; no retrieval-1 <certType> Certificate type; 0-Unknown; 1-ID card; 2-Certificate of officer; 3-Passport; 4-Staff No; no retrieval-1 <certNum> Certificate No., 64 bits at most, no retrieval "####" <country> Country, as per national and local ISO 3166-1 code table (see figure); no retrieval-1 <address> Address, 64 bits at most, no retrieval "####" <company> Company name, 64 bits at most, no retrieval "####" <access> Random information, for calibration use; 32 bits <password> Library password, encryption supports 16 bits at most Return statusCode:9 if password authentication fails statusString:"Securitycode Failed" subStatusCode:"CodeError"</p> <p>Reply xml: <totalCount> Total count <pageSize> Total count of current page <index> Page No. <libKey> key value of face library <faceKey> Face key value <facePath> http path <downloadfacePath> File download path (encrypted, avoid digest authentication to satisfy CGI) <model> Whether modeling; 0-Not establish model; 1-Establish model; send 1 during single import; send 0 during batch import <name> Name, 64 bits at most</p>	

<sex> Gender; 0-Unknown; 1-Male; 2-Female
<birthday> Birthday; format: 2018-07-24, 16 bits at most
<province> Native place, province, as per code of national administrative division; 0 means unknown
<city> Native place, city, as per code of national administrative division; 0 means unknown
<place> Native place, higher 16 digits means province, lower 16 digits means city; as per code of national administrative division; 0 means unknown
<certType> Certificate type; 0-Unknown; 1-ID card; 2-Certificate of officer; 3-Passport; 4-Staff No.
<certNum> Certificate No., 64 bits at most
<country> Country, as per national and local ISO 3166-1 code table (see figure)
<address> Address, 64 bits at most
<company> Company name, 64 bits at most

FacePicQueryCondition XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicQueryCondition version="1.0">
  <pageNo><!-- req,sx:integer--></pageNo>
  <pageSize><!-- req,sx:integer--></pageSize>
  <faceKey><!-- req,sx: integer--></faceKey>
  <model><!-- req,sx:integer--></model>
  <name><!-- req,sx:string--></name>
  <sex><!-- req,sx:integer--></sex>
  <timeSpanList>
    <timeSpan>
      <birthdayStart><!-- req, xs: datetime --></birthdayStart>
      <birthdayEnd><!-- req, xs: datetime --></birthdayEnd>
    </timeSpan>
  </timeSpanList>
  <nation><!-- req,sx:integer--></nation>
  <province><!-- req, xs: integer --></province>
  <city><!-- req, xs: integer --></city>
  <certType><!-- req,sx:integer--></certType>
  <certNum><!-- req,sx: string--></certNum>
  <country><!-- req, xs: integer --></country>
  <address><!-- req, xs: string --></address>
  <company><!-- req, xs: string --></company>
</facePicQueryCondition>
```

FacePicQueryResult XML Block

```
<facePicQueryResult version="2.0">
  <totalCount><!-- req,sx:integer--></totalCount>
  <pageSize><!-- req,sx:integer--></pageSize>
  <matchList>
    <matchElement>
      <index><!-- req,sx:integer--></index>
      <libKey><!-- req,sx: integer--></libKey>
      <faceKey><!-- req,sx: integer--></faceKey>
```

```
<facePath><!--req,sx: string--></facePath>
<downloadfacePath><!--req,sx: string--></downloadfacePath>
<model><!--req,sx:integer--></model>
<name><!--req,sx:integer--></name>
<sex><!--req,sx:integer--></sex>
<birthday><!-- req, xs: datetime --></birthday>
<nation><!--req,sx:integer--></nation>
<province><!-- req, xs: integer --></province>
<city><!-- req, xs: integer --></city>
<certType><!--req,sx:integer--></certType>
<certNum><!--dep,sx: string--></certNum>
</matchElement>
</matchList>
<country ><!-- req, xs: integer --></country>
<address ><!-- req, xs: string --></address>
<company ><!-- req, xs: string --></company>
</facePicQueryResult>
```

Test cases

POST /CGI/Smart/FacePic/Query

Request XML: <facePicQueryCondition> as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicQueryCondition version="1.0">
<pageNo>1</pageNo>
<pageSize>12</pageSize>
<libKey>123</libKey>
<faceKey>456</faceKey>
<model>1</model>
<name>zhangsan</name>
<sex>1</sex>
<timeSpanList>
<timeSpan>
<birthdayStart>1991-2-1</birthdayStart>
<birthdayEnd>2001-2-1</birthdayEnd>
</timeSpan>
</timeSpanList>
<nation>55</nation>
<province>66</province>
<city>77</city>
<certType>1</certType>
<certNum>341886199212171516</certNum>
<access>94AAABB419A9820DC171B43240CEEF41</access>
<password>/T6g05arqzu4=</password>
<country >0</country>
```

```
<address>huake</address>
<company>Tiandy</company>
</facePicQueryCondition>
```

Response XML: <facePicQueryResult>

```
<facePicQueryResult version="2.0">
<totalCount>120</totalCount>
<pageNo>1</pageNo>
<pageSize>12</pageSize>
<matchList>
<matchElement>
<index>1</index>
<libKey>123</libKey>
<faceKey>456</faceKey>
<facePath>XXXXXXXX</facePath>
<downloadfacePath>XXXXXXXX</downloadfacePath>
<model>1</model>
<name>zhangsan</name>
<sex>1</sex>
<birthday>1991-2-1</birthday>
<nation>55</nation>
<province>66</province>
<city>77</city>
<certType>1</certType>
<certNum>341886199212171516</certNum>
</matchElement>
...
</matchList>
<country>0</country>
<address>huake</address>
<company>Tiandy</company>
</facePicQueryResult>
```

<!-- Face Gallery Search -->

<!--Post /CGI/Smart/FacePic/Query-->

<!--request-->

<facePicQueryCondition>

<pageNo><!-- page number --></pageNo>

<pageSize><!-- Articles per page --></pageSize>

<libKey><!-- Face library id, LibKey = 0, do not retrieve -1--></libKey>

<model><!-- Modeling status, 0-ignore, 1-modeling success, 2-modeling failure, 3-not modeling --></model>

<name><!-- Name, up to 64 bytes, supports wildcard query, such as Zhang *, Lisi *, Wang * five, * six, does not retrieve "####"--></name>

```

    <sex><!-- Gender, 0-ignore, 1-male, 2-female, do not retrieve -1--></sex>
    <timeSpanList>
        <timeSpan>
            <birthdayStart><!-- Birth date start time, 2017-10-19, maximum 16 bytes
--></birthdayStart>
            <birthdayEnd><!-- End of birth date, 2017-10-19, maximum 16 bytes
--></birthdayEnd>
        </timeSpan>
    </timeSpanList>
    <nation><!-- Nationality, according to national standard code, do not retrieve
-1--></nation>
    <province><!-- Country of origin, province, assigned by national administrative division code,
0 means unknown, -1 is not retrieved--></province>
    <city><!-- Hometown, city, assigned by national administrative division code, 0 means
unknown, -1 is not retrieved--></city>
    <certType><!-- Document Type, 0-Unknown, 1-ID, 2-Military Officer ID, 3-Passport, 4
Employee Number, Do Not Retrieve -1--></certType>
    <certNum><!-- Document number, up to 64 bytes, "#####" is not retrieved--></certNum>
</facePicQueryCondition>
<!--response-->
<facePicQueryResult>
    <totalCount><!-- Total number --></totalCount>
    <pageSize><!-- Current page number --></pageSize>
    <matchList>
        <matchElement>
            <libKey><!-- Face library id--></libKey>
            <faceKey><!-- Face base map id--></faceKey>
            <index><!-- Page number --></index>
            <model><!-- Whether to model, 0-do not model, 1-model; send 1 for single sheet
import, 0 for batch import--></model>
            <sex><!-- Gender, 0-Unknown, 1-Male, 2-Female --></sex>
            <nation><!-- Ethnic --></nation>
            <province><!-- Country of origin, province, assigned by national administrative
division code, 0 means unknown--></province>
            <city><!-- Hometown, city, assigned by national administrative division code, 0
means unknown --></city>
            <certType><!-- Document Type, 0-Unknown, 1-ID, 2-Military Officer 3-Passport, 4
Employee Number --></certType>
            <certNum><!-- ID number, up to 64 bytes --></certNum>
            <name><!-- Name, up to 64 bytes --></name>
            <facePath><!-- http path --></facePath>
            <birthday><!-- Date of birth, format: 2018-07-24, maximum 16 bytes
--></birthday>
        </matchElement>

```

</matchList>
</facePicQueryResult>

2.7.56 Query by picture and analyze the face in group photo

url: /CGI/Smart/Import/AnalysisImage/SessionId/<ID>	
POST	
Description	Query by picture and analyze the face in group photo
Query	None
Inbound Data	File content
Success Return	<ResponseStatus>
Explanations on protocol: Query by picture and analyze the face in group photo Explanations of parameters: None	

Test cases

PUT /CGI/Smart/Import/AnalysisImage/SessionId/1

Request XML: none

Response XML: <ResponseStatus>

<!-- Face basemap add-upload search face pictures -->

<!--Post /CGI/Smart/Import/AnalysisImage/SessionId/<ID>-->

<!--<ID> Depends on the current time to obtain a random decimal number between 0x4000 ~ 0x4fff -->

<!-- Form upload image -->

<!-- Callback call /CGI/Smart/Query/AnalysisImage/SessionId/<ID> Get matting results -->

2.7.57 Query analyzed face

url: /CGI/Smart/Query/AnalysisImage/SessionId/<ID>	
POST	
Description	Query analyzed face
Query	None
Inbound Data	<FaceCondition>

Success Return	<FaceResult>
Explanations on protocol: Acquire analyzed face Explanations of parameters: URL <sessionId> client side and equipment interaction ID, refer to Schedule 1 Query xml <pageNo> Page, pageNo >=0 <pageSize> Item per page, pageSize>=0 Response xml <faceNum> Total file number (-2=Error handling; -1-Handling in progress; 0-Finished, no face; >0Finished, face found) <faceName> Picture name <faceSrc> Picture path	

FaceCondition XML Block

```
<faceCondition version="1.0">
<pageNo><!--req,sx: integer--></pageNo>
<pageSize><!--req,sx: integer--></pageSize>
</faceCondition>
```

FaceResult XML Block

```
<faceResult version="1.0">
<faceNum><!--req,sx: integer--></faceNum>
<faceInfoList>
<faceInfo>
<faceName><!--req,sx:string--></faceName>
<faceSrc><!--req,sx:string--></faceSrc>
</faceInfo>
...
</faceInfoList>
</faceResult>
```

POST /CGI/Smart/Query/AnalysisImage/SessionId/<ID>

Request XML: <FaceCondition>

```
<faceCondition version="1.0">
<pageNo>1</pageNo>
<pageSize>20</pageSize>
</faceCondition>
```

Response XML: <FaceResult>

```
<faceResult version="1.0">
<faceNum>20</faceNum>
<faceInfoList>
<faceInfo>
<faceName>name0.jpg</faceName>
```

```

<faceSrc>/tmp/face/face/name0.jpg</faceSrc>
</faceInfo>
...
</faceInfoList>
</faceResult>

```

```

<!-- Get human face map results -->
<!--Get /CGI/Smart/Query/AnalysisImage/SessionId/<ID>-->
<!--<ID> Depends on the current time to obtain a random decimal number between 0x4000 ~
0x4fff -->
<!--request-->
<!-- Success callback call /CGI/Smart/FacePic/<Key>/Manage Upload basemap information -->
<faceCondition>
    <pageNo><!-- page number --></pageNo>
    <pageSize><!-- Articles per page --></pageSize>
</faceCondition>
<!--response-->
<faceResult>
    <faceNum><!-- Total number of files --></faceNum>
    <faceInfoList>
        <faceInfo>
            <faceName><!-- Picture name --></faceName>
            <faceSrc><!-- Picture path --></faceSrc>
        </faceInfo>
    </faceInfoList>
</faceResult>

```

2.7.54 Add face picture, import picture firstly

url: /CGI/Smart/FacePic/<Key>/Manage	
POST	
Description	Add face picture, import picture firstly
Query	None
Inbound Data	<FacePicParas>
Success Return	<FacePicResult>
PUT	
Description	Set face parameters
Query	None

Inbound Data	<FacePicParas>
Success Return	<ResponseStatus>
DELETE	
Description	Delete face
Query	None
Inbound Data	None
Success Return	<ResponseStatus>
<p>Explanations on protocol: Acquire, set, add and delete face parameters.</p> <p>Explanations on key parameters: URL <facePicKey> key value of face, faceKey=0: Add face; faceKey>0: Modify face Request xml <libKey> key value of face library <checkCode> File check code; valid when faceKey=0, for calibrating face picture; no calibration needed when transmitted file is null <model> Whether modeling; 0-Not establish model; 1-Establish model; send 1 during single import; send 0 during batch import <name> Name, 64 bits at most <sex> Gender; 0-Unknown; 1-Male; 2-Female <birthday> Birthday; format: 2018-07-24, 16 bits at most <nation> Nationality, as per national standard code <province> Native place, province, as per code of national administrative division; 0 means unknown <city> Native place, city, as per code of national administrative division; 0 means unknown <certType> Certificate type; 0-Unknown; 1-ID card; 2-Certificate of officer; 3-Passport; 4-Staff No. <certNum> Certificate No., 64 bits at most <facePath> Path; upload upload.jpg or upload.png to add path, or transmit the file name to copy the path <country> Country, as per national and local ISO 3166-1 code table (see figure) <address> Address, 64 bits at most <company> Company name, 64 bits at most Response xml <key> key value of face library, responded when adding value..</p>	

FaceParas XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicParas>
<libKey><!-- req, xs: integer --></libKey>
<checkCode><!-- req, xs:string --></checkCode>
<model><!-- req, xs: integer --></model>
<name><!-- req, xs:string --></name>
<sex><!-- req, xs: integer --></sex>
<birthday><!-- req, xs: string--></birthday>
<nation><!-- req, xs: integer --></nation>
<province><!-- req, xs: integer --></province>
<city><!-- req, xs: integer --></city>
<certType><!-- req, xs: integer --></certType>
<certNum><!-- req, xs: string --></certNum>
<facePath><!-- req, xs: string --></facePath>
```



```
<country><!-- req, xs: integer --></country>
<address ><!-- req, xs: string --></address>
<company ><!-- req, xs: string --></company>
</facePicParas>
```

ResponseResult XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicResult>
<key><!-- req, xs: integer --></key>
</facePicResult>
```

Test cases

POST /CGI/Smart/FacePic/0/Manage

Request XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicParas>
<libKey>2</libKey>
<checkCode>120142</checkCode>
<model>1</model>
<name> Zhang San </name>
<sex>1</sex>
<birthday>2017-10-19</birthday>
<nation>66</nation>
<place>33</place>
<certType>1</certType>
<certNum>341886199212171516</certNum>
<facePath>P_01_50.jpg</facePath>
<country >0</country>
<address >huake</address>
<company >Tiandy</company>
</facePicParas>
```

Response XML: <FaceParas>

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicResult>
<key>456</key>
</facePicResult>
```

PUT /CGI/Smart/FacePic/1/Manage

Request XML: <FacePicParas>

Response XML: <ResponseStatus>

DELETE /CGI/Smart/FacePic/1/Manage

Request XML: none

Response XML: <ResponseStatus>

<!-- Add / edit / copy basemap information -->

<!--Post /CGI/Smart/FacePic/<Key>/Manage-->

```

<!--<Key> Library id, 0 means add face, greater than 0 means modify face -->
<!--request-->
<facePicParas>
    <libKey><!-- Library Id --></libKey>
    <checkCode><!-- File check code is valid when faceKey = 0, it is used to check the face
picture. --></checkCode>
    <model><!-- Whether to model, 0-do not model, 1-model; send 1 for single sheet import, 0
for batch import--></model>
    <name><!-- Name, up to 64 bytes --></name>
    <sex><!-- Gender, 0-Unknown, 1-Male, 2-Female --></sex>
    <birthday><!-- Date of birth, format: 2018-07-24, maximum 16 bytes --></birthday>
    <nation><!-- Nationality, according to national standard code --></nation>
    <province><!-- Country of origin, province, assigned by national administrative division code,
0 means unknown --></province>
    <city><!-- Hometown, city, assigned by national administrative division code, 0 means
unknown --></city>
    <certType><!-- Document Type, 0-Unknown, 1-ID Card, 2-Military Officer Card, 3-Passport, 4
Employee Number --></certType>
    <certNum><!-- ID number, up to 64 bytes --></certNum>
    <facePath><!-- Path, upload.jpg or upload.png when adding, file name when copying
--></facePath>
</facePicParas>

<!-- Delete basemap -->
<!--Delete /CGI/Smart/FacePic/<Key>/Manage-->
<!--<Key> Library id-->

```

2.7.53 Add face picture, import picture firstly

url: /CGI/Smart/Import/FacePic	
POST	
Description	Add face picture, import picture firstly
Query	None
Inbound Data	File content
Success Return	<ResponseStatus>
Explanations on protocol: Query by picture and analyze the face in group photo Explanations of parameters: None	

Test cases
PUT /CGI/Smart/Import/FacePic

Request XML: none
Response XML: <ResponseStatus>

<!-- Import pictures -->
<!--Post /CGI/Smart/Import/FacePic-->
<!-- Form upload image -->
<!-- transfer /CGI/Smart/FacePic/<Key>/ManageAdd basemap information-->

2.7.44 Acquire enabling status of face configuration channel

url: /CGI/Smart/FaceCnf/FaceCnfEnable/channels/<ID>/Event/<ID>	
GET	
Description	Acquire enabling status of face configuration channel
Query	None
Inbound Data	None
Success Return	<FaceCnfEnable>
PUT	
Description	Set enabling status of face configuration channel
Query	None
Inbound Data	<FaceCnfEnable>
Success Return	<ResponseStatus>
Explanations on protocol: This protocol is prepared for acquiring and setting the intelligent analysis on enabling status of face configuration channel. Explanations on key parameters: Event type; 1: Face detection; 2: Face recognition, sent to url Scene id, sent to url <ipcEnable>ipc enabling <nvrEnable>nvr enabling	

FaceCnfEnable XML Block

```
<?xml version="1.0" encoding="UTF-8"?>  
<FaceCnfEnable>  
<ipcEnable><!-- req, xs: boolean --></ipcEnable>  
<nvrEnable><!-- req, xs: boolean --></nvrEnable>  
</FaceCnfEnable>
```

Test cases

GET /CGI/Smart/FaceCnf/FaceCnfEnable/channels/1/Event/1/Scene/1

Request XML: none
Response XML: <FaceCnfEnable>
PUT /CGI/Smart/FaceCnf/FaceCnfEnable/channels/1/Event/1/Scene/1
Request XML: <FaceCnfEnable>
Response XML: <ResponseStatus>

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceCnfEnable>
<ipcEnable>true</ipcEnable>
<nvrEnable>true</nvrEnable>
</FaceCnfEnable>
  
```

```

<!-- Face detection / recognition enabled-->
<!--Put、Get /CGI/Smart/FaceCnf/FaceCnfEnable/channels/<ID>/Event/<ID>-->
<!--first<ID> Channel number, second <ID>1 face detection, 2 face recognition -->
<FaceCnfEnable>
    <ipcEnable><!-- ipc enabled --></ipcEnable>
    <nvrEnable><!--nvr enable--></nvrEnable>
</FaceCnfEnable>

<!-- Download by file for playback -->
<!--Post /ISAPI/Record/DownLoad/<filename>-->
<!--<filename> File name ending in .mp4 -->
<!-- Form download -->
  
```

2.7.61 Release cache data of affair

url: /CGI/Smart/SessionId/<ID>/Release	
PUT	
Description	Release cache data of affair
Query	None
Inbound Data	None
Success Return	<ResponseStatus>
Explanations on protocol: Release affair cache Explanations of parameters: <sessionId> Interaction ID of client and device, see attached Table 1	

Test cases

PUT /CGI/Smart/SessionId/0/Release
Request XML: none
Response XML: <ResponseStatus>

2.7.56 Query by picture and analyze the face in group photo

url: /CGI/Smart/Import/AnalysisImage/SessionId/<ID>	
PUT	
Description	Query by picture and analyze the face in group photo
Query	None
Inbound Data	File content
Success Return	<ResponseStatus>
Explanations on protocol: Query by picture and analyze the face in group photo Explanations of parameters: None	

Test cases

PUT /CGI/Smart/Import/AnalysisImage/SessionId/1

Request XML: none

Response XML: <ResponseStatus>

2.7.57 Query analyzed face

url: /CGI/Smart/FacePic/AnalysisImage/SessionId/<ID>	
POST	
Description	Query analyzed face
Query	None
Inbound Data	<FaceCondition>
Success Return	<FaceResult>
Explanations on protocol: Acquire analyzed face Explanations of parameters: URL <sessionId> client side and equipment interaction ID, refer to Schedule 1 Query xml <pageNo> Page, pageNo >=0 <pageSize> Item per page, pageSize>=0 Response xml <faceNum> Total file number (-2=Error handling; -1-Handling in progress; 0-Finished, no face; >0Finished, face found) <faceName> Picture name <faceSrc> Picture path	

FaceCondition XML Block

```
<faceCondition version="1.0">
<pageNo><!--req,sx: integer--></pageNo>
<pageSize><!--req,sx: integer--></pageSize>
</faceCondition>
```

FaceResult XML Block

```
<faceResult version="1.0">
<faceNum><!--req,sx: integer--></faceNum>
<faceInfoList>
<faceInfo>
<faceName><!--req,sx:string--></faceName>
<faceSrc><!--req,sx:string--></faceSrc>
</faceInfo>
...
</faceInfoList>
</faceResult>
```

POST /CGI/Smart/Query/AnalysisImage/SessionId/<ID>

Request XML: <FaceCondition>

```
<faceCondition version="1.0">
<pageNo>1</pageNo>
<pageSize>20</pageSize>
</faceCondition>
```

Response XML: <FaceResult>

```
<faceResult version="1.0">
<faceNum>20</faceNum>
<faceInfoList>
<faceInfo>
<faceName>name0.jpg</faceName>
<faceSrc>/tmp/face/face/name0.jpg</faceSrc>
</faceInfo>
...
</faceInfoList>
</faceResult>
```

2.7.59 Search face information by picture

url:
/CGI/Smart/Face/QueryByPic
POST

Description	Search face information by picture
Query	None
Inbound Data	<FacePicQueryCondition>
Success Return	<FacePicQueryResult>
<p>Explanations on protocol: Query face data by picture</p> <p>Explanations of parameters: Query xml: <sessionId> Interaction ID between client and device <libKey> key value of face library, LibKey>=0 <similarity> Similarity <faceName> Name of analyzed face picture <pageNo> Page, pageNo >=0 <pageSize> Item per page, pageSize>=0 <access> Random information, for calibration use; 32 bits <password> Library password, max. length of encryption is 16 bits Return statusCode:9 if password authentication fails statusString:"Securitycode Failed" subStatusCode:"CodeError"</p> <p>Reply xml: <sessionId> Interaction ID between client and device <totalCount> Total count <pageSize> Total count of current page <index> Page No. <libKey> key value of face library <faceKey> Face key value, faceKey=0 if it is put, means adding of face; faceKey>0 means face modification <model> Whether modeling; 0-Not establish model; 1-Establish model; send 1 during single import; send 0 during batch import <name> Name, 64 bits at most <sex> Gender; 0-Unknown; 1-Male; 2-Female <birthday> Birthday; format: 2018-07-24, 16 bits at most <nation> Nationality, as per national standard code <province> Native place, province, as per code of national administrative division; 0 means unknown <city> Native place, city, as per code of national administrative division; 0 means unknown <certType> Certificate type; 0-Unknown; 1-ID card; 2-Certificate of officer; <certNum> Certificate No., 64 bits at most <country> Country, as per national and local ISO 3166-1 code table (see figure) <address> Address, 64 bits at most <company> Company name, 64 bits at most <similarity> Similarity</p>	

FacePicQueryCondition XML Block

```

<?xml version="1.0" encoding="UTF-8"?>
<facePicQueryCondition version="1.0">
  <sessionId><!--req,sx:integer--></sessionId>
  <libKey><!--req,sx:integer--></ libKey>
  <similarity><!--req,sx:integer--></similarity>
  <faceName><!--req,sx:string--></faceName>
  <pageNo><!--req,sx:integer--></pageNo>
  <pageSize><!--req,sx:integer--></pageSize>

```

```
<access><!--req,sx:string--></access>
<password><!--req,sx:string--></password>

</facePicQueryCondition>
```

FacePicQueryResult XML Block

```
<facePicQueryResult version="2.0">
  <sessionId><!--req,sx:integer--></sessionId>
  <totalCount><!--req,sx:integer--></totalCount>
  <pageSize><!--req,sx:integer--></pageSize>
  <matchList>
    <matchElement>
      <index><!--req,sx:integer--></index>
      <libKey><!--req,sx: integer--></libKey>
      <faceKey><!--req,sx: integer--></faceKey>
      <fileType><!--req,sx:integer--></fileType>
      <model><!--req,sx:integer--></model>
      <name><!--req,sx:integer--></name>
      <sex><!--req,sx:integer--></sex>
      <timeSpanList>
        <timeSpan>
          <birthday><!-- req, xs: datetime --></birthday>
        </timeSpan>
      </timeSpanList>
      <nation><!--req,sx:integer--></nation>
      <province><!--req,sx:integer--></province>
      <city><!--req,sx:integer--></city>
      <certType><!--req,sx:integer--></certType>
      <certNum><!--dep,sx: string--></certNum>
    </matchElement>
  </matchList>
  <country><!-- req, xs: integer --></country>
  <address><!-- req, xs: string --></address>
  <company><!-- req, xs: string --></company>
</facePicQueryResult>
```

Test cases

POST /CGI/Smart/FacePic/QueryByPic/Result

Request XML: <facePicQueryCondition> As follows

```
<?xml version="1.0" encoding="UTF-8"?>
<facePicQueryCondition version="1.0">
  <sessionId>1</sessionId>
  <libKey>456</libKey>
  <similarity>50</similarity>
```



```
<faceName>1.jpg</faceName>
<pageNo>1</pageNo>
<pageSize>12</pageSize>
<access>94AAABB419A9820DC171B43240CEE41</access>
<password>/T6g05arqzu4=</password>
</facePicQueryCondition>
```

Response XML: <facePicQueryResult>

```
<facePicQueryResult version="2.0">
  <sessionId>1</sessionId>
  <totalCount>120</totalCount>
  <pageSize>12</pageSize>
  <matchList>
    <matchElement>
      <index>1</index>
      <libKey>123</libKey>
      <faceKey>456</faceKey>
      <fileType>0</fileType>
      <model>1</model>
      <name>zhangsan</name>
      <sex>1</sex>
      <birthday>1991-2-1</birthday>
      <nation>55</nation>
      <province>66</province>
      <city>77</city>
      <certType>1</certType>
      <certNum>341886199212171516</certNum>
    </matchElement>
    ...
  </matchList>
  <country>0</country>
  <address>huake</address>
  <company>Tiandy</company>
</facePicQueryResult>
```

2.7.66 Acquire NVR performance

/CGI/Smart/Property/Dev/<ID>	
GET	
Description	Acquire NVR performance
Query	None
Inbound Data	None
Success Return	<Property>

Explanations on protocol:

This protocol is prepared for acquiring the NVR performance. Dev: Device No., reserved, 0 as default; device type: 0 nvr 1 ipc.

Explanations on key parameters:

<total> Total device performance, 16 as default

<itemList>

<item>

<type> Type: 1. NVR face detection; 2. NVR face recognition; 3: IPC face detection; 4. IPC face recognition

<algtpe> Algorithm type: Behavior: Behavior analysis; face: Face recognition; Human: Human detection; Pept: Oilfield monitoring; PeopleNumAlarm: People number error alarm; Audio: Audio diagnosis; Video: Video diagnosis; Group: Group; OnDuty: On-duty detection; person amount statistics, PlatLicense: Place recognition; ParkGuard: Park guard; IllegalPark: Illegal park; IntelliTrace: Intelligent traction; structurization: Structuration; Helmet: Helmet; Prctduty: Single interrogation/unattended; Sleep: Sleep; NewFight: New fight; GetUp: Get up; HeightLimit: Height limit; NewDuty: New duty; Stranded: Stranded; Alone: Alone; Delivergoods: Deliver goods; FaceMosaic: Face mosaic; ColorTrack: Color traction; Loitering: Loitering; AttendedBaggage: Attended baggage; UnattendedBaggage: Unattended baggage

<everyProperty> Performance of every circuit

<cnt> Enabling amount

<maxCnt> Max. enabling amount

Property XML Block

```
<?xml version="1.0" encoding="UTF-8"?>
<Property>
<total><!-- req, xs: integer --></total>
<itemList>
<item>
<type><!-- req, xs: integer --></type>
<everyProperty><!-- req, xs: integer --></everyProperty>
<cnt><!-- req, xs: integer --></cnt>
<maxCnt><!-- req, xs: integer --></maxCnt>
</item>
</itemList>
</Property>
```

Test cases

GET /CGI/Smart/Property/Dev/0

Request XML: none

```
<?xml version="1.0" encoding="UTF-8"?>
<Property>
<total>16</total>
<itemList>
<item>
<type>1</type>
<everyProperty>1</everyProperty>
<cnt>1</cnt>
</item>
<item>
<type>2</type>
<everyProperty>1</everyProperty>
```

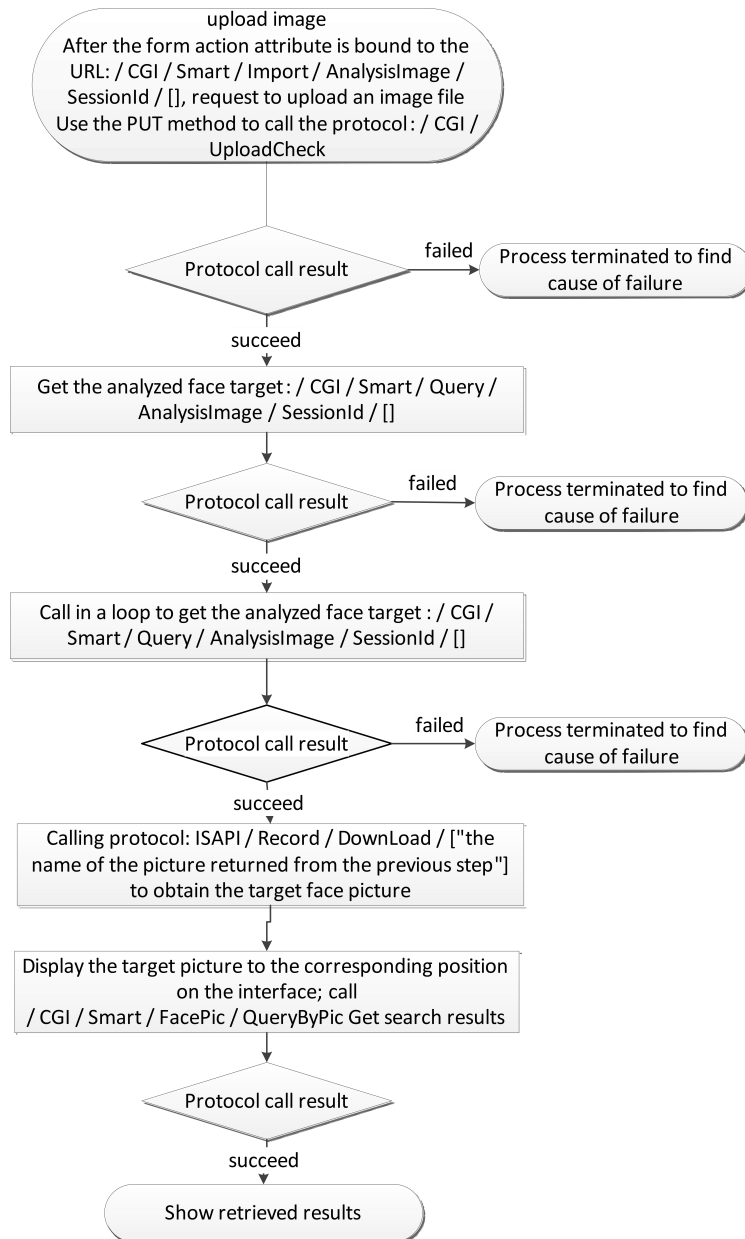
```
<cnt>1</cnt>
</item>
</itemList>
</Property>
```

2.10.1 Make request for file uploading

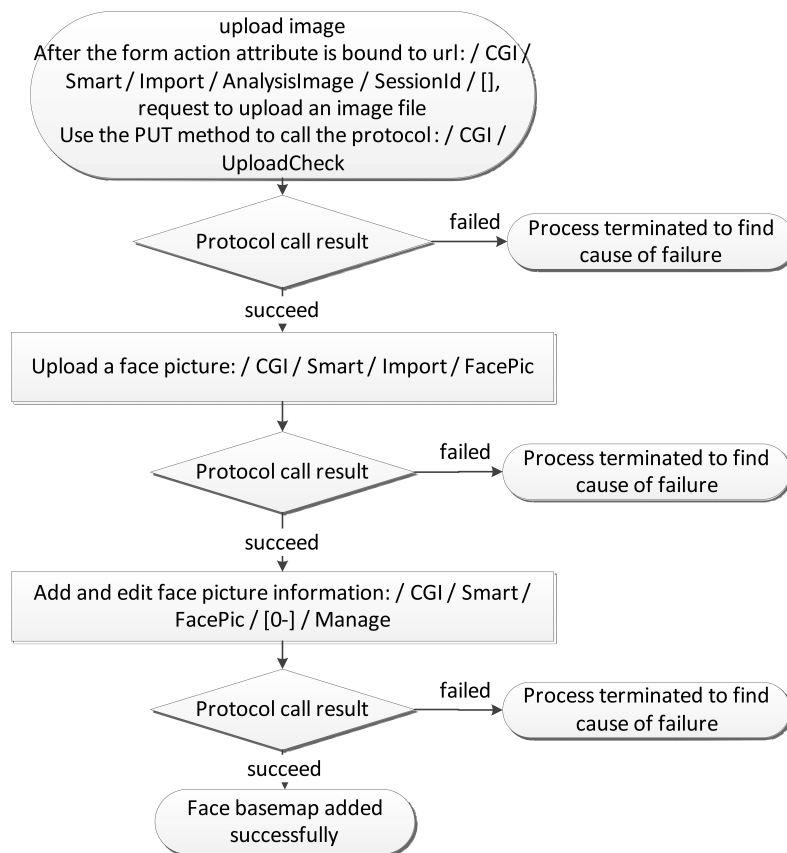
url: /CGI/UploadCheck	
GET	
Description	Make request for file uploading
Query	None
Inbound Data	None
Success Return	<ResponseStatus>
Explanations on protocol: This protocol is prepared for make request for file uploading, helping client or IE make request for file uploading via CGI protocol. Send this protocol before file uploading for calibration.	

2 Interface call flow chart

3.1 Find by image



3.2 Face basemap added



3 Response status code

4.1 Business response status code

Common HTTP status codes and field values in the response body:

Status Code	Description
200 OK	Successful operation
400 Bad Request	Operation exception, exception information in statusString field
401 Bad Request	Insufficient permissions

4.2 Response body

1. Successful:

return value: 200

statusCode:1

statusString:"OK"

subStatusCode:"ok"

Note: Return after successful setting.

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_OK, STATUS_CODE_OK_1);

2. failure:

return value: 400

statusCode:2

statusString:"Device Busy"

subStatusCode:"deviceBusy"

Description: Return after setting failure.

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_DEVBUSY, STATUS_CODE_DEVBUSY_4);

statusCode:3

statusString:"Device Error"

subStatusCode:"deviceError"

Description: Return after setting failure.

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_DEVERR, STATUS_CODE_DEVERR_1);

statusCode:4

statusString:"Invalid Operation"

subStatusCode:"notSupport"

Description: Return when using a non-existent instruction. For example, a protocol has only put, but the user calls get.

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_INVALID_OPERATE, STATUS_CODE_INVALID_OPERATE_1);

s

statusCode:4

statusString: "Invalid Operation"

subStatusCode: "invalidOperation"

Description: Returns after failing to obtain the ID in the URL. Such as getting the channel ID in the url

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_INVALID_OPERATE, STATUS_CODE_INVALID_OPERATE_6);

statusCode:5

statusString:"Invalid XML Format"

subStatusCode:" badXmlFormat"

Description: returned when the xml format is wrong.

CGI code: xml_responses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_INVALID_XMLFORMAT,

STATUS_CODE_INVALID_XMLFORMAT_1);

statusCode:6

statusString:"Invalid XML Content"

subStatusCode:"badXmlContent"

Description: return when the xml content is wrong.

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_INVALID_XMLCONTENT, STATUS_CODE_INVALID_XMLCONTENT_3);

statusCode:8

statusString:"IP Error"

subStatusCode:"GateWayConflict"

Description: When setting the IP or network segment, if it is occupied, the corresponding conflict error code will be returned to the WEB browser..

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_IP_CONFLICT, STATUS_CODE_IP_CONFLICT_2);

statusCode:9

statusString:"Securitycode Failed"

subStatusCode:"CodeError"

Description: When verifying the verification code, there are many cases of failure. For example, the verification code is wrong, the verification code is invalid, and the error reaches 5 times..

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_CHECK_SECURITY_CODE, STATUS_CODE_CHECK_SECURITY_CODE_1);

statusCode:10

statusString:"FuncExclu Failed"

subStatusCode:"WdrExcluError"

Description: When the mutual exclusion function is set at 50 \ 60 frames, there will be multiple cases of failure, for example: WDR is not allowed to be set, and the return fails..

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_FUNCEXCLU_ERR, STATUS_CODE_FUNCEXCLU_ERR_11);

// subStatusCode Of the type described below:

STATUS_CODE_FUNCEXCLU_ERR_1 "lcd" // Distortion correction

STATUS_CODE_FUNCEXCLU_ERR_2 "vcabaisc" // Behavior analysis

STATUS_CODE_FUNCEXCLU_ERR_3 "vcavfd" // Face Detection

STATUS_CODE_FUNCEXCLU_ERR_4 "vcacdd" // Crowd gathering

STATUS_CODE_FUNCEXCLU_ERR_5 "vcacpc" // People Counting

STATUS_CODE_FUNCEXCLU_ERR_6 "vcasvd" // On duty detection

STATUS_CODE_FUNCEXCLU_ERR_7 "vcavideo" // Video diagnosis

STATUS_CODE_FUNCEXCLU_ERR_8 "vcaalert" // Alert plan

STATUS_CODE_FUNCEXCLU_ERR_9 "vcalpr" // License Plate Recognition

STATUS_CODE_FUNCEXCLU_ERR_10 "vcaillpark" // Illegal parking

STATUS_CODE_FUNCEXCLU_ERR_11	"wdr"	// Wide dynamic
STATUS_CODE_FUNCEXCLU_ERR_12	"videocover"	// Video occlusion
STATUS_CODE_FUNCEXCLU_ERR_13	"corridor"	// Corridor mode
STATUS_CODE_FUNCEXCLU_ERR_14	"vcaspalert"	// Featured alert
STATUS_CODE_FUNCEXCLU_ERR_15	"vcalpd"	// Parking guard
STATUS_CODE_FUNCEXCLU_ERR_16	"vcafollow"	// track
STATUS_CODE_FUNCEXCLU_ERR_17	"vcaallfolloew"	// Panoramic sphere algorithm
STATUS_CODE_FUNCEXCLU_ERR_18	"vcahat"	// Hard hat algorithm
STATUS_CODE_FUNCEXCLU_ERR_19	"antishake"	// Electronic image stabilization
STATUS_CODE_FUNCEXCLU_ERR_20	"digitalzoom"	// Electronic zoom
.....		
STATUS_CODE_FUNCEXCLU_ERR_32	"allvca"	// All intelligent algorithms

statusCode:254

statusString: "Parameter Error"

subStatusCode: "parameterFailed"

Description: General error code, this error code can be used in case of wrong parameters.

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_PARAM_ERR, STATUS_CODE_PARAM_ERR_1);

statusCode:255

statusString: "Operation Failed"

subStatusCode: "operationFailed"

Description: For example, if you perform an email test, the test fails because you are not on the Internet or for other reasons.

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_COMMON_ERR, STATUS_CODE_COMMON_ERR_1);

return value: 403

statusCode:4

statusString:"Invalid Operation"

subStatusCode:"badAuthorization"

Description: Returned if the user does not have permission to set.

CGI code: xml_rsponses_status(_pcHttpReq, _pcXmlRsp, STATUS_CODE_INVALID_OPERATE, STATUS_CODE_INVALID_OPERATE_3);

return value: 401

<userCheck>

<statusValue>401</statusValue>

<statusString>lpFilter</statusString>

<passwdLeftValue>0</passwdLeftValue>

</userCheck>

Description: Check failed due to black and white list, prompting "username and password are

wrong".

```
<userCheck>  
<statusValue>401</statusValue>  
<statusString>Unauthorized</statusString>  
<passwdLeftValue>4</passwdLeftValue>  
</userCheck>
```

Description: Login failed due to incorrect user or password. For each error, the passwdLeftValue is decremented by 1 and the prompt "Username or password is wrong. Lock after 5 times of passwdLeftValue errors. Lock time is 5 minutes.".