



# Cisco Unity Connection Provisioning Interface (CUPI) API for System Settings

---

- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- General Configuration, on page 1](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Cluster, on page 5](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Configuration Values, on page 10](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Authentication Rules, on page 15](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Roles , on page 20](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Restriction Patterns, on page 27](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Restriction Tables, on page 32](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- License Dump APIs, on page 38](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Smart Licensing , on page 43](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Custom Subject Line, on page 53](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Language Map , on page 58](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- LDAP Phone Number Transform , on page 60](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Authorization Server, on page 62](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- System Default Language and TTS Language, on page 65](#)
- [Cisco Unity Connection Provisioning Interface \(CUPI\) API -- Wave Formats , on page 66](#)

## Cisco Unity Connection Provisioning Interface (CUPI) API -- General Configuration

### Default Unity Connection Sender Name

This section contains information on how to use the API to change the display name of Unity Connection Default User. Display name of default user is used as the sender name for System Generated Notifications from Connection and Outside Caller Messages. This API updates the "Unity Connection Default Sender Name" field under General Configuration.

## Viewing the sender name for System Generated Notifications

The following is an example of the GET request that lists the details of sender name for System Generated Notifications:

```
GET https://<connection-server>/vmrest/setDefaultSenderName
```

The following is response for the above GET request:

```
<User>
  <ObjectId>b296be34-0f21-449a-acce-23b8eadf1af9</ObjectId>
  <DisplayName>Cisco Unity Connection Messaging System</DisplayName>
</User>
```

```
Response Code: 200
```

### JSON Example

The following is an example to GET the display name and object id of "Unity Connection" users.

```
Request URI:
GET https://<connection-server>/vmrest/setDefaultSenderName
Accept : application/json
Connection : keep-alive
```

The following is response from the above GET request and the actual response will depend upon the information given by you:

```
{
  "ObjectId": "b296be34-0f21-449a-acce-23b8eadf1af9",
  "DisplayName": "NewUser"
}
```

```
Response Code: 200
```

## Updating the Sender Name for System Generated Notification

The following is an example of the PUT request that allows you to update the system generated notification:

**Example :** Update the system generated notification

```
PUT https://<connection-server>/vmrest/setDefaultSenderName
```

The following is response for the above PUT request.

```
Request Body:
<User>
<DisplayName>NewUser</DisplayName>
</User>
```

```
Response Code: 204
```

### JSON Example

Following is an example to update the display name.

```
Request URI:
  PUT https://<connection-server>/vmrest/setDefaultSenderName
```

```
Accept : application/json
Content-Type : application/json
Connection : keep-alive
{
  "DisplayName": "NewUser"
}
```

The following is response for the above PUT request:

```
Request URI: 204
```

## Ciphers

This section contains information on how to use the API to change the value of ciphers supported with Unity Connection. Using this API, you can change the value of following ciphers:

- TLS ciphers over SIP interface
- SRTP ciphers over RTP interface
- HTTPS Ciphers over HTTPS interface

This API updates the "TLS Ciphers", "SRTP Ciphers", and "HTTPS Ciphers" field under General Configuration.

## Viewing the Selected Option of TLS, SRTP, and HTTPS Cipher values

The following is an example of GET request to get the details of TLS, SRTP, and HTTPS ciphers.

```
GET https://<connection-server>/vmrest/generalconfigurations/
```

The following is response for the above GET request:

```
<GeneralConfigurations total="1">
  <GeneralConfiguration>
    <URI>/vmrest/generalconfigurations/b572544f-49ce-4ec4-874e-41e414597c14</URI>
    <ObjectId>b572544f-49ce-4ec4-874e-41e414597c14</ObjectId>
    <TlsCiphers>0</TlsCiphers>
    <SrtpCiphers>0</SrtpCiphers>
    <HttpsCiphers>0</HttpsCiphers>
  </GeneralConfiguration>
</GeneralConfigurations>
```

```
Response Code: 200
```

### JSON Example

The following is an example to GET the values for TLS and SRTP Ciphers.

```
Request URI:
  GET https://<connection-server>/vmrest/generalconfigurations/
Accept : application/json
Connection : keep-alive
```

```
{
  {
    "GeneralConfigurations": {
      "-total": "1",
      "GeneralConfiguration": {
        "URI": "/vmrest/generalconfigurations/b572544f-49ce-4ec4-874e-41e414597c14",
        "ObjectId": "b572544f-49ce-4ec4-874e-41e414597c14",
        "TlsCiphers": "0",
        "SrtpCiphers": "0"
        "HttpsCiphers": "0"
      }
    }
  }
}
```

Response Code: 200

## Updating the Values For TLS, SRTP, HTTPS Ciphers and TLS Version

The following is an example of the PUT request that allows you to update the value of TLS, SRTP, HTTPS Ciphers and TLS Version.

**Example :** Update the TLS, SRTP, HTTPS Ciphers and TLS Version.

```
PUT https://<connection-server>/vmrest/generalconfigurations/<objectid>
```

The following is the response of above PUT request.

```
<GeneralConfiguration>
  <TlsCiphers>1</TlsCiphers>
  <SrtpCiphers>1</SrtpCiphers>
  <HttpsCiphers>1</HttpsCiphers>
</GeneralConfiguration>
```

Response Code: 204



### Note

- Make sure to restart the Connection Conversation Manager service in Cisco Unity Connection Serviceability for TLS, SRTP ciphers and TLS Version changes to take effect.
- Make sure to restart the tomcat service for HTTPS cipher changes to take effect.
- Make sure to restart the Tomcat service using CLI 'utils service restart Cisco Tomcat' for TLS Version changes to take effect.

### JSON Example

The following is an example to update the TLS and SRTP Ciphers values.

```
Request URI:
PUT https://<connection-server>/vmrest/generalconfigurations/<objectid>
```

```

Accept : application/json
Content-Type : application/json
Connection : keep-alive
{
  "GeneralConfiguration": {
    "TlsCiphers": "1",
    "SrtspCiphers": "1"
    "HttpsCiphers": "1"
  }
}

```

The following is the response of the above PUT request :

```
Response Code: 204
```

## Corresponding Enum values for TLS, SRTP and HTTPS Ciphers

The following table describes the associated Enum values of the TLS and SRTP ciphers with Unity Connection 11.5(1).

Enum Value	Drop-Down Menu option for TLS Cipher	Drop-Down Menu option for SRTP Cipher	Drop-Down Menu option for HTTPS Cipher
0	Strongest- AES-256 SHA-384 Only: RSA Preferred	All supported AES-256, AES-128 ciphers	All Supported EC and RSA ciphers
1	Strongest- AES-256 SHA-384 Only: ECDSA Preferred	AEAD AES256 GCM based ciphers only	RSA ciphers Only
2	Medium- AES-256 AES-128 Only: RSA Preferred	AEAD AES-256, AES-128 GCM-based ciphers:	-
3	Medium- AES-256 AES-128 Only: ECDSA Preferred	-	-
4	All Ciphers RSA Preferred	-	-
5	All Ciphers ECDSA Preferred	-	-

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Cluster

## About Clusters

Administrator can use this API to fetch the details of a Unity Connection cluster. This page contains information on how to use the CUPI API to view cluster information.

## Listing Cisco Unity Connection Cluster locations

The following is an example of a GET that displays cluster information:

## Listing Cisco Unity Connection Cluster locations

```
GET http://<connection-server>/vmrest/cluster
```

The following is the response from the above GET request:

```
200
OK
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<VmsServers total="2">
  <VmsServer>
    <URI>/vmrest/vmsservers/3ee10726-9634-4066-bfcb-c936bc074be6</URI>
    <ObjectId>3ee10726-9634-4066-bfcb-c936bc074be6</ObjectId>
    <ServerName> qa-ks-vm-279</ServerName>
    <IpAddress>10.65.156.193</IpAddress>
    <VmsServerObjectId>3ee10726-9634-4066-bfcb-c936bc074be6</VmsServerObjectId>
    <ClusterMemberId>0</ClusterMemberId>
    <ServerState>1</ServerState>
    <HostName> qa-ks-vm-279.cisco.com</HostName>
    <ServerDisplayState>3</ServerDisplayState>
    <IpAddressV6>fe80::250:56ff:feab:1c6</IpAddressV6>
    <SubToPerformReplicationRole>>false</SubToPerformReplicationRole>
  </VmsServer>
  <VmsServer>
    <URI>/vmrest/vmsservers/a57a68c6-782f-4757-945c-270e0fba2734</URI>
    <ObjectId>a57a68c6-782f-4757-945c-270e0fba2734</ObjectId>
    <ServerName> qa-ks-vm-279.cisco.com </ServerName>
    <IpAddress>10.65.156.252</IpAddress>
    <VmsServerObjectId>a57a68c6-782f-4757-945c-270e0fba2734</VmsServerObjectId>
    <ClusterMemberId>1</ClusterMemberId>
    <ServerState>8</ServerState>
    <HostName> qa-ks-vm-279.cisco.com </HostName>
    <ServerDisplayState>4</ServerDisplayState>
    <IpAddressV6>fe85::450:56ff:feab:1c6</IpAddressV6>
    <SubToPerformReplicationRole>>false</SubToPerformReplicationRole>
  </VmsServer>
</VmsServers>
```

In case of a standalone location, only the data related to that location gets displayed.

### JSON Example

The following is an example of a GET that displays the list of servers in the cluster:

```
GET http://<connection-server>/vmrest/vmsservers
Accept: application/json
Content-type: application/json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```

200
OK
{
  "@total": "2",
  "VmsServer": [
    {
      "URI": "/vmrest/vmsservers/ef03ca05-3c05-4465-9985-9de371e2e94e",
      "ObjectId": "ef03ca05-3c05-4465-9985-9de371e2e94e",
      "ServerName": "qa-ks-vm-19",
      "IpAddress": "10.78.171.134",
      "VmsServerObjectId": "ef03ca05-3c05-4465-9985-9de371e2e94e",
      "ClusterMemberId": "0",
      "ServerState": "20",
      "HostName": "qa-ks-vm-19.cisco.com",
      "ServerDisplayState": "3",
      "IpAddressV6": "fe80::250:56ff:feab:1c6"
      "SubToPerformReplicationRole": "false"
    },
    {
      "URI": "/vmrest/vmsservers/a57a68c6-782f-4757-945c-270e0fba2734",
      "ObjectId": "a57a68c6-782f-4757-945c-270e0fba2734",
      "ServerName": "qa-ks-vm-279.cisco.com",
      "IpAddress": "10.65.156.25",
      "VmsServerObjectId": "a57a68c6-782f-4757-945c-270e0fba2734",
      "ClusterMemberId": "1",
      "ServerState": "8",
      "HostName": "qa-ks-vm-279.cisco.com",
      "ServerDisplayState": "4",
      "IpAddressV6": "fe85::450:56ff:feab:1c6"
      "SubToPerformReplicationRole": "false"
    }
  ]
}

```

## Viewing a specific location in Cisco Unity Connection Cluster

Administrator can use this API to fetch details of a particular server in a Unity Connection cluster. The following is an example of the GET request :

```
GET https://<connection-server>/vmrest/vmsservers/<vmsServerObjectId>
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```

200
ok
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<VmsServer>
  <URI>/vmrest/vmsservers/3ee10726-9634-4066-bfcb-c936bc074be6</URI>
  <ObjectId>3ee10726-9634-4066-bfcb-c936bc074be6</ObjectId>
  <ServerName> qa-ks-vm-279</ServerName>
  <IpAddress>10.65.156.193</IpAddress>
  <VmsServerObjectId>3ee10726-9634-4066-bfcb-c936bc074be6</VmsServerObjectId>
  <ClusterMemberId>0</ClusterMemberId>
  <ServerState>1</ServerState>
  <HostName> qa-ks-vm-279.cisco.com</HostName>
  <ServerDisplayState>3</ServerDisplayState>
  <IpAddressV6>fe85::450:56ff:feab:1c6</IpAddressV6>
  <SubToPerformReplicationRole>>false</SubToPerformReplicationRole>
</VmsServer>

```

### JSON Example

The following is an example of the GET command to view the details of particular server in the cluster.

```
GET https://< connection-server >/vmrest/vmsservers/<vmsServerObjectId>
Accept: application/json
Content-type: application/json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
200
Ok
{
  {
    "URI": "/vmrest/vmsservers/ef03ca05-3c05-4465-9985-9de371e2e94e",
    "ObjectId": "ef03ca05-3c05-4465-9985-9de371e2e94e",
    "ServerName": " qa-ks-vm-19",
    "IpAddress": "10.78.171.134",
    "VmsServerObjectId": "ef03ca05-3c05-4465-9985-9de371e2e94e",
    "ClusterMemberId": "0",
    "ServerState": "20",
    "HostName": " qa-ks-vm-19.cisco.com",
    "ServerDisplayState": "3",
    "IpAddressV6": "fe80::250:56ff:feab:1c6"

    "SubToPerformReplicationRole": "false"
  }
}
```

## Explanation of Data Fields

Parameter	Data Type	Operations	Comments
Object id	char	Read Only	Acts as a primary key for the API. The ObjectID is a unique system-generated identifier for a VMSServer object.
ServerName	nvarchar	Read Only	Specifies the name of the server.
IpAddress	varchar	Read Only	Specifies the IP address of the server.
VmsServerObjectId	char	Read Only	Acts as a primary key for the API. The VmsServerObjectId is a unique system-generated identifier for a VMSServer object.
ClusterMemberId	smallint	Read Only	Specifies a unique number assigned to each server in the cluster as per their failover order.  Values can be: <ul style="list-style-type: none"> <li>• 0-Primary</li> <li>• 1-Secondary</li> </ul>



Parameter	Data Type	Operations	Comments
ServerState	int	Read Only	<p>Specifies the current state of the server.</p> <p>Values can be:</p> <ul style="list-style-type: none"> <li>• 0-Pri_Init</li> <li>• 1-Pri_Active</li> <li>• 2-Pri_Act_Secondary</li> <li>• 3-Pri_Idle</li> <li>• 4-Pri_Failover</li> <li>• 5-Pri_Takeover</li> <li>• 6-Pri_SBR</li> <li>• 7-Sec_Init</li> <li>• 8-Sec_Active</li> <li>• 9-Sec_Act_Primary</li> <li>• 10-Sec_Idle</li> <li>• 11-Sec_Takeover</li> <li>• 12-Sec_Failover</li> <li>• 13-Sec_SBR</li> <li>• 14-Db_Sync</li> <li>• 15-Set_Peer_Idle</li> <li>• 16-Undefined</li> <li>• 17-Pri_Active_Disconnected</li> <li>• 18-Pri_Connecting</li> <li>• 19-Pri_Choose_Role</li> <li>• 20-Pri_Single_Server</li> <li>• 21-Sec_Act_Primary_Disconnected</li> <li>• 22-Sec_Connecting</li> <li>• 23-Sec_Choose_Role</li> <li>• 24-Shutdown</li> </ul>
HostName	varchar	Read Only	Indicates the hostname of the VMS server
ServerDisplayState	int	Read Only	<p>Specifies the values for the admin to display current server status.</p> <p>Values can be:</p> <ul style="list-style-type: none"> <li>• 0 - UNKNOWN</li> <li>• 1 - DOWN</li> <li>• 2 - INITIALIZING</li> <li>• 3 - PRIMARY</li> <li>• 4 - SECONDARY</li> <li>• 5 - IDLE</li> <li>• 6 - IN_DB_SYNC</li> <li>• 7 - IN_SBR</li> </ul>

Parameter	Data Type	Operations	Comments
IpAddressV6	varchar	Read Only	Specifies the IPV6 address of the server.
SubscriberReplicationRole	boolean	Read Only	Note - Defined for future purpose. Indicates whether the subscriber machine is enabled for directory replication; On Publisher machine its value is always false. Values can be: <ul style="list-style-type: none"> <li>• 0- false</li> <li>• 1- true</li> </ul>

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Configuration Values

## About Configuration Values

This page contains information on how to use the API to read configuration values which are also referred to system settings. You cannot create or delete configuration values.

Most configuration values are read-only in the API. Beginning with Unity Connection 8.5, following are the configuration values that can be updated:

- System.API.CumiAccessSecureMessageAttachments
- System.API.CumiAllowSecureMessageHeaders
- System.Conversations.UserMaxConcurrentSessionsTUI (Unity Connection 11.5(1) and later only)
- System.SA.UserInactivityTimeout (Unity Connection 11.5(1) and later only)

## Listing and Viewing

The following is an example of **GET** that lists all system configuration values:

```
GET https://<server>/vmrest/configurationvalues
```

Beginning with Connection 8.5, you can perform **GET** on individual settings using their full name in the URI (instead of getting the whole collection):

```
GET https://<server>/vmrest/configurationvalues/<full name>
```

You can access the configuration values as part of a list using the regular configuration value query. For example, to find all configuration values with a fullname that starts with "System.API.Cumi":

```
GET https://<server>/vmrest/configurationvalues?query=(fullname startswith System.API.Cumi)
```

## Updating

Only following configuration values have write access:

- System.API.CumiAccessSecureMessageAttachments
- System.API.CumiAllowSecureMessageHeaders
- System.Conversations.UserMaxConcurrentSessionsTUI (Unity Connection 11.5(1) and later only)
- System.SA.UserInactivityTimeout (Unity Connection 11.5(1) and later only)



---

**Note** For System.Conversations.UserMaxConcurrentSessionsTUI, the minimum configuration value is 0 and maximum configuration value is 99. Whereas for System.SA.UserInactivityTimeout, the minimum configuration value is 0 and maximum configuration value is 9999 (value 0 auto disables the feature).

---

Attempting **PUT** on any other configuration values will result in a 403 (access denied) error.

When modifying a configuration value, only the value field is used (see PUT examples below). All other fields cannot be modified and will be ignored.

## HTTP Examples

### GET Examples

Example of **GET** for a list:

```

GET /vmrest/configurationvalues?query=(fullname%20startswith%20System.API.Cumi) HTTP/1.1
Accept: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Connection: keep-alive

HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=658B27F18EF8424C11D261FEC776285D; Path=/
Set-Cookie: JSESSIONID=DE6BE806075303C8A73E666E0B478725; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 20 May 2010 10:26:42 GMT
Server:

[REDACTED]

XML

<ConfigurationValues total="2">
  <ConfigurationValue>
    <Type>11</Type>
    <LastModifiedTime>2010-05-21T17:29:52Z</LastModifiedTime>
    <LastModifiedByComponent>VMREST</LastModifiedByComponent>
    <FullName>System.API.CumiAccessSecureMessageAttachments</FullName>
    <Value>1</Value>
    <UserSetting>true</UserSetting>
    <MinVal>0</MinVal>
    <MaxVal>0</MaxVal>
    <RequiresRestart>false</RequiresRestart>
  </ConfigurationValue>
  <ConfigurationValue>
    <Type>11</Type>
    <FullName>System.API.CumiAllowSecureMessageHeaders</FullName>
    <Value>1</Value>
    <UserSetting>true</UserSetting>
    <MinVal>0</MinVal>
    <MaxVal>0</MaxVal>
    <RequiresRestart>false</RequiresRestart>
  </ConfigurationValue>
</ConfigurationValues>

```

Example of **GET** for individual setting:

```

GET /vmrest/configurationvalues/System.Messaging.RelaySecureMessage HTTP/1.1
Accept: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRtaW5pc3RyYXRvcjplY3NidWxhYg==

HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=B7839B7E290A9763ABD7751A6FBCAA5C; Path=/
Set-Cookie: JSESSIONID=E01E5B7E9F57692810ECF8AEAEDBD2B0; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 20 May 2010 10:39:48 GMT
Server:

{"Type":11,"LastModifiedTime":"2005-2010-11-02","LastModifiedByComponent":"CAMN","FullName":"System.Messaging.RelaySecureMessage","Value":0,"UserSetting":true,"RequiresRestart":false"}

XML

<ConfigurationValue>
  <Type>11</Type>
  <FullName>System.Messaging.RelaySecureMessage</FullName>
  <Value>0</Value>
  <UserSetting>true</UserSetting>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>

```

#### Example of GET for User Inactivity Timeout:

```

GET /vmrest/configurationvalues/System.SA.UserInactivityTimeout
HTTP/1.1 200 OK
Cache-Control: private
Expires: Thu, 01 Jan 1970 05:30:00 IST
Set-Cookie: JSESSIONIDSSO=F41326B2A9A8472368FE580917E944B7; Path=/; Secure; HttpOnly
Set-Cookie: JSESSIONID=CEADA77CB21BD27EE1C3386B3CFED149; Path=/vmrest/; Secure; HttpOnly
Set-Cookie: REQUEST_TOKEN_KEY=-1968493036399359092; Path=/; Secure; HttpOnly
X-Frame-Options: SAMEORIGIN
Content-Type: application/xml
Content-Length: 400
Date: Thu, 10 Dec 2015 06:38:42 GMT
Server:

<?xml version="1.0" encoding="UTF-8"
xmlns:cu="http://schemas.cisco.com/2008/03/unity-services" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://schemas.cisco.com/2008/03/unity-services http://schemas.cisco.com/2008/03/unity-services.xsd">
  <ConfigurationValue>
    <Type>3</Type>
    <LastModifiedTime>2015-12-10T06:23:26Z</LastModifiedTime>
    <LastModifiedByComponent>VMREST</LastModifiedByComponent>
    <FullName>System.SA.TUIInterfaceDisable</FullName>
    <Value>55</Value>
    <UserSetting>true</UserSetting>
    <MinVal>1</MinVal>
    <MaxVal>500</MaxVal>
    <RequiresRestart>false</RequiresRestart>
  </ConfigurationValue>

```

## PUT Examples

An example of **PUT** for configuration value:

```
PUT /vmrest/configurationvalues/System.API.CumiAccessSecureMessageAttachments HTTP/1.1
Content-Type: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 13

{"Value":"1"}

XML

<ConfigurationValue>
  <Value>0</Value>
</ConfigurationValue>

HTTP/1.1 204 No Content
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=E3EB2F2AB5593902F00ECE80298ED82B; Path=/
Set-Cookie: JSESSIONID=4B3EA5586E13B955D2CC9A8C46EE12FE; Path=/vmrest
Date: Thu, 20 May 2010 10:58:05 GMT
Server:
```

An example of **PUT** for configuration values that does not support PUT:

```
PUT /vmrest/configurationvalues/System.Messaging.RelaySecureMessage HTTP/1.1
Content-Type: application/json
User-Agent: Java/1.6.0_20
Host: cuc-install-69.cisco.com
Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/*; q=.2
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 240

HTTP/1.1 403 Forbidden
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=6BE354F4A2C20A2190C9DF78D91D7AB; Path=/
Set-Cookie: JSESSIONID=32DF55D02DA33C43938F43C2FC7A13FF; Path=/vmrest
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Thu, 20 May 2010 10:26:43 GMT
Server:
```

An example of **PUT** for User Inactivity Timeout:

```

PUT /vmrest/configurationvalues/System.SA.UserInactivityTimeout

HTTP/1.1 204 No Content
Cache-Control: private
Expires: Thu, 01 Jan 1970 05:30:00 IST
Set-Cookie: JSESSIONIDSSO=2F68733BA443959388E841535A717DCC; Path=/; Secure; HttpOnly
Set-Cookie: JSESSIONID=A982BB1A93DAB2C294A64BD5D7DD03D0; Path=/vmrest/; Secure; HttpOnly
Set-Cookie: REQUEST_TOKEN_KEY=138477933938641394; Path=/; Secure; HttpOnly
X-Frame-Options: SAMEORIGIN
Content-Type: application/xml
Date: Thu, 10 Dec 2015 06:55:12 GMT
Server:

XML

<ConfigurationValue>
  <Value>36</Value>
</ConfigurationValue>

```

An example of **PUT** for TUI/VUI Logon Session Limit:

```

PUT /vmrest/configurationvalues/System.Conversations.UserMaxConcurrentSessionsTUI

XML

<ConfigurationValue>
  <Value>3</Value>
</ConfigurationValue>

```

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Authentication Rules

## About Authentication Rules

This section contains information on how to create, list, modify, and delete Authentication Rules. In Cisco Unity Connection, authentication rules govern user passwords, PINs, and account lockouts for all user accounts. You use authentication rules to secure how users access Connection by phone, and how users access Cisco Unity Connection Administration and the Cisco Personal Communications Assistant (PCA).

For example, an authentication rule determines:

- The number of failed sign-in attempts that are allowed before an account is locked
- The number of minutes an account remains locked before it is reset
- Whether a locked account must be unlocked manually by an administrator
- The minimum length allowed for passwords and PINs
- The number of days before a password or PIN expires

## Listing Authentication Rules

### Example 1

The following is an example of the GET request that lists the Authentication Rules:

```
https://<connection_server>/vmrest/authenticationrules
```

The following is an example of the response from the above \*GET\* request and the actual response will depend upon the information given by you:



```

<AuthenticationRules total="2">
  <AuthenticationRule>
    <URI>/vmrest/authenticationrules/db3999dc-7afc-4a1e-be64-c5647c7f2226</URI>
    <ObjectId>db3999dc-7afc-4a1e-be64-c5647c7f2226</ObjectId>
    <HackResetTime>30</HackResetTime>
    <LocationObjectId>be27976c-6e6e-419f-853d-b3764881dfb0</LocationObjectId>

  <LocationURI>/vmrest/locations/connectionlocations/be27976c-6e6e-419f-853d-b3764881dfb0</LocationURI>

    <LockoutDuration>30</LockoutDuration>
    <MaxDays>120</MaxDays>
    <MaxHacks>7</MaxHacks>
    <MinLength>8</MinLength>
    <PrevCredCount>5</PrevCredCount>
    <TrivialCredChecking>true</TrivialCredChecking>
    <DisplayName>Papa Recommended Web Application Authentication Rule</DisplayName>
    <MinDuration>1440</MinDuration>
    <ExpiryWarningDays>15</ExpiryWarningDays>
    <MinCharsToChange>1</MinCharsToChange>
  </AuthenticationRule>
  <AuthenticationRule>
    <URI>/vmrest/authenticationrules/df3611a3-d372-4a38-a7a4-7d4aba4febd2</URI>
    <ObjectId>df3611a3-d372-4a38-a7a4-7d4aba4febd2</ObjectId>
    <HackResetTime>30</HackResetTime>
    <LocationObjectId>be27976c-6e6e-419f-853d-b3764881dfb0</LocationObjectId>

  <LocationURI>/vmrest/locations/connectionlocations/be27976c-6e6e-419f-853d-b3764881dfb0</LocationURI>

    <LockoutDuration>30</LockoutDuration>
    <MaxDays>180</MaxDays>
    <MaxHacks>3</MaxHacks>
    <MinLength>6</MinLength>
    <PrevCredCount>5</PrevCredCount>
    <TrivialCredChecking>true</TrivialCredChecking>
    <DisplayName>Recommended Voice Mail Authentication Rule</DisplayName>
    <MinDuration>1440</MinDuration>
    <ExpiryWarningDays>15</ExpiryWarningDays>
    <MinCharsToChange>1</MinCharsToChange>
  </AuthenticationRule>
  <AuthenticationRule>
    <URI>/vmrest/authenticationrules/5780fb3f-8f70-4d3d-9c18-2832f9d11642</URI>
    <ObjectId>5780fb3f-8f70-4d3d-9c18-2832f9d11642</ObjectId>
    <HackResetTime>30</HackResetTime>
    <LocationObjectId>be27976c-6e6e-419f-853d-b3764881dfb0</LocationObjectId>

  <LocationURI>/vmrest/locations/connectionlocations/be27976c-6e6e-419f-853d-b3764881dfb0</LocationURI>

    <LockoutDuration>30</LockoutDuration>
    <MaxDays>180</MaxDays>
    <MaxHacks>3</MaxHacks>
    <MinLength>8</MinLength>
    <PrevCredCount>12</PrevCredCount>
    <TrivialCredChecking>true</TrivialCredChecking>
    <DisplayName>test</DisplayName>
    <MinDuration>0</MinDuration>
    <ExpiryWarningDays>15</ExpiryWarningDays>
    <MinCharsToChange>1</MinCharsToChange>
  </AuthenticationRule>
  <AuthenticationRule>
    <URI>/vmrest/authenticationrules/ac80ee1d-0f09-42d0-ae7c-adb56a82b340</URI>
    <ObjectId>ac80ee1d-0f09-42d0-ae7c-adb56a82b340</ObjectId>
    <HackResetTime>30</HackResetTime>
    <LocationObjectId>be27976c-6e6e-419f-853d-b3764881dfb0</LocationObjectId>

```

```
<LocationURI>/vmrest/locations/connectionlocations/be27976c-6e6e-419f-853d-b3764881dfb0</LocationURI>

  <LockoutDuration>30</LockoutDuration>
  <MaxDays>180</MaxDays>
  <MaxHacks>3</MaxHacks>
  <MinLength>8</MinLength>
  <PrevCredCount>12</PrevCredCount>
  <TrivialCredChecking>true</TrivialCredChecking>
  <DisplayName>Papa Recommended Web Application Authentication Rule Paapas</DisplayName>
  <MinDuration>1440</MinDuration>
  <ExpiryWarningDays>15</ExpiryWarningDays>
  <MinCharsToChange>1</MinCharsToChange>
</AuthenticationRule>
</AuthenticationRules>
```

Response code: 200

### Example 2

The following is an example of the GET request that lists a specified Authentication Rule as represented by <objectId>:

```
https://<connection_server>/vmrest/authenticationrules/<objectId>
```

The following is an example of the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<AuthenticationRule>
  <URI>/vmrest/authenticationrules/db3999dc-7afc-4a1e-be64-c5647c7f2226</URI>
  <ObjectId>db3999dc-7afc-4a1e-be64-c5647c7f2226</ObjectId>
  <HackResetTime>30</HackResetTime>
  <LocationObjectId>be27976c-6e6e-419f-853d-b3764881dfb0</LocationObjectId>

  <LocationURI>/vmrest/locations/connectionlocations/be27976c-6e6e-419f-853d-b3764881dfb0</LocationURI>

  <LockoutDuration>30</LockoutDuration>
  <MaxDays>120</MaxDays>
  <MaxHacks>7</MaxHacks>
  <MinLength>8</MinLength>
  <PrevCredCount>5</PrevCredCount>
  <TrivialCredChecking>true</TrivialCredChecking>
  <DisplayName>Papa Recommended Web Application Authentication Rule</DisplayName>
  <MinDuration>1440</MinDuration>
  <ExpiryWarningDays>15</ExpiryWarningDays>
  <MinCharsToChange>1</MinCharsToChange>
</AuthenticationRule>
```

Response code: 200

## Adding a new Authentication Rule

The following is an example of the POST request that lists the Authentication Rules:

```
https://<connection_server>/vmrest/authenticationrules
```

The following is an example of the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```

<AuthenticationRule>
<DisplayName>New Authentication Rule</DisplayName>
<HackResetTime>30</HackResetTime>
  <LockoutDuration>30</LockoutDuration>
  <MaxDays>120</MaxDays>
  <MaxHacks>7</MaxHacks>
  <MinLength>8</MinLength>
  <PrevCredCount>5</PrevCredCount>
  <TrivialCredChecking>true</TrivialCredChecking>
  <MinDuration>1440</MinDuration>
  <ExpiryWarningDays>15</ExpiryWarningDays>
  <MinCharsToChange>1</MinCharsToChange>
</AuthenticationRule>

```

Response code: 201

## Modifying Authentication Rules

The following is an example of the PUT request that modifies the Authentication Rule:

```
https://<connection_server>/vmrest/ authenticationrules/<objectId>
```

The actual response will depend upon the information given by you.

```

<AuthenticationRule> <AuthenticationRule>
<HackResetTime>60</HackResetTime>
  <LockoutDuration>60</LockoutDuration>
  <MaxDays>120</MaxDays>
  <MaxHacks>7</MaxHacks>
  <MinLength>8</MinLength>
  <PrevCredCount>5</PrevCredCount>
  <TrivialCredChecking>true</TrivialCredChecking>
  <MinDuration>1440</MinDuration>
  <ExpiryWarningDays>15</ExpiryWarningDays>
  <MinCharsToChange>1</MinCharsToChange>
</AuthenticationRule>
<HackResetTime>60</HackResetTime>
  <LockoutDuration>60</LockoutDuration>
  <MaxDays>120</MaxDays>
  <MaxHacks>7</MaxHacks>
  <MinLength>8</MinLength>
  <PrevCredCount>5</PrevCredCount>
  <TrivialCredChecking>true</TrivialCredChecking>
  <MinDuration>1440</MinDuration>
  <ExpiryWarningDays>15</ExpiryWarningDays>
  <MinCharsToChange>1</MinCharsToChange>
</AuthenticationRule>

```

Response code: 204

## Deleting Authentication Rules

The following is an example of the DELETE request that deletes a Authentication Rule as represented by <objectId>:

```
https://<connection_server>/vmrest/authenticationrules/<objectId>
```

The output for this request returns the successful response code.

Response Code: 204
--------------------

## Explanation of Data Fields

The following chart lists all of the data fields available on Authentication Rules.

Field Name	Operation	Values	Comments
HackResetTime	Read/Write	1-120	Reset Every Failed Sign-in Attempts. Default: 30
LockoutDuration	Read/Write	0-1440	LockoutDuration. Default: 30
MaxDays	Read/Write	0-3653	Credential Expires After. Default=180
MaxHacks	Read/Write	0-100	Failed Sign-In. Default=3
MinLength	Read/Write	1-64	Minimum credential length. Default: 8
PrevCredCount	Read/Write	0-25	Stored Number of Previous Credentials. Default=12
TrivialCredChecking	Read/Write	true/false	Check for Trivial Passwords. Default=false
DisplayName	Read/Write	1-64	Friendly name for the Authentication Rule.
MinDuration	Read/Write	0-129600	Minimum Duration between Credential Changes
ExpiryWarningDays	Read/Write		Expiration Warning Days. Default=15
MinCharsToChange	Read/Write	1-64	Minimum Number of Character Changes between Successive Credentials. Default: 1

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Roles

## Roles

A role comprises of set of privileges that define the access level to the system. System Administrator can configure multiple roles based on the administrative needs. The role assignment for a user account can be done based on the set of operations required. Unity Connection offers two types of roles:

- **System Roles:** System roles are predefined roles that come installed with Unity Connection. You cannot create, modify or delete these roles. System roles can only be assigned or unassigned to users by System Administrator.
- **Custom Roles:** Custom roles are the roles that you create with a list of privileges based on your organizational requirements. Custom roles can be assigned or unassigned to users by System Administrator or a custom role user with role assignment privilege



**Note** Only a user with System Administrator role can create, update or delete custom roles.

## Listing Privileges

The following is an example of the GET request that lists all the privileges:

```
GET https://<connection-server>/vmrest/privileges
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<Privileges total="46">

  <Privilege>
    <ObjectId>56ea5443-4832-40b9-88b7-879068465bc9</ObjectId>
    <DisplayName>Manage Users - Full Access</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>37ae97bf-a0ca-4adf-a498-alaf3f4d4acd</ObjectId>
    <DisplayName>Manage Users: Assign/Unassign Roles</DisplayName>
  </Privilege>

  .....
  .....
  .....

  <Privilege>
    <ObjectId>140a9af7-b76a-49de-9ed3-a020bcd2c1bb</ObjectId>
    <DisplayName>Reset User Passwords</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>7c0326f7-e7e6-4853-8e4e-b93dd1c13ce6</ObjectId>
    <DisplayName>Run Serviceability Page</DisplayName>
  </Privilege>
</Privileges>
```

```
Response Code: 200
```

## Listing Roles

This API list all the System roles as well as Custom roles. The following is an example of the GET request that lists all the Roles:

```
GET https://<connection-server>/vmrest/roles
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

## Listing System Roles

```

<Roles total="19">
<Role>
<URI>/vmrest/roles/0d44b89f-6b93-4f16-a8cb-cd07c01e524d</URI>
<ObjectId>0d44b89f-6b93-4f16-a8cb-cd07c01e524d</ObjectId>
<RoleName>Audio Text Administrator</RoleName>
<RoleDescription>Administers call handlers, directory handlers and interview
handlers</RoleDescription>
<IsEnabled>true</IsEnabled>
<ReadOnly>>false</ReadOnly>
</Role>
.....
.....
.....
.....

<Role>
<URI>/vmrest/roles/58f51c8a-74e4-4db9-babc-8e7e9aad4385</URI>
<ObjectId>58f51c8a-74e4-4db9-babc-8e7e9aad4385</ObjectId>
<RoleName>New Custom Role</RoleName>
<RoleDescription>New custom role for users</RoleDescription>
<IsEnabled>true</IsEnabled>
<ReadOnly>>false</ReadOnly>
<InheritedObjectId>dc32b8c1-bc71-4a81-9538-72ebb88a3f31</InheritedObjectId>
</Role>
</Roles>

```

Response Code: 200

## Listing System Roles

The following is an example of the GET request that lists all the system roles:

```
GET https://<connection-server>/vmrest/systemroles
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```

<Roles total="11">
  <Role>
    <ObjectId>7c99cd0b-c480-483d-a7d1-44824d50903d</ObjectId>
    <RoleName>Audio Text Administrator</RoleName>
    <RoleDescription>Administers call handlers, directory handlers and interview
handlers</RoleDescription>
    <IsEnabled>>true</IsEnabled>
    <ReadOnly>>false</ReadOnly>
  </Role>
  <Role>
    <ObjectId>4f077e4e-61c7-4ce8-a58a-2c4bc6089319</ObjectId>
    <RoleName>Greeting Administrator</RoleName>
    <RoleDescription>Manages call handler recorded greetings via TUI</RoleDescription>
    <IsEnabled>>true</IsEnabled>
    <ReadOnly>>false</ReadOnly>
  </Role>
  .....
  .....
  .....
  <Role>
    <ObjectId>15930390-7ef1-4389-abc0-020962e32cc5</ObjectId>
    <RoleName>Read Only Administrator</RoleName>
    <RoleDescription>Read only access to view all Connection administrative
functions</RoleDescription>
    <IsEnabled>>true</IsEnabled>
    <ReadOnly>>false</ReadOnly>
  </Role>

```

Response Code: 200

## Listing Custom Roles

The following is an example of the GET request that lists all the custom roles:

```
GET https://<connection-server>/vmrest/customroles
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```

<Roles total="4">
  <Role>
    <ObjectId>5c5abfb9-eec9-4df5-877d-972f65b73df6</ObjectId>
    <RoleName>Custom Role 1</RoleName>
    <RoleDescription>New custom role for user 1</RoleDescription>
    <IsEnabled>true</IsEnabled>
    <ReadOnly>false</ReadOnly>
    <InheritedObjectId>5976fb56-e398-4659-841a-5519c97a036f</InheritedObjectId>
  </Role>
  .....
  .....
  .....

  <Role>
    <ObjectId>c616cc89-16f0-4fe5-9213-c5ce4ea5e81e</ObjectId>
    <RoleName>Custom role 4</RoleName>
    <RoleDescription>New custom role for user 4</RoleDescription>
    <IsEnabled>true</IsEnabled>
    <ReadOnly>false</ReadOnly>
    <InheritedObjectId>dc32b8c1-bc71-4a81-9538-72ebb88a3f31</InheritedObjectId>
  </Role>
</Roles>

```

Response Code: 200

## Creating a New Custom Role

The following is an example of POST request that creates a new Custom Role.

```

POST
https://<connection-server>/vmrest/customroles?systemRoleToInherit=<systemRoleToInherit_objId>

```



**Note** In **systemRoleToInherit** field, provide the object id of the System Role that you want to inherit while creating a Custom Role. The privileges of inherited system role are assigned to the custom role. Below are the system roles that can be inherited:

- Audio Text Administrator
- Greeting Administrator
- Help Desk Administrator
- Technician
- User Administrator
- Tenant Administrator

### Request Body



```
<Role>
<RoleName>Custom Role New</RoleName>
<RoleDescription>Administers call handlers, directory handlers and interview
handlers</RoleDescription>
<AssignPrivileges>
  <PrivilegeObjectId>0db37044-b459-4784-a82c-fea1ddb35b73</PrivilegeObjectId>
  <PrivilegeObjectId>0d20ccc4-2694-4762-bfaa-649871fd0e99</PrivilegeObjectId>
</AssignPrivileges>
</Role>
```

The following is the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
Response Code: 201
/vmrest/customroles/9166f041-aec4-4e5a-9600-6cc27ef99ec7
```

## Modifying Custom Role

The following is an example of the PUT request that modifies the Custom Roles.

```
https://<connection_server>/vmrest/customroles/<objectId>
```

### Request body

- *When Inherit System Role is Modified*

```
<Role>
<AssignPrivileges>
  <PrivilegeObjectId>0db37044-b459-4784-a82c-fea1ddb35b73</PrivilegeObjectId>
  <PrivilegeObjectId>0d20ccc4-2694-4762-bfaa-649871fd0e99</PrivilegeObjectId>
</AssignPrivileges>
<InheritedObjectId>61a3d5bd-74a0-40e4-a770-f927faa52bb3</InheritedObjectId>
</Role>
```

- *When Privileges are Modified (Assigned or Removed)*

```
<Role>
<AssignPrivileges>
  <PrivilegeObjectId>0db37044-b459-4784-a82c-fea1ddb35b73</PrivilegeObjectId>
  <PrivilegeObjectId>0d20ccc4-2694-4762-bfaa-649871fd0e99</PrivilegeObjectId>
</AssignPrivileges>
<RemovePrivileges>
  <PrivilegeObjectId>93c8c92f-a6c7-4458-bbde-1b28dcc5adc6</PrivilegeObjectId>
</RemovePrivileges>
</Role>
```

```
Response code: 201
```

## Listing Privileges for Roles

Using this API you can get all the privileges of System and Custom roles.

The following is an example of the GET request that lists all the privileges of roles:

```
https://<connection-server>/vmrest/roles/<object_id>/privileges
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<Privileges total="7">
  <Privilege>
    <ObjectId>f34d7ee3-43a0-411a-8de9-724937ea8fc8</ObjectId>
    <DisplayName>Manage Call Handler Templates And System Call Handlers - View, Create,
Update</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>7ea8e97a-f620-4fdf-b70f-7eaff092647f</ObjectId>
    <DisplayName>Call Management: Directory Handlers - View, Create, Update</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>62778f0d-cdc5-495c-9602-1e95b1c5e641</ObjectId>
    <DisplayName>Call Management: Interview Handlers - View, Create, Update</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>fef92b42-77c1-40aa-99df-fafdf793e763</ObjectId>
    <DisplayName>Call Management: Call Routing Rules - Full Access</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>8979151b-8554-4483-b75b-d46d56e1fb98</ObjectId>
    <DisplayName>Networking: VPIM - Full Access</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>2431bc0b-1251-4c95-aaf9-2a82a893a60d</ObjectId>
    <DisplayName>Distribution Lists - Full Access</DisplayName>
  </Privilege>
  <Privilege>
    <ObjectId>2c2ef5b5-b15c-42b3-9cc3-f7d4de137825</ObjectId>
    <DisplayName>System Settings: Advanced - Full Access</DisplayName>
  </Privilege>
</Privileges>
```

Response Code: 200

## Deleting a Custom Role

The following is an example of the DELETE request that deletes a Custom Roles:

```
DELETE https://<connection-server>/vmrest/customroles/<objectId>
```

The following is the response from the above \*DELETE\* request and the actual response will depend upon the information given by you:

Response Code: 204

## Explanation of Data Fields

The following chart lists all of the data fields:

Parameter	Data Type	Operations	Description
RoleName	String	Read/Write	Specifies the name of the Role.
RoleDescription	String	Read/Write	Specifies the description of the Role.

Parameter	Data Type	Operations	Description
InheritedObjectId	String	Read/Write	Specifies the object id of the System Role that is inherited while creating or modifying a Custom Role.
AssignPrivileges	String	Read/Write	Specifies the object id of the privileges that are assigned to the Custom Role.
RemovePrivileges	String	Write only	Specifies the object id of the privileges that will be removed from a custom role.  <b>Note</b> You can not remove the privileges of inherited System Role.

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Restriction Patterns

## Restriction Patterns

Administrator can use this API to create/update/delete/fetch the restriction pattern. Various attributes of restriction pattern can also be updated using this API.

### Listing the Restriction Pattern for a Particular Restriction Table

The following is an example of the GET request that lists all the restriction tables:

```
GET https://<connection-server>/vmrest/restrictiontables
/<restrictiontableobjectid>/restrictionpatterns
```

The following is the response from the above \*GET\* request:

```
<RestrictionPatterns total="2">
  <RestrictionPattern>
    <URI>/vmrest/restrictiontables/38de2a1f-ca74-4be3-bb7cd315df4c0fc5/
restrictionpatterns/6564adb9-9090-42e0-81e6-04e132c4382c</URI>
    <Blocked>>true</Blocked>
    <NumberPattern>*</NumberPattern>
    <RestrictionTableObjectId>38de2a1f-ca74-4be3-bb7cd315df4c0fc5</
RestrictionTableObjectId>
    <SequenceNumber>1</SequenceNumber>
    <ObjectId>6564adb9-9090-42e0-81e6-04e132c4382c</ObjectId>
  </RestrictionPattern>
  <RestrictionPattern>
    <URI>/vmrest/restrictiontables/38de2a1f-ca74-4be3-bb7cd315df4c0fc5/
restrictionpatterns/15a68e67-16b9-4847-8230-29be9baa88cf</URI>
    <Blocked>>true</Blocked>
    <NumberPattern>*</NumberPattern>
    <RestrictionTableObjectId>38de2a1f-ca74-4be3-bb7cd315df4c0fc5</
RestrictionTableObjectId>
    <SequenceNumber>0</SequenceNumber>
    <ObjectId>15a68e67-16b9-4847-8230-29be9baa88cf</ObjectId>
  </RestrictionPattern>
</RestrictionPatterns>
```

```
Response Code: 200
```

## Viewing the Specific Restriction Pattern for a Particular Restriction Table

The following is an example of the GET request that lists the details of specific restriction pattern represented by the provided value of object ID:

```
GET https://<connection-server>/vmrest/restrictiontables/
  <restrictiontableobjectid>/restrictionpatterns/<restrictiontableobjectid>
The following is the response from the above *GET* request:
<RestrictionPattern>
  <URI>/vmrest/restrictiontables/2c2c9504-8fb4-44e3-983c-
  93eb4e20325c/restrictionpatterns/2c2c9504-8fb4-44e3-983c-93eb4e20325c/</URI>
  <Blocked>false</Blocked>
  <NumberPattern>*</NumberPattern>
  <RestrictionTableObjectId>255bfdd2-5686-47f2-b40a-
  320c194521ba</RestrictionTableObjectId>
  <RestrictionTableURI>/vmrest/restrictiontables/255bfdd2-5686-47f2-b40a-
  320c194521ba</RestrictionTableURI>
  <SequenceNumber>1</SequenceNumber>
  <ObjectId>2c2c9504-8fb4-44e3-983c-93eb4e20325c</ObjectId>
</RestrictionPattern>
```

```
Response Code: 200
```

## Creating a Restriction Pattern

The following is an example of the POST request that creates a restriction pattern:

```
POST https://<connectionserver>/
vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns
```

**Example 1:** The following is the example of the create request from the above \*POST\* request.

```
Request Body:
<RestrictionPattern>
  <NumberPattern>*</NumberPattern>
</RestrictionPattern>
```

```
Response Code: 201
```

### JSON Example

To create new restriction pattern, do the following:

```
Request URI:
POST https://<connectionserver>/
vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns
Accept: application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "NumberPattern": "*"
}
```

The following is the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
Response Code: 201
```

**Example 2:** The following is the example of the create request with specific restriction table Object ID

```
Request Body:
<RestrictionPattern>
  <NumberPattern>*#??</NumberPattern>

<RestrictionTableObjectId>38de2a1f-ca74-4be3-bb7c-d315df4c0fc5</RestrictionTableObjectId>
</RestrictionPattern>
```

```
Response Code: 201
```

## Updating a Restriction Pattern

The following is an example of the PUT request that allows you to update the restriction pattern:

**Example 1:** Change number pattern of restriction pattern

```
PUT https://<connection-
server>/vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns/<restrictionpatternob
jectid>
Request Body:
<RestrictionPattern>
  <NumberPattern>??999#</NumberPattern>
</RestrictionPattern>
```

```
Response Code: 204
```

### JSON Example

To change number pattern of restriction pattern, do the following:

```
Request URI:
PUT https://<connection-
server>/vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns/<restrictionpatternobject
id>
Accept: application/json
Content-Type: application/json
Connection: keep-alive
{
  "NumberPattern": "99??"
}
```

The following is the response from the above \*PUT\* request and the actual response will depend upon the information given by you:

```
Request URI: 204
```

**Example 2:** Change sequence of restriction pattern



- Note**
- To change the sequence of restriction patterns GET the object id of restriction patterns and arrange them in the required sequence.
  - To change the sequence of restriction patterns through API, you must provide all the restriction pattern object Id present in that restriction table

```
PUT https://<connectionserver>/
vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns
Request Body:
<RestrictionPatterns>
  <RestrictionPattern>
    <ObjectId>6564adb9-9090-42e0-81e6-04e132c4382c</ObjectId>
  </RestrictionPattern>
  <RestrictionPattern>
    <ObjectId>d9cd1525-462b-4eef-8629-0be9d0db2d18</ObjectId>
  </RestrictionPattern>
  <RestrictionPattern>
    <ObjectId>db0aed70-316b-47cb-8335-52fe34d3ca94</ObjectId>
  </RestrictionPattern>
  <RestrictionPattern>
    <ObjectId>46485dc6-bf31-4b31-be58-9eacbe718d02</ObjectId>
  </RestrictionPattern>
  <RestrictionPattern>
    <ObjectId>b7d51bb8-0059-4b56-aadd-6f9e2eaea624</ObjectId>
  </RestrictionPattern>
</RestrictionPatterns>
```

Request URI: 204



- Note** Default restriction pattern can neither be edited nor be deleted.



- Note** The entire restriction pattern Object Id is given including default restriction pattern. But sequence of default restriction pattern cannot be changed because default restriction pattern cannot be edited nor be deleted.

## Deleting a Restriction Pattern

**Example 1: Delete particular restriction pattern** The following is an example of the DELETE request that deletes a specific restriction table where you need to mention the restriction pattern object ID

```
DELETE https://<connection-
server>/vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns/<restrictionpatterno
bjectid>
```

Response Code: 204

### JSON Example

To delete particular restriction pattern, do the following:

```
DELETE https://<connectionserver>/
vmrest/restrictiontables/<restrictiontableobjectid>/restrictionpatterns/<restrictionpatternobject
id>
Accept: application/json
Connection: keep-alive
```

Response Code: 204



- Note**
- Default restriction pattern cannot be edited or deleted.
  - When restriction pattern is blocked then then calls matching to dial string will not be allowed.

## Explanation of Data Fields

The following chart lists all of the data fields:

Device Name	Data Type	Operation	Comments
Number Pattern	Varchar	Read/Write	The specific numbers or patterns of numbers (including external and long-distance access codes) that can be permitted or restricted. It uses digits 0 through 9 plus the following special characters: <ul style="list-style-type: none"> <li>• *Match zero or more digits</li> <li>• ? Match exactly one digit. Each '?' serves as a placeholder for one digit.</li> <li>• # Corresponds to the # key on the phone</li> <li>• + to call from one country to other country</li> </ul> For example, to screen out all phone numbers that start with 206 but are longer than 7 digits, enter 9206?????* for the pattern (and set "Blocked" == true). Maximum length can be 40.
Blocked	Boolean	Read/Write	A flag indicating whether Cisco Unity Connection permits the use of dial strings matching the pattern. <p>Values:</p> <ul style="list-style-type: none"> <li>• false: Permit use of dial strings matching the pattern</li> <li>• true: Do not permit use of dial strings matching the pattern</li> </ul> Default Value - true
ObjectId	String(36)	Read Only	Specifies the object ID of restriction patterns.
RestrictionTableObjectId	String(36)	Read Only	Specifies the restriction table object id to which restriction pattern belongs.
SequenceNumber	Integer(2)	Read Only	A sequential index for this restriction pattern within the parent restriction table, which specifies the order in which Cisco Unity Connection will apply each call pattern.

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Restriction Tables

## Restriction Tables

Administrator can use this API to create/update/delete/fetch the restriction table. Various attributes of restriction table can also be updated using this API.

### Listing the Restriction Tables

The following is an example of the GET request that lists all the restriction tables:

```
GET https://<connection-server>/vmrest/restrictiontables
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<RestrictionTables total="2">
  <RestrictionTable>
    <URI>/vmrest/restrictiontables/71be3f47-fcf4-463d-8e8a-1710a2180de6</URI>
    <ObjectId>71be3f47-fcf4-463d-8e8a-1710a2180de6</ObjectId>
    <CreationTime>2013-01-29T09:46:52Z</CreationTime>
    <DefaultBlocked>false</DefaultBlocked>
    <LocationObjectId>9f59d35f-104b-4875-9995-39925dd024c0</LocationObjectId>
    <LocationURI>/vmrest/locations/connectionlocations/9f59d35f-104b-4875-9995-39925dd024c0</LocationURI>
    <MaxDigits>40</MaxDigits>
    <MinDigits>1</MinDigits>
    <DisplayName>Default Transfer</DisplayName>
    <Undeletable>true</Undeletable>
    <RestrictionPatternsURI>/vmrest/restrictiontables/71be3f47-fcf4-463d-8e8a-1710a2180de6/restrictionpatterns</RestrictionPatternsURI>
  </RestrictionTable>
  <RestrictionTable>
    <URI>/vmrest/restrictiontables/a056f147-6469-4bb3-8314-5d0ff8011bad</URI>
    <ObjectId>a056f147-6469-4bb3-8314-5d0ff8011bad</ObjectId>
    <CreationTime>2013-01-29T09:46:52Z</CreationTime>
    <DefaultBlocked>false</DefaultBlocked>
    <LocationObjectId>9f59d35f-104b-4875-9995-39925dd024c0</LocationObjectId>
    <LocationURI>/vmrest/locations/connectionlocations/9f59d35f-104b-4875-9995-39925dd024c0</LocationURI>
    <MaxDigits>40</MaxDigits>
    <MinDigits>1</MinDigits>
    <DisplayName>Default Outdial</DisplayName>
    <Undeletable>true</Undeletable>
    <RestrictionPatternsURI>/vmrest/restrictiontables/a056f147-6469-4bb3-8314-5d0ff8011bad/restrictionpatterns</RestrictionPatternsURI>
  </RestrictionTable>
</RestrictionTables>
```

```
Response Code: 200
```

### JSON Example

To list all the restriction tables use the following command, do the following:



```
Request URI:
GET https://<connection-server>/vmrest/restrictiontables
Accept: application /json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
{
  "@total": "2",
  "RestrictionTable": [
    {
      "URI": "/vmrest/restrictiontables/4f01e5b1-b649-4f94-b55e-0c53d0e29c38",
      "ObjectId": "4f01e5b1-b649-4f94-b55e-0c53d0e29c38",
      "CreationTime": "2013-02-14T05:05:09Z",
      "DefaultBlocked": "false",
      "LocationObjectId": "bbf3e6ed-0278-479c-9a6e-2da8756eeb6f",
      "LocationURI": "/vmrest/locations/connectionlocations/bbf3e6ed-0278-479c-9a6e-2da8756eeb6f",
      "MaxDigits": "40",
      "MinDigits": "1",
      "DisplayName": "Default Transfer",
      "Undeletable": "true",
      "RestrictionPatternsURI": "/vmrest/restrictiontables/4f01e5b1-b649-4f94-b55e-0c53d0e29c38/restrictionpatterns"
    },
    {
      "URI": "/vmrest/restrictiontables/d66b1140-986a-40f1-a7d0-09714652d53f",
      "ObjectId": "d66b1140-986a-40f1-a7d0-09714652d53f",
      "CreationTime": "2013-02-14T05:05:09Z",
      "DefaultBlocked": "false",
      "LocationObjectId": "bbf3e6ed-0278-479c-9a6e-2da8756eeb6f",
      "LocationURI": "/vmrest/locations/connectionlocations/bbf3e6ed-0278-479c-9a6e-2da8756eeb6f",
      "MaxDigits": "40",
      "MinDigits": "1",
      "DisplayName": "Default Outdial",
      "Undeletable": "true",
      "RestrictionPatternsURI": "/vmrest/restrictiontables/d66b1140-986a-40f1-a7d0-09714652d53f/restrictionpatterns"
    }
  ]
}
```

```
Response Code: 200
```

## Viewing the Specific Restriction Table

**Example 1:** With valid object ID The following is an example of the GET request that lists the details of specific restriction table represented by the provided value of object ID:

```
GET https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<RestrictionTable>
  <URI>/vmrest/restrictiontables/71be3f47-fcf4-463d-8e8a-1710a2180de6</URI>
  <ObjectId>71be3f47-fcf4-463d-8e8a-1710a2180de6</ObjectId>
  <CreationTime>2013-01-29T09:46:52Z</CreationTime>
  <DefaultBlocked>>false</DefaultBlocked>
  <LocationObjectId>9f59d35f-104b-4875-9995-39925dd024c0</LocationObjectId>
  <LocationURI>/vmrest/locations/connectionlocations/9f59d35f-104b-4875-9995-
39925dd024c0</LocationURI>
  <MaxDigits>40</MaxDigits>
  <MinDigits>1</MinDigits>
  <DisplayName>Default Transfer</DisplayName>
  <Undeletable>>true</Undeletable>
  <RestrictionPatternsURI>/vmrest/restrictiontables/71be3f47-fcf4-463d-8e8a-
1710a2180de6/restrictionpatterns</RestrictionPatternsURI>
</RestrictionTable>
```

Response Code: 200

### JSON Example

To view specific restriction table, do the following:

```
Request URI:
GET https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
Accept: application/json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
{
  "URI": "/vmrest/restrictiontables/d66b1140-986a-40f1-a7d0-09714652d53f",
  "ObjectId": "d66b1140-986a-40f1-a7d0-09714652d53f",
  "CreationTime": "2013-02-14T05:05:09Z",
  "DefaultBlocked": "false",
  "LocationObjectId": "bbf3e6ed-0278-479c-9a6e-2da8756eeb6f",
  "LocationURI": "/vmrest/locations/connectionlocations/bbf3e6ed-0278-479c-9a6e-
2da8756eeb6f",
  "MaxDigits": "40",
  "MinDigits": "1",
  "DisplayName": "Default Outdial",
  "Undeletable": "true",
  "RestrictionPatternsURI": "/vmrest/restrictiontables/d66b1140-986a-40f1-a7d0-
09714652d53f/restrictionpatterns"
}
```

Response Code: 200

## Creating a Restriction Table

The following is an example of the POST request that creates a restriction table:

```
POST https://<connection-server>/vmrest/restrictiontables
```

**Example 1:** The following is the example of the create request from the above \*POST\* request.

```
Request Body:
<RestrictionTable>
  <DisplayName>Texoma 1</DisplayName>
</RestrictionTable>
```

```
Response Code: 201
```

### JSON Example

To create new restriction table, do the following:

```
POST https://<connection-server>/vmrest/restrictiontables
Accept: application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "DisplayName": "Texoma 1"
}
```

The following is the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
Response Code: 201
```

**Example 2:** The following is the example of the create request with maximum and minimum length

```
Request Body:
<RestrictionTable>
  <DisplayName>Texoma Restriction Table_7</DisplayName>
  <MinDigits>50</MinDigits>
  <MaxDigits>60</MaxDigits>
</RestrictionTable>
```

The following is the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
Response Code: 201
```

### JSON Example

To create restriction table with maximum and minimum length, do the following:

```
Request URI:
POST https://<connection-server>/vmrest/restrictiontables
Accept: application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "MaxDigits": "60",
  "MinDigits": "50",
  "DisplayName": "Texoma Restriction Table_7"
}
```

The following is the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
Response Code: 201
```

## Updating a Restriction Table Parameters

The following is an example of the PUT request that allows you to update the parameters of the restriction tables:

```
PUT https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
```

### Example 1: Update display name of restriction table

```
Request Body:
<RestrictionTable>
  <DisplayName>Texoma_123</DisplayName>
</RestrictionTable>
```

```
Response Code: 204
```

### JSON Example

To update display name of restriction table, do the following:

```
Request URI:
PUT https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
Accept: application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "DisplayName": "Texoma_123"
}
```

The following is the response from the above \*PUT\* request and the actual response will depend upon the information given by you:

```
Response Code: 204
```

### Example 2: Update maximum and minimum digits of restriction table

```
Request Body:
<RestrictionTable>
  <DisplayName>Texoma Restriction Table_7</DisplayName>
  <MinDigits>241</MinDigits> //Minimum digit range is 1-300, it should not exceed above
  300
  <MaxDigits>256</MaxDigits> //Maximum digit range is 1-300, it should not exceed above
  300
</RestrictionTable>
```

```
Response Code: 204
```

### Example 3: Update default blocked parameter

```
Request Body:
<RestrictionTable>
  <DefaultBlocked>true</DefaultBlocked>
</RestrictionTable>
```

```
Response Code: 204
```

## Deleting a Restriction Table

**Example 1:** Deleting a restriction table with a valid object id

The following is an example of the DELETE request that deletes a specific restriction table where you need to mention the object ID:

```
DELETE https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
```

```
Response Code: 204
```

### JSON Example

To delete restriction table with a valid object id, do the following:

```
DELETE https://<connection-server>/vmrest/restrictiontables/<restrictiontableobjectid>
Accept: application/json
Connection: keep-alive
```

```
Response Code: 204
```



**Note** Default restriction table cannot be edited or deleted.

## Explanation of Data Fields

The following chart lists all of the data fields:

Device Name	Data Type	Operation	Comments
ObjectId	String (36)	Read only	Specifies an unique and system generated ID of a particular restriction table object.
DisplayName	String (64)	Read/Write	Specifies unique display name of restriction table in order to display entries in the administrative console.
MaxDigits	Integer	Read/Write	Specifies the maximum number of digits in a dial string (including access codes) allowed by Cisco Unity Connection. Only dial strings that contain a number of digits fewer than or equal to the number of digits specified in this column are checked against the restriction table. Dial strings that contain more than the numbers of digits specified in this column are not permitted. For example, if the local calls in your area are seven digits long, and you want to prevent subscribers from using long distance phone numbers, enter 8 in this column.(Range 1-300).

Device Name	Data Type	Operation	Comments
MinDigits	Integer	Read/Write	Specifies the minimum number of digits in a dial string (including access codes) allowed by Cisco Unity Connection. Only dial strings that contain a number of digits greater than or equal to the number of digits specified in this column are checked against the restriction table. Dial strings that contain fewer than the number of digits specified in this column are not permitted. For example, to prohibit subscribers from using four-digit numbers, enter a value of 5 in this column. (Range 1-300)
DefaultBlocked	Boolean	Read/Write	A flag indicating whether Cisco Unity Connection permits the phone number in the case where a phone number does not match any call patterns in this restriction table. This condition should never occur. By default, all restriction tables should have * as the call pattern in the last dial string of the table and an administrator should not be allowed to modify this call pattern setting.  Possible values: <ul style="list-style-type: none"> <li>• false: Blocked</li> <li>• true: Not Blocked</li> </ul> Default value: false
CreationTime	DateTime	Read only	Creation date and time of restriction table.
Undeletable	Boolean	Read/Write	A flag indicating whether this restriction table can be deleted via an administrative application such as Cisco Unity Connection Administration. It is used to prevent deletion of factory defaults.  Default Value - false

## Cisco Unity Connection Provisioning Interface (CUPI) API -- License Dump APIs

### License Dump APIs

Administrator can use this API to fetch license usage information for 6 licensing parameters/features. These 6 license parameters are:

License Parameter	License Tag	Feature	Description
CUC_BasicMessaging	LicSubscribersMax	Total number of voicemail users	<p>Specifies the maximum number of voice mail users configured in Cisco Unity Connection.</p> <p><b>Note</b> All the users which are the time of installation, example, "operator" and "undeliverablemessage" are removed from this. Also all the tenant oper count is not included in for this tag.</p>
CUC_SpeechView	LicSTTSubscribersMax	Total number of speech view users using the standard transcription services	Specifies the maximum number of speech view users using the standard transcription services configured in Cisco Unity Connection.
CUC_SpeechViewPro	LicSTTProSubscribersMax	Total number of speech view users using the professional transcription services	Specifies the maximum number of Speech view users using the professional transcription services configured in Cisco Unity Connection
CUC_SpeechConnectPort	LicRealspeakSessionsMax	Total number of speech connect ports	Specifies the maximum number of Speech Connect calls configured in Cisco Unity Connection
CUC_EnhancedMessaging	LicSrvCueSubscribersMax	Total number of enhanced messaging users	Specifies the maximum number of Connection SRSV users configured on Cisco Unity Connection. The Connection SRSV users are reflected under this tag only when the branch is active. This feature is not applicable for Tenant Partitioning system.
CUC_SpeechConnectGuestUser	LicContactsMax	Total number of Contacts.	Specifies the maximum number of local contacts, along with VPIM contacts created from Non Unity Connection End Point.

## Listing the License Status Count

The following is an example of the GET request that lists all the license status count:

```
GET https://<connection-server>/vmrest/licensestatuscounts
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<LicenseStatusCounts total="6">
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicSubscribersMax</URI>
    <TagName>LicSubscribersMax</TagName>
    <ObjectId>bc1f1f5d-f628-4bca-b0c9-f225d07d677f</ObjectId>
    <Count>3</Count>
    <featureName>CUC_BasicMessaging</featureName>
    <description>Total Number of Voicemail users</description>
  </LicenseStatusCount>
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicSrsvCuceSubscribersMax</URI>
    <TagName>LicSrsvCuceSubscribersMax</TagName>
    <ObjectId>99541ef5-4602-424c-84e6-6c9fe8b4044b</ObjectId>
    <Count>0</Count>
    <featureName>CUC_EnhancedMessaging</featureName>
    <description>Total Number of Enhanced Messaging Users</description>
  </LicenseStatusCount>
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicSTTSubscribersMax</URI>
    <TagName>LicSTTSubscribersMax</TagName>
    <ObjectId>dfbaf0f8-38b0-4e14-9f62-9b1e59cc2426</ObjectId>
    <Count>0</Count>
    <featureName>CUC_SpeechView</featureName>
    <description>Speechview Standard users</description>
  </LicenseStatusCount>
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicSTTProSubscribersMax</URI>
    <TagName>LicSTTProSubscribersMax</TagName>
    <ObjectId>d1f15aa9-2482-4a80-93e8-58fb402e31dd</ObjectId>
    <Count>3</Count>
    <featureName>CUC_SpeechViewPro</featureName>
    <description>Speechview Professional Users</description>
  </LicenseStatusCount>
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicRealspeakSessionsMax</URI>
    <TagName>LicRealspeakSessionsMax</TagName>
    <ObjectId>2b1a7da9-6b13-4302-8efd-91db94b2c241</ObjectId>
    <Count>3</Count>
    <featureName>CUC_SpeechConnectPort</featureName>
    <description>Total Number of speech connect sessions</description>
  </LicenseStatusCount>
  <LicenseStatusCount>
    <URI>/vmrest/licensestatuscounts/LicContactsMax</URI>
    <TagName>LicContactsMax</TagName>
    <ObjectId>417f531f-077b-4d64-8f5b-749a56623f400</ObjectId>
    <Count>1</Count>
    <featureName>CUC_SpeechConnectGuestUser</featureName>
    <description>Total Number of Contacts</description>
  </LicenseStatusCount></LicenseStatusCounts>
</pre>


Response Code: 200


```

### JSON Example



To list all the license status count use the following command:

```
Request URI:
GET https://<connection-server>/vmrest/licensestatuscounts
Accept: application /json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
{
  "@total": "2"
  "LicenseStatusCount": [
    {
      "URI": "/vmrest/licensestatuscounts/LicSubscribersMax"
      "TagName": "LicSubscribersMax"
      "ObjectId": "bc1f1f5d-f628-4bca-b0c9-f225d07d677f"
      "Count": "3"
      "featureName": "CUC_BasicMessaging"
      "description": "Total Number of Voicemail users"
    }
    {
      "URI": "/vmrest/licensestatuscounts/LicSrsvCuceSubscribersMax"
      "TagName": "LicSrsvCuceSubscribersMax"
      "ObjectId": "99541ef5-4602-424c-84e6-6c9fe8b4044b"
      "Count": "0"
      "featureName": "CUC_EnhancedMessaging"
      "description": "Total Number of Enhanced Messaging Users"
    }
  ]
}
```

```
Response Code: 200
```

## Viewing the Specific License Status Count

```
Request URI:
GET https://<connection-server>/vmrest/licensestatuscounts/<TagName>
```

Tag names can be:

- LicSubscribersMax
- LicSrsvCuceSubscribersMax
- LicSTTSubscribersMax
- LicSTTProSubscribersMax
- LicRealspeakSessionsMaxSample Request

The following is an example of the GET request that lists the details of specific license status count represented by the provided value of object ID:

```
GET https://<connection-server>/vmrest/licensestatuscounts/LicSubscribersMax
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<LicenseStatusCount>
  <URI>/vmrest/licensestatuscounts/LicSubscribersMax</URI>
  <TagName>LicSubscribersMax</TagName>
  <ObjectId>bc1f1f5d-f628-4bca-b0c9-f225d07d677f</ObjectId>
  <Count>3</Count>
  <featureName>CUC_BasicMessaging</featureName>
  <description>Total Number of Voicemail users</description>
</LicenseStatusCount>
```

Response Code: 200

### JSON Example

To view specific license status count, do the following:

```
GET https://<connection-server>/vmrest/licensestatuscounts/<TagName>
Request URI:
GET https://<connection-server>/vmrest/licensestatuscounts/LicSubscribersMax
Accept: application /json
Connection: keep-alive
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
{
  "URI": "/vmrest/licensestatuscounts/LicSubscribersMax"
  "TagName": "LicSubscribersMax"
  "ObjectId": "bc1f1f5d-f628-4bca-b0c9-f225d07d677f"
  "Count": "3"
  "featureName": "CUC_BasicMessaging"
  "description": "Total Number of Voicemail users"
}
```

Response Code: 200

## Explanation of Data Fields

The following chart lists all of the data fields:

Parameter	Operation	Data Type	Comment
URI	Read Only	String	Specifies the URI of the API.
TagName	Read Only	String	Specifies the license tag of the feature. Tag names can be: <ul style="list-style-type: none"> <li>• LicSubscribersMax</li> <li>• LicSrsvCuceSubscribersMax</li> <li>• LicSTTSubscribersMax</li> <li>• LicSTTProSubscribersMax</li> <li>• LicRealspeakSessionsMax</li> </ul>
ObjectId	Read Only	String (36)	Identifier for a LicenseStatus object.

Parameter	Operation	Data Type	Comment
Count	Read Only	String	The current usage of the feature in the system. It is a number specifying the current usage count or if the feature is in use.
FeatureName	Read Only	String	The name of the feature.
Description	Read Only	String	The description of the feature

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Smart Licensing

## Overview

With Unity Connection 12.0(1) and later, a new simple and enhanced way of licensing, Cisco Smart Software Licensing is introduced to use various licensed feature of Unity Connection. Using Cisco Smart Software Licensing, you can manage all the licenses associated with an organization through a single interface, which is Cisco Smart Software Manager (CSSM) or Cisco Smart Software Manager satellite. Cisco Smart Software Licensing provides the visibility of your licenses ownership and consumption. Unity Connection must be registered with the Cisco Smart Software Manager (CSSM) or Cisco Smart Software Manager satellite to use various licensed feature.

Unity Connection remains in the Evaluation Mode until it registers with the Cisco Smart Software Manager (CSSM) or Cisco Smart Software Manager satellite.

For information on Unity Connection licenses, see the "[Managing Licenses](#)" chapter of the Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x, available at [https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/connection/12x/install\\_upgrade/guide/b\\_12xcuciumg.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html).

With release 12.5(1) and later, Unity Connection provides the APIs to perform the various operations of Cisco Smart Software Licensing.

## Transport Settings

To view and manage the licenses, Unity Connection must communicate with the Cisco Smart Software Manager (CSSM) or Cisco Smart Software Manager satellite.

Following are the options to deploy the Cisco Smart Software Licensing in Unity Connection,

- **Direct Cloud Access:** In this option, Unity Connection can directly communicate with CSSM and transfer the usage information over internet. No additional components are required.
- **Mediated Access through an On-Premises Collector:** In this option, Unity Connection communicates with on-prem version of CSSM called Cisco Smart Software Manager satellite. Periodically satellite communicates with CSSM using Cisco network and exchange of license information will be performed.
- **Direct Cloud Access through an HTTPs Proxy:** In this option, Unity Connection directly transfers the usage information to CSSM over internet through proxy server.



**Note** Cisco Unity Connection Release 12.5(1) Service Update 4 and later, allows you to authenticate the proxy server for more secure communication with CSSM. To authenticate the proxy server below fields are introduced in Transport Settings API.

- isProxyAuthReq
- httpProxyUser
- httpProxyPasswd

For more information on transport settings, see [Deployment Options](#) section of the Managing Licenses chapter of the *Install, Upgrade, and Maintenance Guide for Cisco Unity Connection Release 12.x* available at [https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/connection/12x/install\\_upgrade/guide/b\\_12xcuciumg.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html)

## Listing Transport Settings

The following is an example of the GET request that displays the configured transport setting:

```
GET https://<connection-server>/vmrest/smartlicense/transportsettings
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<TransportSetting>
<transportMode>2</transportMode>
<transportUrl>https://tools.cisco.com/its/service/odce/services/DDCEService</transportUrl>
<httpHost></httpHost>
<httpPort>0</httpPort>
<isPrivacyEnabled>>false</isPrivacyEnabled>
<isProxyAuthReq>>true</isProxyAuthReq>
<httpProxyUser>James</httpProxyUser></TransportSetting>
```

```
Response Code: 200
```

### JSON Example

```
{
  "transportMode": "2",
  "transportUrl":
"https://tools.cisco.com/its/service/odce/services/DDCEService</transportUrl",
  "httpHost": "",
  "httpPort": "0"
  "isPrivacyEnabled": "false"
  "isProxyAuthReq": "true"
  "httpProxyUser" : "James" }
```

```
Response Code: 200
```

## Modifying Transport Settings

The following is an example of the PUT request that modifies the Transport Settings:

```
PUT https://<connection-server>/vmrest/smartlicense/transportsettings
```

The following is the response from the above \*PUT\* request and the actual response will depend upon the information given by you:

```
<TransportSetting>
<transportMode>0</transportMode>
<transportUrl>https://tools.cisco.com/its/service/odce/services/DDCEService</transportUrl>
<httpHost></httpHost>
<httpPort>0</httpPort>
<isPrivacyEnabled>>false</isPrivacyEnabled>
<isProxyAuthReq>1</isProxyAuthReq>
<httpProxyUser>James</httpProxyUser>
<httpProxyPasswd>****</httpProxyPasswd></TransportSetting>
```

Response Code: 200

### JSON Example

```
{
  "transportMode": "0"
  "transportUrl": "https://tools.cisco.com/its/service/odce/services/DDCEService"
  "httpHost": ""
  "httpPort": "0"
  "isPrivacyEnabled" : "false"
  "isProxyAuthReq" : "true"
  "httpProxyUser" : "James"
  "httpProxyPasswd" : "****"}

```

Response Code: 200



**Note** If the product is registered with CSSM or satellite, you must deregister the product before changing the transport settings.

## Registering the Unity Connection

Using this API, you can register the product with CSSM or satellite. For registration, you need a registration token from CSSM or satellite.

For information on how to create a token on CSSM or satellite, see "[Configuring Cisco Smart Software Licensing in Unity Connection](#)" section of "[Managing Licenses](#)" chapter of the Install, Upgrade, and Maintenance Guide for Cisco Unity Connection, Release 12.x, available at [https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/connection/12x/install\\_upgrade/guide/b\\_12xcuciumg.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/install_upgrade/guide/b_12xcuciumg.html).

Following is the PUT request to register the product with CSSM or satellite :

```
PUT https://<connection-server>/vmrest/smartlicense/register
```

```
<RegisterDetails>
<token>NDJiMjI0YTAtMjc0MCM00.....1JakvZMU5JWGHLOX1P%0AMmVxMD0%3D%0A</token>
<force>>false</force>
</RegisterDetails>
```

```
Response Code: 200
```

**JSON Example**

```
{
  "token" : "NDJiMjI0YTAtMjc0MC00.....s1RVdnR2c2VEExDRXEvGtUWWVh%0ASTRNdz0%3D%0B"
  "force" : "false"
}
```

```
Response Code: 200
```

## Reregistering the Unity Connection

Using this API, you can reregister the product with CSSM or satellite. For reregistration, you need a registration token from CSSM or satellite.

Following is the PUT request to reregister the product with CSSM or satellite:

```
PUT https://<connection-server>/vmrest/smartlicense/register
```

```
<RegisterDetails>
<token>NDJiMjI0YTAtMjc0MC00.....1JakVZMU5JWGhLOXlP%0AMmVxMD0%3D%0A</token>
<force>>true</force>
</RegisterDetails>
```

```
Response Code: 200
```

**JSON Example**

```
{
  "token" : "NDJiMjI0YTAtMjc0MC00.....1JakVZMU5JWGhLOXlP%0AMmVxMD0%3D%0A"
  "force" : "true"
}
```

```
Response Code: 200
```

## Deregistering the Unity Connection

Using this API, you can deregister the product from CSSM or satellite. All license entitlements used for the product are released back to its virtual account.

The following is an example of the PUT request that deregisters the product from CSSM:

```
PUT https://<connection-server>/vmrest/smartlicense/deregister
```

```
Response Code: 200
```

## Renew Authorization of the Unity Connection

Using this API, you can renew the license authorization for the product with CSSM or satellite. However, the licenses are automatically authorized in every 30 days.

Following is the PUT request to renew the license authorization for the product:

```
PUT https://<connection-server>/vmrest/smartlicense/renewAuth
```

```
Response Code: 200
```

## Renew Registration of the Unity Connection

Using this API, you can renew the registration of the product with CSSM or satellite. However, the registration of the product is automatically renewed in every six month.

Following is the PUT request to renew the registration of the product:

```
PUT https://<connection-server>/vmrest/smartlicense/renewID
```

```
Response Code: 200
```

## Listing Licensing Details

This API is used for listing all the licensing information of the product.

Following is the GET request to listing all the licensing details:

```
GET https://<connection-server>/vmrest/smartlicense/licensedetails
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

## Listing Licensing Details

```

<LicenseDetails>
<SmartLicensing>Enabled</SmartLicensing>
<Register>
<Status>Registered</Status>
<SmartAccount>BU Production Test</SmartAccount>
<VirtualAccount>Cisco_VA</VirtualAccount>
<LastRenewalAttempt>SUCCEDED 2017-11-22 13:17:19.0</LastRenewalAttempt>
<NextRenewalAttempt>2018-05-23 13:17:19.0</NextRenewalAttempt>
<LastRegistrationSuccessTime>2017-11-24 13:17:19.0</LastRegistrationSuccessTime>
<ProductUDI>pid=Cisco Unity Connection sn=cbd7561030494c60af97aabbcbca</ProductUDI>
<ExportControlFunctionality>Allowed</ExportControlFunctionality>
</Register>
<Authorization>
<Status>Authorized</Status>
<LastCommunicationAttempt>SUCCEDED 2017-11-27 13:17:19.0</LastCommunicationAttempt>
<NextCommunicationAttempt>2017-12-27 13:17:19.0</NextCommunicationAttempt>
<EvaluationPeriodRemaining>87:0:36</EvaluationPeriodRemaining>
</Authorization>
<LicenseUsage>
<EntitlementTag>
<Id>regid.2020-06.com.cisco.CUC_BasicMessaging,14_74b6bdc1-b3e1-45dc-9468-6412b12dfef2</Id>
<Count>1</Count>
<Status>Evals</Status>
</EntitlementTag>
<EntitlementTag>
<Id>regid.2020-06.com.cisco.CUC_EnhancedMessaging,14_5e05b515-5be5-429e-b301-5de5de8d2c09</Id>
<Count>0</Count>
<Status>Init</Status>
</EntitlementTag>
<EntitlementTag>
<Id>regid.2020-11.com.cisco.CUC_SpeechViewPro,14_9de30ccc-1671-4c32-9f57-f63c3f66af84</Id>
<Count>0</Count>
<Status>Init</Status>
</EntitlementTag>
<EntitlementTag>
<Id>regid.2020-11.com.cisco.CUC_SpeechView,14_5a0e3202-2743-4401-a117-1e9c239acbbf</Id>
<Count>10</Count>
<Status>InCompliance</Status>
</EntitlementTag>
<EntitlementTag>
<Id>regid.2020-06.com.cisco.CUC_SpeechConnectPort,14_d0201ba3-0b71-403d-9b74-5cd643dd662b</Id>
<Count>0</Count>
<Status>Init</Status>
</EntitlementTag>
<EntitlementTag>
<Id>regid.2020-06.com.cisco.CUC_SpeechConnectGuestUser,14_d49c924c-e587-4a02-8376-7c48bc7ac3ba</Id>
<Count>0</Count>
<Status>Init</Status>
</EntitlementTag>
</LicenseUsage>
</LicenseDetails>

```

Response Code: 200

**JSON Example**



```

{
  "SmartLicensing": "Enabled",
  "Register": {
    "Status": "Reservation In Progress",
    "SmartAccount": "BU Production Test",
    "VirtualAccount": "Cisco_VA",
    "LastRenewalAttempt": "SUCCEEDED 2017-11-22 13:17:19.0",
    "NextRenewalAttempt": " 2018-05-23 13:17:19.0",
    "LastRegistrationSuccessTime": "2017-11-24 13:17:19.0",
    "ProductUDI": "pid=Cisco Unity Connection sn=cbd7561030494c60af97aabbcb",
    "ExportControlFunctionality": "Allowed"
  },
  "Authorization": {
    "Status": "Evaluation",
    "LastCommunicationAttempt": "SUCCEEDED 2017-11-27 13:17:19.0",
    "NextCommunicationAttempt": " 2017-12-27 13:17:19.0",
    "EvaluationPeriodRemaining": "87:0:36"
  },
  "LicenseUsage": {
    "EntitlementTag": [
      {
        "Id": "regid.2020-06.com.cisco.CUC_BasicMessaging,14_74b6bdc1-b3e1-45dc-9468-6412b12fdef2",
        "Count": "1",
        "Status": "Eval"
      },
      {
        "Id":
        "regid.2020-06.com.cisco.CUC_EnhancedMessaging,14_5e05b515-5be5-429e-b301-5de5de8d2c09",
        "Count": "0",
        "Status": "Init"
      },
      {
        "Id":
        "regid.2020-11.com.cisco.CUC_SpeechViewPro,14_9de30ccc-1671-4c32-9f57-f63c3f66af840Init",
        "Count": "0",
        "Status": "Init"
      },
      {
        "Id": "regid.2020-11.com.cisco.CUC_SpeechView,14_5a0e3202-2743-4401-a117-1e9c239acbbf",
        "Count": "0",
        "Status": "Init"
      },
      {
        "Id":
        "regid.2020-06.com.cisco.CUC_SpeechConnectPort,14_d0201ba3-0b71-403d-9b74-5cd643dd662b",
        "Count": "0",
        "Status": "Init"
      }
    ]
  }
}

```

Response Code: 200

Cisco Unity Connection provides different Entitlement Tag based on the supported deployment modes of Unity Connection.

Cisco Unity Connection Release 12.5(1) Service Update 4 and later, administrator provide Speech View functionality with **Hosted Collaboration Services** (HCS) deployment mode. To use SpeechView feature in HCS mode, you must have **HCS SpeechView Standard User Licenses** with the users. **HCUC\_SpeechView** entitlement tag is added to support the functionality.



**Note** In HCS mode, only Standard SpeechView Transcription Service is supported.

For SpeechView configuration refer chapter "[SpeechView](#)" of *System Administration Guide Cisco Unity Connection Release 12* available at link [https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/connection/12x/administration/guide/b\\_12xcucsag.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/12x/administration/guide/b_12xcucsag.html).

Following table describes the deployment modes with corresponding Entitlement Tags:

Deployment Mode	Description	Entitlement Tag
Enterprise	This is the default mode of Unity Connection	<ul style="list-style-type: none"> <li>• CUC_BasicMessaging</li> <li>• CUC_EnhancedMessaging</li> <li>• CUC_SpeechView</li> <li>• CUC_SpeechViewPro</li> <li>• CUC_SpeechConnectPort</li> <li>• CUC_SpeechConnectGuestUser</li> </ul>
HCS	This mode comprises Hosted Collaboration Services	<ul style="list-style-type: none"> <li>• HCUC_BasicMessaging</li> <li>• HCUC_EnhancedMessaging</li> <li>• HCUC_StandardMessaging</li> <li>• HCUC_SpeechConnectPort</li> <li>• HCUC_SpeechView</li> </ul>
HCS-LE	This mode comprises Hosted Collaboration Services -Large Enterprise	<ul style="list-style-type: none"> <li>• HLECUC_BasicMessaging</li> <li>• HLECUC_EnhancedMessaging</li> <li>• HLECUC_StandardMessaging</li> <li>• HLECUC_SpeechConnectPort</li> </ul>



**Note** To get the latest licensing details of a product, you must renew the authorization of the product. For more information, see "[Renew Authorization of the Unity Connection](#)" section.

## Getting License Data for STT

Using this API, you can acquire the SpeechView license data (Certificates and Voucher code) locally to the connection server from CSSM or satellite.

The following is an example of the PUT request that acquires the Speech View license data from CSSM or satellite:

```
PUT https://<connection-server>/vmrest/smartlicense/configurenuancecerts
```

```
Response Code: 200
```

## Explanation of Data Fields

The following chart lists all of the data fields:

Parameter	Data Type	Operations	Descriptions
transportMode	Integer	Read\Write	Specifies the different transport settings option through which Unity Connection communicates with CSSM. Possible values are: <ul style="list-style-type: none"> <li>• 0-Direct option.</li> <li>• 1-Mediated Access through an On-Premises Collector or satellite</li> <li>• 2-HTTP/HTTPS Proxy</li> </ul> Default-0
transportUrl	String	Read\Write	Specifies the URL through which Unity Connection communicates with CSSM. Default URL: <a href="https://tools.cisco.com/its/service/oddce/services/DDCService">https://tools.cisco.com/its/service/oddce/services/DDCService</a> <b>Note</b> This field is applicable when transportMode is set to Direct or satellite .
httpHost	String	Read\Write	Specifies the IP address or Hostname of the proxy server. <b>Note</b> This field is applicable when transportMode is set to HTTP/HTTPS Proxy.
httpPort	Integer	Read\Write	Specifies the port through which Unity Connection connects to the proxy server. <ul style="list-style-type: none"> <li>• Range-1 to 65535</li> </ul> Default-0 <b>Note</b> This field is applicable when transportMode is set to HTTP/HTTPS Proxy.

Parameter	Data Type	Operations	Descriptions
isProxyAuthReq <i>(Applicable from Unity Connection 12.5SU4 and later releases)</i>	Boolean	Read\Write	Specifies option to enter the details of password protected proxy server for more secure communication. Possible values are: <ul style="list-style-type: none"> <li>• 0- false.</li> <li>• 1- true. It is mandatory to provide username and password details of proxy server.</li> </ul> Default -0
httpProxyUser <i>(Applicable from Unity Connection 12.5SU4 and later releases)</i>	String	Read\Write	Specifies the proxy server username that is required to connect to Unity Connection through more secure channel.
httpProxyPasswd <i>(Applicable from Unity Connection 12.5SU4 and later releases)</i>	String	Read\Write	Specifies the proxy server password that is required to connect to Unity Connection through more secure channel.
SmartAccount	String	Read Only	Smart Account is a simple and organized way to manage the product licenses and entitlements. Using this account, you can register, view, and manage your Cisco Software Licenses across your organization.
VirtualAccount	String	Read Only	Virtual Accounts are the sub accounts within the Smart Accounts. Licenses and Product instances can be distributed across virtual accounts.
token	String	Read\Write	Token is required for registering the product with CSSM or satellite.
force	String	Read\Write	This flag is enable when you reregister the product with CSSM or satellite. The possible values are: <ul style="list-style-type: none"> <li>• False-For registering the product.</li> <li>• True-For reregistering the product.</li> </ul>
LastRenewalAttempt	String	Read Only	Specifies the status, date and time on which, the product is last renewed the registration with CSSM or satellite. Possible values are: <ul style="list-style-type: none"> <li>• SUCCEEDED-When the product successfully renews the registration with CSSM.</li> <li>• FAILED-When renew registration of the product with CSSM is failed.</li> <li>• Not Applicable-When the product is not registered once with CSSM or satellite.</li> </ul>

Parameter	Data Type	Operations	Descriptions
NextRenewalAttempt	Date\Time	Read Only	Specifies the date and time on which, the product is required to renew the registration with CSSM.
LastRegistrationSuccessTime	Date\Time	Read Only	Specifies the date and time on which, the product is successfully registered with CSSM or satellite.
LastCommunicationAttempt	String	Read Only	Specifies the status, date and time on which, the product last communicated with CSSM or satellite. Possible values are: <ul style="list-style-type: none"> <li>• SUCCEEDED-When the product is successfully communicated with CSSM or satellite.</li> <li>• FAILED-When the product is failed to communicate with CSSM or satellite.</li> <li>• Not Applicable-When the product is not registered once with CSSM or satellite.</li> </ul>
NextCommunicationAttempt	Date\Time	Read Only	Specifies the date and time on which, the product is required to communicate with CSSM or satellite.
ExportControlFunctionality	String	Read Only	Specifies the Export Controlled Functionality is enabled for the product or not at the time of token creation on the CSSM or satellite. Possible values are: <ul style="list-style-type: none"> <li>• Allowed-When Export Controlled Functionality is enabled for the product.</li> <li>• Not Allowed-When the Product is in Evaluation Mode or Export Controlled Functionality is not enabled for the product..</li> </ul>
Id	String	Read Only	Specifies the License Parameters.
Count	Integer	Read Only	Specifies the number of licenses that are used

## Cisco Unity Connection Provisioning Interface (CUPI) API -- Custom Subject Line

### Listing Subject Line Formats

This page contains information on how to use the API to list and update subject line formats and its parameters. Subject line format enables the user to configure the subject lines of the notification emails for Message Notifications, Missed Call Notifications, and Scheduled Summary Notifications.



**Note** POST and DELETE APIs are not supported.

The following is an example of the GET request that lists all subject line formats:

```
GET https://<connection_server>/vmrest/subjectlineformats
```

The following is an example of the GET request to list the subject line for a specific message type:

```
https://<connection_server>/vmrest/subjectlineformats?query=(MessageType is 2)
```



**Note** Message type is 1 for voice messages and 2 for notifications.

The following is an example of the response from the above GET requests and the actual response will depend upon the information given by you:

```
<SubjectLineFormats total="7">
  <SubjectLineFormat>
    <URI>/vmrest/subjectlineformats/7048d455-374b-4b81-bf0b-ebb70239b441</URI>
    <ObjectId>7048d455-374b-4b81-bf0b-ebb70239b441</ObjectId>
    <LanguageCode>1033</LanguageCode>
    <FactoryDefaultSubject>Message notification</FactoryDefaultSubject>
    <Subject>Message notification</Subject>
    <SubjectType>7</SubjectType>
    <MessageType>2</MessageType>

    <CustomSubjectParameterURI>/vmrest/subjectlineparameters?query=(MessageType%20is%202)</CustomSubjectParameterURI>

  </SubjectLineFormat>
  ...
</SubjectLineFormats>
```

```
Response code: 200 OK
```

## Listing a particular subject line format

The following is an example of the GET request that lists a particular subject line format:

```
GET https://<connection_server>/vmrest/subjectlineformats/<objectID>
```

The following is an example of the response from the above GET request and the actual response will depend upon the information given by you:

```
<SubjectLineFormat>
  <URI>/vmrest/subjectlineformats/75b13d88-7313-491a-b239-dff1f1372379</URI>
  <ObjectId>75b13d88-7313-491a-b239-dff1f1372379</ObjectId>
  <LanguageCode>1033</LanguageCode>
  <FactoryDefaultSubject>%D% %U% %P% Message from %NAME%
(%CALLERID%)</FactoryDefaultSubject>
  <Subject>%D% %U% %P% Message from %NAME% (%CALLERID%)</Subject>
  <SubjectType>1</SubjectType>
  <MessageType>1</MessageType>
  <CustomSubjectParameterURI>/vmrest/subjectlineparameters</CustomSubjectParameterURI>
</SubjectLineFormat>
```

Response code: 200 OK

## Modifying Subject Line Formats

The following is an example of the PUT request that modifies subject field in the subject line formats:

```
PUT https://<connection_server>/vmrest/subjectlineformats/<objectID>
```

The following is the input for above PUT request.

```
<SubjectLineFormat>
  <Subject>%D% %U% %P% Message from %NAME% 123(%CALLERID%)</Subject>
</SubjectLineFormat>
```

The following is the response of above PUT request.

Response code: 204 OK

## Listing Subject Line Parameters

The following is an example of the GET request that lists all subject line parameters:

```
GET https://<connection_server>/vmrest/subject-line-parameters
```

The following is an example of the response from the above GET request.

```
<SubjectLineParameters total="11">
  <SubjectLineParameter>
    <URI>/vmrest/subject-line-parameters/462d2eaa-8109-470e-982e-00c2c99e8770</URI>
    <ObjectId>462d2eaa-8109-470e-982e-00c2c99e8770</ObjectId>
    <LanguageCode>1033</LanguageCode>
    <Parameter>Unknown caller ID1</Parameter>
    <FactoryDefaultParameter>Unknown caller ID</FactoryDefaultParameter>
    <ParameterType>1</ParameterType>
    <MessageType>1</MessageType>
  </SubjectLineParameter>
  ....
</SubjectLineParameters>
```

Response code: 200 OK

## Listing a particular Subject Line Parameter

The following is an example of the GET request that list a particular subject line parameter:

```
GET https://<connection_server>/vmrest/subject-line-parameters/<objectID>
```

The following is an example of the GET request to list a particular subject line parameter for a specific message type:

```
https://<connection_server>/vmrest/subject-line-parameters?query=(MessageType is 2)
```



**Note** The Message Type is 1 for voice messages and 2 for notifications.

The following is an example of the response from the above GET request.

```
<SubjectLineParameter>
  <URI>/vmrest/subject-line-parameters/462d2eaa-8109-470e-982e-00c2c99e8770</URI>
  <ObjectId>462d2eaa-8109-470e-982e-00c2c99e8770</ObjectId>
  <LanguageCode>1033</LanguageCode>
  <Parameter>Unknown caller ID1</Parameter>
  <FactoryDefaultParameter>Unknown caller ID</FactoryDefaultParameter>
  <ParameterType>1</ParameterType>
  <MessageType>1</MessageType>
</SubjectLineParameter>
```

```
Response code: 200 OK
```

## Modifying Subject Line Parameter

The following is an example of the PUT request that modifies the subject line parameters:

```
PUT https://<connection_server>/vmrest/subject-line-parameters/<objectID>
```

The following is the response of above PUT request.

```
<SubjectLineParameter>
  <Parameter>Unknown caller ID1</Parameter>
</SubjectLineParameter>
```

```
Response code: 204 OK
```

## Explanation of Data Fields

For Subject Type, Message Type and Parameter Type

Field Name	Read/Write	Type	Explanation/Comments
ObjectId	Read	String	Object ID of the entity
LanguageCode	Read	Integer	Indicates the language code



Field Name	Read/Write	Type	Explanation/Comments
ParameterType	Read	Integer	1-%CALLERID% of the message sender 2-%CALLEDID% for the number dialed 3-%NAME%, display name of the sender 4-%EXTENSION%, the extension of the sender 5-%U%, defined value for urgent messages 6-%P%, defined value for private messages 7-%S%, defined value for secure messages 8-%D%, defined value for dispatch messages 9- %TIMESTAMP%, time at which the voice message is received
MessageType	Read	Integer	1- for Voice Messages 2-For Notifications
SubjectType	Read	Integer	0-Ignore, this is for invalid entry 1-Outside Caller Messages, to format string of subject line for Outside Caller Messages 2-User to User Messages, to format string of subject line for User to User Messages 3-Interview Handler Messages, to format string of subject line for Interview Handler Messages 4-Live Record Messages, to format string of subject line for Live Record Messages 5-Message Notification, to format string of subject line for Message Notification 6-Missed Call Notification, to format string of subject line for Missed Call Notification 7-Scheduled Summary Notification, to format string of subject line for Scheduled Notification
Parameter	Write	String	Customizes the Parameter Text
Subject	Write	String	Customizes the SubjectLine text
FactoryDefaultParameter	Read	String	System default value for subject line parameter
CustomSubjectParameterURI	Read	String	URI to get the custom subject parameters details
FactoryDefaultSubject	Read	String	System default value for Subject

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Language Map

## About Language Map

This page contains information on how to use the API to display the language map, which contains mappings between language codes and languages. This is a list of languages available for install on a Cisco Unity Connection server; to view the list of languages actually installed on a server, use this GET method instead:

```
GET http://<connection-server>/vmrest/installedlanguages
```

Note that currently, the API to retrieve the language map and the API to retrieve the installed languages on a server both require System Administrator access.

## Listing and Viewing

The following is an example of a GET that lists all language mappings in the language map:

```
GET http://<connection-server>/vmrest/languagemap
```

The following is an excerpt of the response from the above GET request:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LanguageMappings total="149">
  <LanguageMapping>
    <LanguageCode>1025</LanguageCode>
    <LanguageAbbreviation>ARA</LanguageAbbreviation>
    <LanguageTag>ar-SA</LanguageTag>
  </LanguageMapping>
  <LanguageMapping>
    <LanguageCode>1026</LanguageCode>
    <LanguageAbbreviation>BGR</LanguageAbbreviation>
    <LanguageTag>bg-BG</LanguageTag>
  </LanguageMapping>
  <LanguageMapping>
    <LanguageCode>1027</LanguageCode>
    <LanguageAbbreviation>CAT</LanguageAbbreviation>
    <LanguageTag>ca-ES</LanguageTag>
  </LanguageMapping>
  .
  .
  .
</LanguageMappings>
```

You can also use the query parameters `rowsPerPage` and `pageNumber` to limit the number of returned results. For example:

```
GET http://<connection-server>/vmrest/languagemap?rowsPerPage=2&pageNumber=3
```

This request returns the two languages on the third page, as follows:

```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LanguageMappings total="149">
  <LanguageMapping>
    <LanguageCode>1029</LanguageCode>
    <LanguageAbbreviation>CSY</LanguageAbbreviation>
    <LanguageTag>cs-CZ</LanguageTag>
  </LanguageMapping>
  <LanguageMapping>
    <LanguageCode>1030</LanguageCode>
    <LanguageAbbreviation>DAN</LanguageAbbreviation>
    <LanguageTag>da-DK</LanguageTag>
  </LanguageMapping>
</LanguageMappings>

```

To view the language mapping for a specific language, you can use either the language code or the language abbreviation. For example, to find out what language the language code 1041 corresponds to, use:

```
GET http://<connection-server>/vmrest/languagemap/1041
```

The following is the response from the above GET request:

```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LanguageMapping>
  <LanguageCode>1041</LanguageCode>
  <LanguageAbbreviation>JPN</LanguageAbbreviation>
  <LanguageTag>ja-JP</LanguageTag>
</LanguageMapping>

```

Similarly, you can use the language abbreviation to find the corresponding language code. For example, to find the language code for US English (abbreviated ENU), use:

```
GET http://<connection-server>/vmrest/languagemap/ENU
```

This GET request yields the following response, which indicates that 1033 is the language code for US English:

```

200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LanguageMapping>
  <LanguageCode>1033</LanguageCode>
  <LanguageAbbreviation>ENU</LanguageAbbreviation>
  <LanguageTag>en-US</LanguageTag>
</LanguageMapping>

```

## Miscellaneous

For a list of supported languages and the corresponding language codes/abbreviations, see the section "Numeric and Alphabetic Codes for Supported Languages in Cisco Unity Connection" in the *System Requirements for Cisco Unity Connection Release 8.x* available at [http://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/connection/8x/requirements/8xcucsysreqs.html](http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/connection/8x/requirements/8xcucsysreqs.html)

# Cisco Unity Connection Provisioning Interface (CUPI) API -- LDAP Phone Number Transform

## About LDAP Phone Number Transform

This page contains information on how to use the API to create, list, update, and delete LDAP phone number transforms.

Beginning with Cisco Unity Connection 8.5, we support up to one LDAP phone number transform, which consists of a regular expression and a replacement pattern.

## Listing and Viewing

The following is an example of a GET request that lists all LDAP phone number transforms:

```
GET http://<connection-server>/vmrest/ldapphonenumberttransforms
```

The following is the response from the above GET request:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LdapPhoneNumberTransforms total="1">
  <LdapPhoneNumberTransform>
    <URI>/vmrest/ldapphonenumberttransforms/0d49a281-cc35-4b8b-bccc-f94f8b8903bd</URI>
    <ObjectId>0d49a281-cc35-4b8b-bccc-f94f8b8903bd</ObjectId>
    <Regex>.*([0-9][0-9][0-9][0-9])</Regex>
    <Replacement>9$1</Replacement>
  </LdapPhoneNumberTransform>
</LdapPhoneNumberTransforms>
```

## Creating

The required fields for creating an LDAP phone number transform are Regex and Replacement.

The following is an example of a POST request that creates an LDAP phone number transform with the Regex field set to ".\*([0-9][0-9][0-9][0-9])", and the Replacement field set to "9\$1":

```
POST https://<connection-server>/vmrest/ldapphonenumberttransforms/

<LdapPhoneNumberTransform>
  <Regex>.*([0-9][0-9][0-9][0-9])</Regex>
  <Replacement>9$1</Replacement>
</LdapPhoneNumberTransform>
```

The following is the response from the above POST request:

```
201
Created
/vmrest/ldapphonenumberttransforms/0d49a281-cc35-4b8b-bccc-f94f8b8903bd
```

## Updating

The Regex and Replacement fields of an LDAP Phone Number Transform can be updated via a PUT request.

The following is an example of a PUT request that modifies the Regex and Replacement fields of an existing LDAP phone number transform:

```
https://<connection-server>/vmrest/ldaphonenumbertransforms/  
0d49a281-cc35-4b8b-bccc-f94f8b8903bd  
<LdapPhoneNumberTransform>  
  <Regex>.*([0-9][0-9][0-9][0-9][0-9])</Regex>  
  <Replacement>8$1</Replacement>  
</LdapPhoneNumberTransform>
```

The following is the response from the above PUT request:

```
204  
No Content  
null
```

## Deleting

The following is an example of a DELETE request that deletes an LDAP phone number transform:

```
DELETE https://<connection-server>/vmrest/ldaphonenumbertransforms/  
0d49a281-cc35-4b8b-bccc-f94f8b8903bd
```

The following is the response from the above DELETE request:

```
204  
No Content  
null
```

## Possible Errors

As of Connection 8.5, we support one LDAP phone number transform only, so if you attempt to create a new LDAP phone number transform (via the POST request) when there is already an existing LDAP phone number transform, the following error will be returned:

```
405  
Method Not Allowed  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<ErrorDetails>  
  <errors>  
    <code>METHOD_NOT_ALLOWED</code>  
    <message>Unable to perform requested method because an LDAP phone number transform  
already exists. (This version of Cisco Unity Connection supports up to one LDAP phone number  
transform only.)</message>  
  </errors>  
</ErrorDetails>
```

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Authorization Server

Links to Other API pages: [Cisco\\_Unity\\_Connection\\_APIS](#)

## About Authorization Server (Authz Server)

Unity Connection 11.5(1) SU3 and later allows the use of **OAuth 2.0 Authorization Code Grant Flow** to authenticate the Jabber users. This requires an Authorization Server (Authz Server) that provides the authorization keys to Unity Connection for validating the Jabber user. In Unity Connection, the publisher server of Cisco Unified CM associated with a phone system is configured as an Authz server.

For more information on Authorization Code Grant Flow, see "TBD".

This page contains information on how to use the API to create, list, modify, and delete an Authz server.

## Listing Authz Servers

The following is an example of the GET request that lists the Authz servers:

```
GET https://<connection-server>/vmrest/authz/server/
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<AuthzServers total="2">
<AuthzServer>
  <ObjectId>f976720d-ed5b-4cbc-95b2-e47e1e2663af</ObjectId>
  <DisplayName>AuthzServer1</DisplayName>
  <IgnoreCertificateError>true</IgnoreCertificateError>
  <ServerUsername>admin</ServerUsername>
  <ServerPassword />
  <ServerNodeAddress>10.76.215.238</ServerNodeAddress>
  <Port>8443</Port>
  <LastSyncStatus>Passed - May 12,2017 07:00:53 AM GMT</LastSyncStatus>
</AuthzServer>
<AuthzServer>
  <ObjectId>69b82e6a-a77d-499e-a953-cb7f589254aa</ObjectId>
  <DisplayName>AuthzServer2</DisplayName>
  <IgnoreCertificateError>false</IgnoreCertificateError>
  <ServerUsername>admin</ServerUsername>
  <ServerPassword />
  <ServerNodeAddress>10.76.215.239</ServerNodeAddress>
  <Port>8443</Port>
  <LastSyncStatus>Passed - May 12,2017 07:00:53 AM GMT</LastSyncStatus>
</AuthzServer>
</AuthzServers>
```

```
Response Code: 200
```

## Viewing an Authz Server

The following is an example of the GET request that lists a particular Authz server represented by <ObjectId>:

```
GET https://<connection-server>/vmrest/authz/server/<ObjectId>
```

The following is the response from the above \*GET\* request and the actual response will depend upon the information given by you:

```
<AuthzServer>
  <ObjectId>0fa45cde-c289-415a-853f-3d706b3952d5</ObjectId>
  <DisplayName>AuthzServer1</DisplayName>
  <IgnoreCertificateError>true</IgnoreCertificateError>
  <ServerUsername>admin</ServerUsername>
  <ServerPassword />
  <ServerNodeAddress>10.76.215.238</ServerNodeAddress>
  <Port>8443</Port>
</AuthzServer>
```

```
Response Code: 200
```

## Adding New Authz Server

The following is an example of the POST request that adds the Authz servers:

```
POST https://<connection_server>/vmrest/authz/server/
```

The following is an example of the response from the above \*POST\* request and the actual response will depend upon the information given by you:

```
<AuthzServer>
  <DisplayName>AuthzServer1</DisplayName>
  <IgnoreCertificateError>true</IgnoreCertificateError>
  <ServerUsername>admin</ServerUsername>
  <ServerPassword>*****</ServerPassword>
  <ServerNodeAddress>10.76.215.238</ServerNodeAddress>
  <Port>8443</Port>
</AuthzServer>
```

```
Response Code: 201
```

## Modifying Authz Server

The following is an example of the PUT request that modifies the Authz Server as represented by <ObjectId>:

```
PUT https://<connection_server>/vmrest/authz/server/<objectId>
```

The following is an example of the response from the above \*PUT\* request and the actual response will depend upon the information given by you:

```
<AuthzServer>
  <ServerNodeAddress>ucbu-aricent-vm475</ServerNodeAddress>
  <Port>8443</Port>
</AuthzServer>
```

```
Response Code: 204
```

## Deleting an Authz Server

The following is an example of the DELETE request that deletes an Authz Server as represented by <ObjectId>:

```
DELETE https://<connection_server>/vmrest/authz/server/<objectId>
```

The input for the PUT request will be. The output for this request returns the successful response code.

```
Response Code: 204
```

## Synchronization of Authorization Keys

The following is an example of the PUT request that synchronize the authorization keys between Authz server and Unity Connection:

```
PUT https://<connection_server>/vmrest/authz/sync/<objectId>
```

The output for this request returns the successful response code.

```
Response Code: 204
```

## Explanation of Data Fields

The following chart lists all of the data fields:

Parameter	Operations	Data Type	Comments
ObjectId	Read Only	String (36)	Unique identifier for Authz server
DisplayName	Read/Write	String (64)	Descriptive name of an Authz server
IgnoreCertificateError	Read/Write	Boolean	<p>A flag that validates the certificates for the Authz server. Possible values are:</p> <ul style="list-style-type: none"> <li>• false: Validates the certificates for the Authz server.</li> <li>• true: Ignore the certificate validation errors for the Authz server.</li> </ul> <p>For more information on certificates, see "Security" chapter of <i>Cisco Unified Communications Operating System Administration Guide for Cisco Unity Connection Release 12.x</i> at <a href="https://www.cisco.com/c/en/us/products/collateral/voice/ip_communication/12x_administration_12x_wsg.pdf">https://www.cisco.com/c/en/us/products/collateral/voice/ip_communication/12x_administration_12x_wsg.pdf</a></p>
ServerUsername	Read/Write	String (64)	User name that Unity Connection uses to sign in to the Authz server.
ServerPassword	Write Only	String (64)	Password that Unity Connection uses to sign in to the Authz server.



Parameter	Operations	Data Type	Comments
ServerNodeAddress	Read/Write	String (64)	Hostname, IP address or Fully-Qualified Domain Name (FQDN) of the Authz server that Unity Connection connects to.
Port	Read/Write	Integer	The port through which Unity Connection connects to the Authz server.
LastSyncStatus	Read Only	String (36)	The status and time of the last synchronization of authorization keys between Authz server and Unity Connection.

# Cisco Unity Connection Provisioning Interface (CUPI) API -- System Default Language and TTS Language

## About System Default Language and System Default TTS Language

This page contains information on how to use the API to display and update the System Default Language and the System Default TTS Language.

The System Default Language and System Default TTS Language must be one of the installed languages on your Cisco Unity Connection server. To retrieve the list of installed languages on your system, use:

```
GET http://<connection-server>/vmrest/installedlanguages
```

## Viewing

The System Default Language and System Default TTS Language are considered part of the "local" Connection server. To retrieve the current values for these fields, use:

```
GET http://<connection-server>/vmrest/locations/locallocation
```

The following is the response from the above GET request:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<LocalLocation>
  <DefaultLanguage>1033</DefaultLanguage>
  <DefaultTTSLanguage>1033</DefaultTTSLanguage>
</LocalLocation>
```

Note that the returned values are the language codes for the System Default Language and System Default TTS language. In this case, 1033 is the code for US English.

## Updating

The following is an example of a PUT request that modifies the System Default Language and System Default TTS Language. In this example, we set both to Japanese (language code 1041). Note that the language codes

must be valid, and that they must correspond to a language that has been installed and licensed on your Connection server.

```
PUT https://<connection-server>/vmrest/locations/locallocation

<LocalLocation>
  <DefaultLanguage>1041</DefaultLanguage>
  <DefaultTTSLanguage>1041</DefaultTTSLanguage>
</LocalLocation>
```

The following is the response from the above PUT request:

```
204
No Content
null
```

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Wave Formats

## About Wave Formats

This page contains information on how to use the API to list wave formats. Cisco Unity Connection supports several different wave formats that recorded messages can be stored in. This URI is accessible to both users and administrators

## Listing and Viewing

The following is an example of a GET that lists all wave formats:

```
GET http://<connection-server>/vmrest/waveformats
```

To retrieve a specific wave format by its object ID:

```
GET http://<connection-server>/vmrest/waveformats/<objectid>
```

The following is an example response from the above GET request:

```
200
OK
<?xml version="1.0" encoding="UTF-8"?>
<WaveFormat>
  <URI>/vmrest/waveformats/cb85b520-e2de-4878-96e2-3331607f4671</URI>
  <ObjectId>cb85b520-e2de-4878-96e2-3331607f4671</ObjectId>
  <AvgBytesPerSec>1000</AvgBytesPerSec>
  <BitsPerSample>0</BitsPerSample>
  <BlockAlign>10</BlockAlign>
  <Channels>1</Channels>
  <FormatName>G.729a</FormatName>
  <FormatTag>307</FormatTag>
  <SamplesPerSec>8000</SamplesPerSec>
  <JavaEncoding>G.729a</JavaEncoding>
  <CodecId>3</CodecId>
</WaveFormat>
```