CHAPTER 1

Overview

Access Control Web Services (ACWS) defines an application programming interface (API) for the Cisco Physical Access Manager (Cisco PAM).

This chapter includes general information, and instructions to enable the ACWS API on a Cisco PAM server. It also describes the ACWS authentication method, and the Namespaces and other information used to issue API requests.

Contents

Supported Functionality

The API for Cisco Physical Access Control Release 1.2.0 and higher supports the following features:

- Authentication APIs: applications must call the **authenticateUser** API to retrieve a security context object before calling any other API. The object is provided as a parameter in all subsequent calls for that API session. If the session ends, a new object must be retrieved. See Authentication and Authorization, page 1-14 for more information.
- Physical Security Integration Management (PSIM) APIs: for use by the Physical Security Operations Management applications. These APIs return information on access control devices, users, events and alarms. The API provides mechanisms to query events or alarms based on event type, time-interval, and source device criteria.
- Event Notification: notifies a client application when an event or alarm occurs. You can also query events or alarms based on the event type, time-interval, or source device.
- Door Command APIs: triggers actions based on access control events. For example, when a user attempts to access a door or device, the PSIM APIs can open or close the door.
- Badge Enrollment APIs: provisions badge credentials in the access control system. Also returns information on access levels and schedules.
- Recording External Events: allows external applications to log events and alarms in Cisco PAM.
- Fault Codes: API errors return major and minor fault codes. See Chapter 3, “Fault Codes” for descriptions.

Understanding Cisco PAM Web Services

The Cisco PAM web service (PSIMWsService) can provide information regarding doors, locks, badges, personnel, and alarms. APIs can also execute door commands, act on an alarm, or update badges.

For example:

- An application can receive the events generated when user access is granted or denied.
- The application can open or close a door.
- An application can create visitors and assign access policies to allow access to specific doors or locations.
- An application can provision badge credentials in the access control system.

This section describes the following:

Interfaces

The Cisco PAM web service (PSIMWsService) exposes two interfaces: *PSIMWsPortType* and *AcecssPolicyPortType* (Figure 1-1).

Tip The interface is also known as *portType* in web service terminology. The interfaces are exposed using WSDL 1.1 specifications.

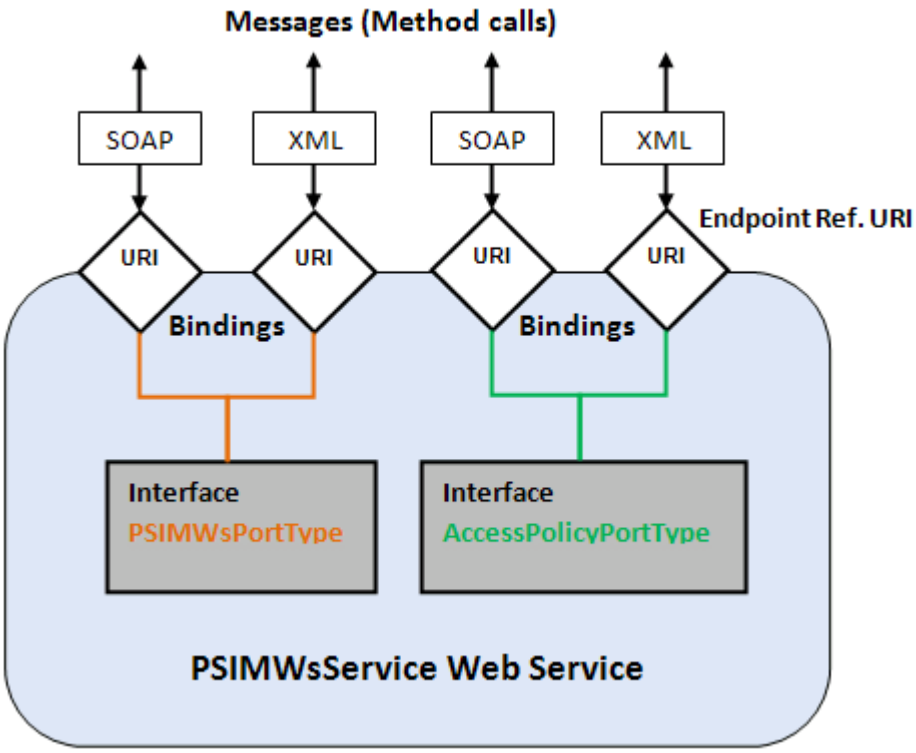
The *PSIMWsPortType* interface provides the following methods:

- Authenticate users (login and logout)
- Get details of access control device, including doors and locks
- Get personnel and organization details
- Act on an alarm
- Execute a door command
- Register a listener to receive Cisco PAM events

The *AccessPolicyPortType* interface provides the following methods:

- Get details of access levels and schedules
- Get details of badges
- Update badges

Figure 1-1 CPAM Web Service Interfaces



Note The PSIMWsService web service does not provide methods to configure the Cisco PAM server. Use the Cisco PAM client to configure the Cisco PAM server.

Bindings and Endpoint Address

Since both interfaces support SOAP/HTTP or XML/HTTP message formats (Figure 1-1), there are a total of four unique endpoint addresses (or URIs). Each endpoint address uniquely identifies the interface and the message format to use.

The client application uses one of the endpoint addresses shown in Table 1-1 to access the web-service method based on the message format (SOAP or XML).

Table 1-1 Interface Message Formats and Endpoint Addresses

Interface	Message Format	Endpoint Address used by client application
PSIMWsPortType	SOAP	http://<*cpam-server*>/acws/services/psimws
PSIMWsPortType	XML	http://<*cpam-server*>/acws/services/psimxml
AcecssPolicyPortType	SOAP	http://<*cpam-server*>/acws/services/acpolicy
AcecssPolicyPortType	XML	http://<*cpam-server*>/acws/services/acpolicyxml

Note The implementation for many individual methods using XML/HTTP format is not complete. Please refer to the actual WSDL or send an e-mail to acws-interest@cisco.com for more inforamtion.

Viewing the Web Service Information on a Cisco PAM server

Enter the following URL to display the interfaces, methods and endpoint addresses available on a Cisco PAM server.

http://<*cpam-server*>/acws/services

Note The *ACVSMPortTYpe* interface methods are not supported. These methods provide integration between Cisco PAM and Cisco Video Surveillance Manager (Cisco VSM) and should not be used by third party applications.

Enter the following URLs to display the actual WSDL for each supported interface.

http://<cpam-server>/acws/services/psimws?wsdl

http://<cpam-server>/acws/services/acpolicy?wsdl

Security

The Cisco PAM Web Service provides the following security measures:

- HTTP or HTTPS based access.
- Application level authentication. Before calling any method, the client application must call ***authenticateUser***. This method returns a security context object used as a required parameter in subsequent API calls.

Error Handling

Method errors or exceptions return an *AcWsFault* that include a description and identify if the event as major or minor fault code.

See Chapter 3, "Fault Codes" for descriptions of the major fault codes.

Request and Response Samples

Request and response examples are provided in Chapter 2, "API Functions". To capture additional examples, use a network packet capture tool such as the TCPMon utility.

Auto-Generating Java or C# Code from the WSDL File

This section describes tools used to auto-convert a WSDL document to fully annotated Java or C# code that can be used to call the Cisco PAM web service methods.

- Client Application Code—Java Example, page 1-5
- Client Application Code—C# Example, page 1-7

Client Application Code—Java Example

Use the *wsdl2java* converter tool to convert a WSDL document to fully annotated Java code that can be used to access the Cisco PAM Web Service methods.

Step 1 Enter the *wsdl2java* command to convert a WSDL document to fully annotated Java code.

The syntax is:

wsdl2java -d <*destination-folder*> <*wsdl-url*>

For example, the following command generates the Java code in the `/generated-source` folder:

wsdl2java -d /generated-source http://*my-cpam-server***/acws/services/psimws?wsdl**

Tip See https://cwiki.apache.org/CXF20DOC/wsdl-to-java.html for more information on *wsdl2java* converter tool.

Step 2 Compile the code and use it in your client application to integrate with CPAM WS.

Step 3 Call the Web Service methods.

For example:

```
// The PSIMWsService Web Service
private final static QName SERVICE = new QName("http://cisco.com/physec/acws/",
"PSIMWsService");
// WSDL location URL
private final static URL WSDL_LOCATION = new
URL("http://cpam-server/acws/services/psimws?wsld");

PSIMWsService _webService = null;
PSIMWsPortType m_ PSIMWsInterface = null;
AccessPolicyPortType m_AccessPolicyInterface
SecurityContext _ctx = null;

…

try
{
    //Reference to the PSIMWsService web service
    _webService = new PSIMWsService(WSDL_LOCATION, SERVICE);

    //Get handle to PSIMWsPortType interface
    _ PSIMWsInterface = _webService.getPSIMWsSoapPort();

    //Get handle to AccessPolicyPortType interface
    _ AccessPolicyInterface = _webService.getAccessPolicySoapPort();
}
 catch (WebServiceException e)
{
    //Handle exception
}
catch (Exception e)
{
    //Handle exception
}

….

// Authenticate user
UserCredentialType uct = new UserCredentialType();
uct.setUsername("username");
uct.setPassword("password");
_ctx = _ PSIMWsInterface.authenticateUser(uct);

…

//Call methods using interface handles
AcDeviceTypeList deviceTypeList= _ PSIMWsInterface.getAcDeviceTypes(_ctx);
…
List<Schedule> schedules = _AccessPolicyInterface.getAllSchedules(_ctx);
…

//In the end logout user and close the session
_PSIMWsInterface.logoutUser(_ctx);
```

Client Application Code—C# Example

Use the *wsdl.exe* converter tool to convert a WSDL document to fully annotated C# code that can be used to access the Cisco PAM Web Service methods.

Step 1 Enter the *wsdl.exe* command to convert a WSDL document to fully annotated C# code.

The syntax is:

wsdl /l:CS /protocol:SOAP <*wsld-file*> <*xsd files …*>

The following example generates the PSIMWsService.cs C# code:

wsdl /l:CS /protocol:SOAP acws.wsdl physic-common.xsd ws-addr.xsd ac-common.xsd ac-schedule.xsd ac-policy.xsd ac-event.xsd

Tip

- If you install MS Dev Studio 2008 and .NET Framework SDK 3.5 then you already have the WSDL converter tool.
- See http://msdn.microsoft.com/en-us/library/7h3ystb6(VS.71).aspx for more information on the *wsdl.exe* tool.

Step 2 Create a Client Proxy DLL using the C# command line compiler.

The following example generates the *PSIMWsService.dll*. Add reference to this DLL in the MS Dev Studio for your C# application.

csc /t:library /r:System.Web.Services.dll /r:System.Xml.dll PSIMWsService.cs

Tip See http://msdn.microsoft.com/en-us/library/78f4aasd(VS.80).aspx for more information on the C# command line compiler.

Step 3 Call the Web Service methods.

For example:

```
// The PSIMWsService Web Service
private String _PSIMServiceUrl = "http://cpam-server/acws/service/psimws";
private String _AccessPolicyServiceUrl = "http://cpam-server/acws/service/acpolicy";
private PSIMWsBinding _psimBinding;
private AccessPolicyBinding _apBinding;
private SecurityContext _ctx;

…

// PSIMWsPortType interface with SOAP/HTTP
_ psimBinding = new PSIMWsBinding();
_ psimBinding.Url = _PSIMServiceUrl;

// AccessPolicyPortType interface with SOAP/HTTP
_apBinding = new AccessPolicyBinding();
_apBinding.URL = _AccessPolicyServiceUrl;

….

// Authenticate user
UserCredentialType uct = new UserCredentialType();
uct.username = "username";
```

```
uct.password = "password";
_ctx = _ psimBinding.authenticateUser(uct);

…

//Call methods using interface handles
getAcDeviceTypesReqType reqType = new getAcDeviceTypesReqType();
reqType.secCtx = _ctx;
getAcDeviceTypesResponse resp = _ psimBinding.getAcDeviceTypes(reqType);
AcDeviceType[] dtypes = resp.deviceTypes;

…

getAllAccessLevels gaal = new getAllAccessLeves();
gaal.ctx = _ctx;
accessLevel[] als = _apBinding.getAllAccessLevels(gaal);
…

//In the end logout user and close the session
logoutUser lu = new logoutUser();
lu.setCtx = _ctx;
_ psimBinding.logoutUser(lu);
```

Namespaces

Web Services APIs are defined using WSDL and various object types defined by the XML schema. The schema definitions uses the following target namespaces.

- Interfaces, methods and types are defined using the target namespace: http://cisco.com/physec/acws
- BaseDevice is defined using the target namespace: http://cisco.com/physec.

Note BaseDevice is a type that defines the base device class. AcDevice and CameraDevices are sub-classes of the BaseDevice in their respective namespaces.

- Camera devices are defined using the target namespace: http://cisco.com/physec/video interfaces.
- ACWS devices and video names are derived by extension.

Enabling Web Services on the Cisco PAM Server

To enable the Web Services API functionality on the Cisco PAM server, you must purchase and install the optional Web Services license, and enable the API service, as described in the following sections:

Enabling the API Service on the Cisco PAM Server

In Cisco PAM Release 1.2.0, you must manually enable the Web Service API, as described in the following procedure.

Beginning with Cisco PAM Release 1.3.0, the Web Service API is enabled by default. Use the following instructions to verify the Status is *Enabled*.

For all releases, you must install the API license, as described in the "Purchasing and Installing the Cisco PAM API License" section on page 1-10.

Procedure

Complete the following procedure to verify that the Web Services API is enabled, or to enable the service if it is stopped.

Step 1 Log on to the Cisco PAM appliance as described in the *Cisco Physical Access Manager User Guide*.

Step 2 Select the **Monitoring** tab and then select **Status**, as shown in Figure 1-2.

The Status window appears by default. This window also appears when you first log on.

Figure 1-2 Services tab in the Cisco PAM Server Administration Utility

Step 3 Verify that the status is *Enabled*.

Step 4 If the status is *Disabled,* click the **Enable** button.

A confirmation message appears and the Status changes to **Enabled**.

Step 5 Continue to the "Purchasing and Installing the Cisco PAM API License" section on page 1-10.

Note Beginning in Release 1.3.0, the message *Web Services license not applied* appears if the API license is not installed.

Purchasing and Installing the Cisco PAM API License

To enable the API functionality, you must purchase the optional API license from the Cisco website and install it on the Cisco PAM server. If the API license is not installed, API requests to the Cisco PAM server return an error.

This section includes the following information:

Tip For more information on server configuration and optional licenses, see the *Cisco Physical Access Manager User Guide*.

Purchasing the API License

To purchase the Cisco PAM API license, do the following:

Step 1 Determine the Cisco PAM appliance serial number (the serial number is required to complete the purchase). See Displaying the Cisco PAM Appliance Serial Number, page 1-13 for more information.

Step 2 Purchase the licence by contacting your Cisco sales representative or any Cisco reseller. For more information, visit http://www.cisco.com/en/US/ordering/index.shtml.

Note The part number for the Web services API optional license is **CIAC-PAME-WSAPI=**.

Step 3 When the purchase is complete, you are issued a Product Authorization Key (PAK) in paper form, or in an email message.

Step 4 Continue to Installing the API License, page 1-11 for information on the two options used to download and install the license file using the PAK number.

Installing the API License

If your PC is connected to the Internet, you can enter the Product Authorization Key (PAK) to download and install a license file. You can also install a license file stored on a local disk.

This section includes the following information:

Option 1: Enter the Product Authorization Key to Download the License File

Note To use this method, your PC must be connected to the Internet.

Step 1 Locate the Product Authorization Key (PAK) created with the purchase of the optional feature.

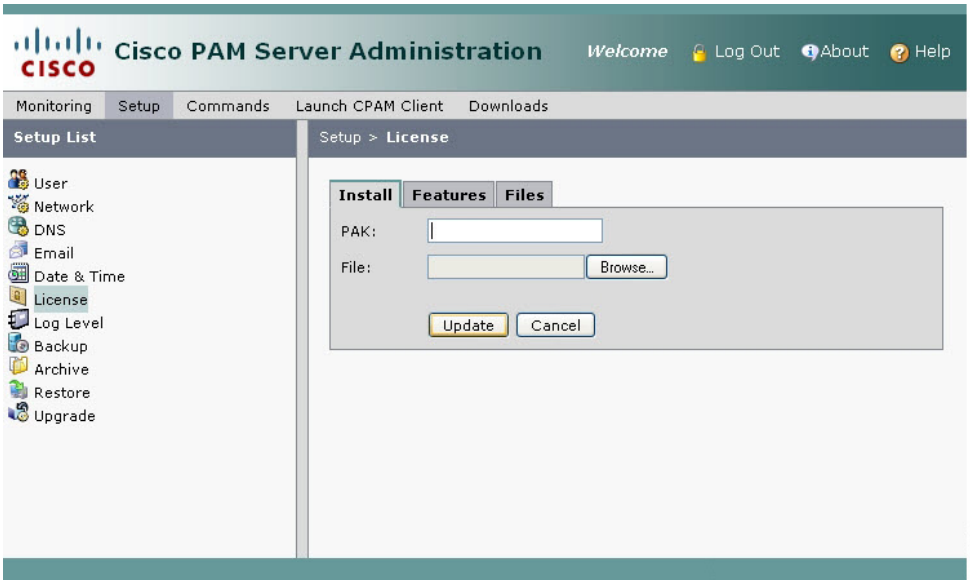
Step 2 Log on to the Cisco PAM appliance. See the *Cisco Physical Access Manager User Guide* for more information.

Step 3 Click the **Setup** tab, and then select the **License** menu, as shown in Figure 1-3.

Step 4 Enter the **PAK** code.

Step 5 Select **Update** to download and install the license file on the appliance and activate the features.

Figure 1-3 Installing Optional Feature Licenses



Note If the license file does not download, verify that your PC has Internet access, or use the method described in Option 2: Obtain the License File from the Cisco Web Site, page 1-12.

Step 6 Select the **Features** tab to verify that the new license was added. See Verifying the Installed Licenses, page 1-12 for more information.

Option 2: Obtain the License File from the Cisco Web Site

To use this method, obtain the license file from the Cisco Web site using a PC connected to the Internet, and transfer the file to the workstation used for server configuration.

Step 1 Locate the Product Authorization Key (PAK) created with the purchase of the optional feature.

Step 2 In a Web browser, open the Cisco Product License Registration Web page.

http://www.cisco.com/go/license/

Step 3 Follow the on-screen instructions to complete the form and enter the Product Authorization Key (PAK). When you are done, a license file with the extension `.lic` is sent to your email address.

Step 4 Transfer the file to the drive of the PC used for the configuration.

Step 5 In the License screen (Figure 1-3 on page 1-11), click **Browse** to select the license file located on your local drive. When selected, the file name appears in the File field.

Step 6 Select **Update** to install the license file on the Cisco PAM appliance and activate the features.

Step 7 Select the **Features** tab to verify that the new license was added. See Verifying the Installed Licenses, page 1-12 for more information.

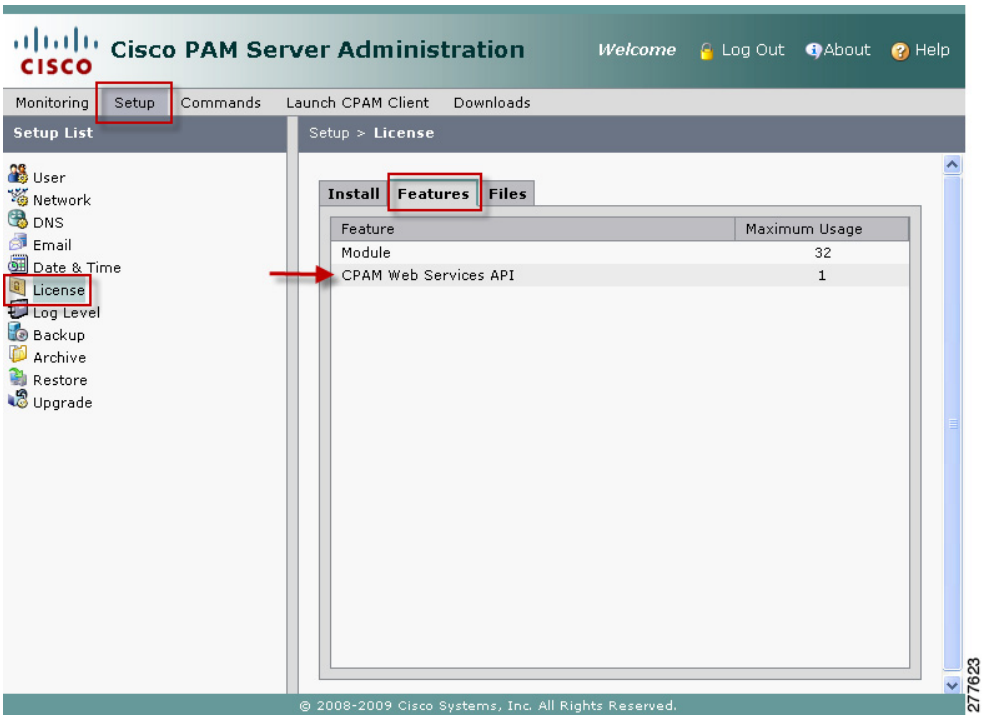
Verifying the Installed Licenses

From the Cisco PAM Server Administration utility, do the following:

Step 1 Select the **Setup** tab and then select the **License** menu, as shown in Figure 1-4.

Step 2 Select the **Features** tab to view the installed licenses.

Figure 1-4 License Features List



Displaying the Cisco PAM Appliance Serial Number

To view the appliance serial number, do the following:

Step 1 Log on to the Cisco PAM Server Administration utility.

See the *Cisco Physical Access Manager User Guide*, or ask your system administrator for assistance.

Step 2 Select the **Monitoring** tab, and then select **Server Status**, as shown in Figure 1-5.

Step 3 Refer to the entry for *Server Serial Number*.

Figure 1-5 Cisco PAM Appliance Serial Number

Authentication and Authorization

Before a method is called, use the **authenticateUser** API to send the Cisco PAM username and password, and retrieve a security context object (**secCtx**). Each subsequent API call uses this **secCtx** object as a parameter to authorize the API action.

Note For the current Cisco PAM release, we recommend using the Administrator username and password for API authentication.

Ending an API session

API sessions end after a default idle time of 10 hours, or you can manually end the session using the **logoutUser** API.

- If a application session remains idle for a default duration of 10 hours, the session automatically ends and the security context object is deactivated.
- Applications can also use the **logoutUser** API to end an API session and deactivate the security context object.
- API calls using an expired security context object return a fault. See Chapter 3, "Fault Codes" for more information.
- Cisco PAM Web Service performs an automatic check every 10 minutes for idle sessions to expire.

Note To begin a new API session, you must retrieve a new security context object (using **authenticateUser**).

API Username and Password

For the current Cisco PAM release, we recommend using the Administrator username and password for API authentication.

Usernames and passwords can also be configured using the Cisco PAM application to limit the API functionality:

- Use the Logins module to create the username and password for Cisco PAM client access.
- Use the Profiles module to define the Cisco PAM modules and commands available to a user.

For example, if an API application or user needs to view devices and events, the Logins username must be assigned a Profile with privileges to view events and devices. If an API user or application will invoke door commands, the username must include a profile with those privileges.

Tip See the *Cisco Physical Access Manager User Guide* for instructions to configure Cisco PAM logins and profiles.

API Security

The Cisco PAM server and API use SSL for secure communication between the server and clients. The server uses a X.509 certificate (also called an *SSL certificate*) to verify its identity when a client attempts to connect to the server.

Be default, the Cisco PAM server provides a self-signed certificate, which a client typically rejects. To prevent a client from rejecting this certificate, take one of the of the actions that Table 1-2 describes.

Table 1-2 Methods for Preventing a Client from Rejecting the Cisco PAM Server Self-Signed Certificate

Method	Notes
If you are using a Java client, configure the SSL libraries for your clients to trust the self-signed certificate by using the Java keytool to import the certificate into the client truststore.	Procedure: 1. SSH to the Cisco PAM server using the user `cpamadmin`. 2. Enter `sudo su` to get a privileged shell. 3. `cpamservercert.jks` has a Cisco PAM server certificate in the Java keystore. Use SFTP to copy this file on a client machine and import this certificate to the client trust store.
If you are using a client other than Java, configure the SSL libraries for your clients to trust the self-signed certificate.	You may be able to copy the self-signed certificate to a special directory on the client system. See your library documentation for detailed information.
Modify the way in which your client code validates certificate trust chains.	Some languages provide configurable SSL files that let you change the default certificate validation behavior.

A client verifies its identity with a user name and password that are sent to the server by the client application.

Understanding Unique IDs

Many parameters include a user-defined ID number, and a machine generated unique ID.

- Readable IDs identify a specific record for an object, such as **personId** or **badgeId**. For example: a **personId** might be 3215. In some cases, this readable ID is used in the API request.
- A *unique ID* (**unid**) for an object is used in most API requests, and is displayed in the API result. The *unique ID* for an object (such as a person or badge) is generated by the database and is the unique identifier for any record. For example, the *unid* for a person object might be `Z4JT5umCTzyCmVfvI6RAKw==`.

Review the description for each API to determine which ID type is required.

API Logging

For debugging, API request and response messages are logged in the `catalina` log file located on the Cisco PAM server in the folder:

`/opt/cisco/cpam/apache-tomcat/logs`

By default, Web Service debug logs are written in `webapp.log` which is located in the folder:

`/opt/cisco/cpam/logs`