# Cisco Unity Connection Provisioning Interface (CUPI) API -- Call Handler Settings


- Cisco Unity Connection Provisioning Interface (CUPI) API -- Assigning a Schedule to a Call Handler, on page 1
- Cisco Unity Connection Provisioning Interface (CUPI) API -- Caller Input Keys , on page 5
- Cisco Unity Connection Provisioning Interface (CUPI) API -- Greetings , on page 9
- Cisco Unity Connection Provisioning Interface (CUPI) API -- Updating Caller Input Keys, on page 11
- Cisco Unity Connection Provisioning Interface (CUPI) API -- Custom Recordings, on page 15


# Cisco Unity Connection Provisioning Interface (CUPI) API -- Assigning a Schedule to a Call Handler

## Introduction

In order to assign a schedule to a call handler, such as the Opening Greeting Call Handler, an administrator should first create a ScheduleSet and its supporting objects, and then change the Call Handler to point to that new ScheduleSet.

## Creating a Schedule

Schedules are relatively complicated objects in Cisco Unity Connection. They are comprised of 4 different types of objects, and the top level object is called a ScheduleSet. For a more detailed explanation of how the schedules work in Unity Connection, check Schedules Overview

When a schedule has been created, the administrator will need the ScheduleSet's ObjectId in order to associate it with a call handler.

## Associating a ScheduleSet with a Call Handler

### Retrieving the List of Call Handlers

In order to retrieve the list of call handlers, an administrator makes a GET request to the call handler resource:

```
GET /vmrest/handlers/callhandlers
```

This will return a list of all call handlers on the system. A portion of that list is shown here:

```
200
OK

<?xml version="1.0" encoding="UTF-8"?>
<Callhandlers total="5">
  <Callhandler>
    <URI>/vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad</URI>
    <CreationTime>2010-05-25T18:38:51Z</CreationTime>
    <Language>1033</Language>
    <Undeletable>true</Undeletable>
    <VoiceName>0164efab-5c35-42bd-8284-018746edc64b.wav</VoiceName>
   <VoiceFileURI>/vmrest/voicefiles/0164efab-5c35-42bd-8284-018746edc64b.wav</VoiceFileURI>

    <LocationObjectId>6919b242-ed60-4f3a-95fa-40967171485e</LocationObjectId>

<LocationURI>/vmrest/locations/connectionlocations/6919b242-ed60-4f3a-95fa-40967171485e</LocationURI>

    <EditMsg>true</EditMsg>
    <IsPrimary>false</IsPrimary>
    <OneKeyDelay>1500</OneKeyDelay>
    <ScheduleSetObjectId>a7e21c61-296d-46f1-9860-50966bbfbb8e</ScheduleSetObjectId>

<ScheduleSetURI>/vmrest/schedulesets/a7e21c61-296d-46f1-9860-50966bbfbb8e</ScheduleSetURI>

    <SendUrgentMsg>0</SendUrgentMsg>
    <MaxMsgLen>300</MaxMsgLen>
    <IsTemplate>false</IsTemplate>
    <ObjectId>03991ce8-0eaa-40cc-86a9-c0c88d9066ad</ObjectId>

<RecipientDistributionListObjectId>7082dace-606c-4f2c-8af1-764d149e4500</RecipientDistributionListObjectId>


<RecipientDistributionListURI>/vmrest/distributionlists/7082dace-606c-4f2c-8af1-764d149e4500</RecipientDistributionListURI>

    <DisplayName>Opening Greeting</DisplayName>
    <AfterMessageAction>2</AfterMessageAction>
    <AfterMessageTargetConversation>PHGreeting</AfterMessageTargetConversation>

<AfterMessageTargetHandlerObjectId>3910fd6b-8d56-4b83-89a6-d5c825d69916</AfterMessageTargetHandlerObjectId>

    <TimeZone>4</TimeZone>
    <UseDefaultLanguage>true</UseDefaultLanguage>
    <UseDefaultTimeZone>true</UseDefaultTimeZone>
    <MediaSwitchObjectId>2d4be643-a206-4705-92a9-261d191f2df4</MediaSwitchObjectId>

<PhoneSystemURI>/vmrest/phonesystems/2d4be643-a206-4705-92a9-261d191f2df4</PhoneSystemURI>

    <UseCallLanguage>true</UseCallLanguage>
    <SendSecureMsg>false</SendSecureMsg>
    <EnablePrependDigits>false</EnablePrependDigits>
    <DispatchDelivery>false</DispatchDelivery>
   <CallSearchSpaceObjectId>44228e57-f458-4d56-809a-8d33fbdecb56</CallSearchSpaceObjectId>


<CallSearchSpaceURI>/vmrest/searchspaces/44228e57-f458-4d56-809a-8d33fbdecb56</CallSearchSpaceURI>

    <InheritSearchSpaceFromCall>true</InheritSearchSpaceFromCall>
    <PartitionObjectId>5d61a103-87a8-41c6-ba0b-90e1c72eda7c</PartitionObjectId>
    <PartitionURI>/vmrest/partitions/5d61a103-87a8-41c6-ba0b-90e1c72eda7c</PartitionURI>

<GreetingsURI>/vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad/greetings</GreetingsURI>


<TransferOptionsURI>/vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad/transferoptions</TransferOptionsURI>
```

```
<MenuEntriesURI>/vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad/menuentries</MenuEntriesURI>

  </Callhandler>
  <Callhandler>
    ...
  </Callhandler>
  ...
</Callhandlers>
```

## Selecting the Call Handler

From the list of call handlers obtained by doing the previous step, the administrator should choose one, usually based on DisplayName. In program code, this can be done by loading the returned XML document into an XML parser, and then finding the node with the requested DisplayName.

In this example, let's choose the Opening Greeting Call Handler, meaning the call handler with the DisplayName "Opening Greeting." The URI field tells the administrator which URI to use in order to retrieve, modify, or delete the call handler. In this example, the URI is /vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad.

## Modifying the Call Handler

In order the change the schedule associated with a call handler, the administrator makes a PUT to the call handler resource, requesting a modification to the ScheduleSetObjectId field. Let's say that 9dd6c1d5-249e-4715-8953-396ce2f26314 is the ObjectId for the ScheduleSet that the administrator created earlier.

```
PUT /vmrest/handlers/callhandlers/03991ce8-0eaa-40cc-86a9-c0c88d9066ad

<Callhandler>
  <ScheduleSetObjectId>9dd6c1d5-249e-4715-8953-396ce2f26314</ScheduleSetObjectId>
</Callhandler>
```

At this point, the call handler will now be active during the time periods described by the ScheduleSet. About Caller Input Keys

## Finding Which Schedule Is Assigned to a Call Handler

In order to find out which schedule is currently assigned to a call handler, an administrator makes a GET request to the call handler resource, as described earlier. After selecting a call handler from the list, the ScheduleSetURI field provides a link to the ScheduleSet that is associated with the call handler. The administrator can then make a GET request to that URI to retrieve the ScheduleSet:

```
GET /vmrest/schedulesets/9dd6c1d5-249e-4715-8953-396ce2f26314
```

That will return the ScheduleSet. From that ScheduleSet object, the administrator can retrieve the ScheduleSetMembers, Schedules, and ScheduleDetails that comprise the entire Schedule.

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Caller Input Keys

## About Caller Input Keys

(Note that this content is applicable to Cisco Unity Connection 7.1(3) and later)

In order to use the CUPI API to update caller input keys, you need to determine the object ID of the call handler whose caller input key you want to change.

To make an update to caller input keys for a user, you need to determine the object ID of the user's call handler. Every user who has caller input keys has a call handler. When looking at the user data returned from the following GET, the object ID is under the element <CallHandlerObjectId>:

```
GET https://<server>/vmrest/users/<userobjectid>
```

When you have determined the object ID of the call handler that you want to change, you can do any of the following operations.

## Ignore

The default setting of most keys is "Ignore." To set a key to ignore, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>0</Action>
</MenuEntry>
```

The Action field is a custom type used to determine at a basic level what the caller input key is going to do. The 0 value denotes that this key should be ignored.

## Hang Up

To set a key to hang up, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>1</Action>
</MenuEntry>
```

The Action field of 1 denotes that this key should terminate the call.

## Restart Greeting

To set a key to restart the greeting, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>6</Action>
</MenuEntry>
```

The Action field of 6 denotes that this key should restart playback of the greeting from the beginning.

# Route from Next Call Routing Rule

To set a key to route the call starting from the next call routing rule, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>8</Action>
</MenuEntry>
```

The Action field of 8 denotes that this key should route the call starting from the next call routing rule.

# Skip Greeting

To set a key to skip the greeting, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>5</Action>
</MenuEntry>
```

The Action field of 5 denotes that this key should skip the remainder of the greeting and take the caller directly to the after greeting action.

# Take a Message

To set a key to take a message, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>4</Action>
</MenuEntry>
```

The Action field of 4 denotes that this key should prompt the caller to record a message.

# Transfer to an Alternate Contact Number

To transfer a caller to an alternate contact number, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>7</Action>
<TransferNumber>number to transfer to</TransferNumber>
</MenuEntry>
```

The Action field of 7 denotes that this key should transfer the caller to an alternate contact number. The TransferNumber is the number that the caller will be release-transferred to. Note that including a TransferNumber is optional; if you choose not to include a TransferNumber, users can configure their own alternate contact numbers by phone (by providing a number to transfers caller to).

Currently, the CUPI API does not allow you to set the TransferNumber to an empty string. Therefore, if you set the TransferNumber, you can change it later to a different number, but you cannot leave it empty. In addition, the CUPI API does not allow read or write access to which type of transfer the alternate contact number uses (release transfer vs. supervised transfer).

# Transfer to a User or a Call Handler

To transfer a caller to a user or a call handler, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>PHTransfer</TargetConversation>
<TargetHandlerObjectId>05d9e169-5c87-4415-aaed-c58a14816c8d</TargetHandlerObjectId>
</MenuEntry>
```

The Action field of 2 denotes that this key should transfer the caller to another conversation. This is used to transfer callers to other objects, or to send callers to other conversations such as the system transfer conversation.

The TargetConversation should be set to PHTransfer if you want to transfer to the call handler in question. If you want to have the call go directly the call handler greeting, set it to PHGreeting instead.

The TargetHandlerObjectId is the object ID of the call handler that you want the key to transfer the caller to.

# Transfer to an Interview Handler

To transfer a caller to an interview handler, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>PHInterview</TargetConversation>
<TargetHandlerObjectId>interview handler object id</TargetHandlerObjectId>
</MenuEntry>
```

The TargetConversation should be set to PHInterview. The TargetHandlerObjectId is the object ID of the interview handler that you want to the caller input key to go to.

# Transfer to a Directory Handler

To transfer a caller to a directory handler, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>AD</TargetConversation>
<TargetHandlerObjectId>object id of directory handler</TargetHandlerObjectId>
</MenuEntry>
```

The TargetConversation should be set to AD. The TargetHandlerObjectId is the object ID of the directory handler that you want to the caller input key to go to.

# Transfer to the Broadcast Message Administrator Conversation

To transfer a caller to the Broadcast Message Administrator conversation, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>BroadcastMessageAdministrator</TargetConversation>
</MenuEntry>
```

# Transfer to the Caller System Transfer Conversation

To transfer a caller to the Caller System Transfer conversation, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>SystemTransfer</TargetConversation>
</MenuEntry>
```

# Transfer to the Greetings Administrator Conversation

To transfer a caller to the Greetings Administrator conversation, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>GreetingsAdministrator</TargetConversation>
</MenuEntry>
```

# Transfer to the Sign-In Conversation

To transfer a caller to the sign-in conversation, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>SubSignIn</TargetConversation>
</MenuEntry>
```

## Transfer to the User System Transfer Conversation

To transfer a caller to the subscriber system transfer conversation, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>SubSysTransfer</TargetConversation>
</MenuEntry>
```

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

## About Greetings

Modifying greetings involves a fairly long hierarchy in the URI, as follows:

```
/vmrest/handlers/callhandlers/<call handler object id>/greetings/<greeting type>/
greetingstreamfiles/<language id>
```

The greeting types are:

- Standard
- Alternate
- Busy
- Closed
- Holiday
- Error
- Internal

For Connection versions 7.x and 8.0.x the greetings are accessed by using the greeting stream file URI. Modifying the greeting requires a three step process as detailed below.

For Connection versions 8.5 and later the greeting audio access has been simplified so that it can be modified in a single step. This new URI is a standard sub-resource of the greeting stream file resource URI. The old greeting stream file modification still works in the later versions, but use of the new URI is easier.

## Listing and Viewing

### Greeting Audio GET for 8.5 and Later

Use the standard greeting audio URI to get the file:

```
GET http://<connection-server>/vmrest/handlers/callhandlers/<call handler object
id>/greetings
/<greeting type>/greetingstreamfiles/<language id>/audio
```

The response will return the audio/wav data for the greeting of the specified type and language.

## Voice Name GET for 7.x and 8.0.x

First get the greeting stream file object for the greeting type and language, then use the voice files URI to get the greeting audio contents:

```
GET http://<connection-server>/vmrest/handlers/callhandlers/<call handler object
id>/greetings
/<greeting type>/greetingstreamfiles/<language id>
GET http://<connection-server>/vmrest/voicefiles/<stream file>
```

The response will return the audio/wav data for the greeting.

# Setting Greetings

## Setting a Greeting in 8.5 and Later

PUT the audio data directly to the standard greeting stream file URI:

```
PUT http://<connection-server>/vmrest/handlers/callhandlers/<call handler object
id>/greetings
/<greeting type>/greetingstreamfiles/<language id>/audio content is audio/wav data
```

The response is a 204 indicating that the content has been accepted and copied into the temporary file.

## Setting a Greeting in 7.x and 8.0.x

To create a greeting for a resource is a three step process.

**Step 1**: A place-holder for the WAV file must be created with a POST. This is a temporary file place-holder that can be used for up to 30 minutes. If it is not used within 30 minutes (assigned to a resource), the file is assumed to be abandoned and gets automatically cleaned.

```
POST /vmrest/voicefiles
```

The response code is 201 and the content is the name of the newly created temporary file.

**Step 2**: Use the temporary file name to PUT the new audio data. The HTTP content type is "audio/wav" and the payload content is the audio data.

```
PUT /vmrest/voicefiles/<temporary file name>
```

The response is a 204 indicating that the content has been accepted and copied into the temporary file.

**Step 3**: Set the greeting stream file of the target resource to the temporary file name.

See the following example

```
PUT http://<connection-server>/vmrest/handlers/callhandlers/<call handler object
id>/greetings
/<greeting type>/greetingstreamfiles/<language id>
```

Only the stream file field needs to be filled out for PUT. All the other greeting stream file fields are derived from the URI:

```
<?xml version="1.0" encoding="UTF-8"?>
<GreetingStreamFile>
<StreamFile>temporary file name</StreamFile>
</GreetingStreamFile>
```

# Cisco Unity Connection Provisioning Interface (CUPI) API -- Updating Caller Input Keys

## Introduction

This document describes how to change the caller input key on a call handler to take different actions. This content is applicable to Cisco Unity Connection release 7.1(3) and later. These examples have not been verified with versions prior to 7.1(3).

In order to change a caller input key for a call handler, you need to know the object ID of the call handler.

In order to change a caller input key for a user, you need the object ID of that user's call handler. Every user who has caller input keys has a call handler, and if you are looking at the user data returned from a GET request like the following, the object ID is the element **CallHandlerObjectId**.

```
GET https://<server>/vmrest/users/<userobjectid>
```

When you have obtained the object ID of the call handler you want to change, you can do any of the following actions.

## Examples

### Ignore

The default setting of most keys is "Ignore." To set a key to ignore, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>0</Action>
</MenuEntry>
```

The Action field is a custom type used to determine at a basic level what the caller input key is going to do. The 0 value denotes that this key should be ignored.

### Hang Up

To set a key to hang up, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>1</Action>
</MenuEntry>
```

The Action field of 1 denotes that this key should terminate the call.

## Restart Greeting

To set a key to restart the greeting, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>6</Action>
</MenuEntry>
```

The Action field of 6 denotes that this key should restart playback of the greeting from the beginning.

## Route from Next Call Routing Rule

To set a key to route the call starting from the next call routing rule, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>8</Action>
</MenuEntry>
```

The Action field of 8 denotes that this key should route the call starting from the next call routing rule.

## Skip Greeting

To set a key to skip the greeting, do the following PUT request:

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>5</Action>
</MenuEntry>
```

The Action field of 5 denotes that this key should skip the remainder of the greeting and take the caller directly to the after greeting action.

## Take a Message

To set a key to take a message, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>4</Action>
</MenuEntry>
```

The Action field of 4 denotes that this key should prompt the caller to record a message.

## Alternate Contact Number (ACN)

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <MenuEntry>
    <Action>7</Action>
    <TransferNumber>number to transfer to</TransferNumber>
  </MenuEntry>
```

The Action field of 7 denotes that this key should transfer the caller to an alternate contact number. The TransferNumber is the number that the caller will be release-transferred to. Including a transfer number is optional; if you do not include a TransferNumber, and set a key to the Action of 7, users will have the option of configuring their own ACNs by phone, which provides a number to transfer callers to.

The API currently does not let you set the TransferNumber to an empty string. That means that when you have set the TransferNumber, you can only change it to a different number, but cannot empty it out. The API also currently does not give you read or write access to the type of transfer the ACN uses (release vs. supervised).

## Transfer to a User or a Call Handler

To transfer a caller to a user or a call handler, do the following PUT request:

```
PUT https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
<Action>2</Action>
<TargetConversation>PHTransfer</TargetConversation>
<TargetHandlerObjectId>05d9e169-5c87-4415-aaed-c58a14816c8d</TargetHandlerObjectId>
</MenuEntry>
```

The Action field of 2 denotes that this key should transfer the caller to another conversation. This is used to transfer callers to other objects, or to send callers to other conversations such as the system transfer conversation.

The TargetConversation should be set to PHTransfer if you want to transfer to the call handler in question. If you want to have the call go directly the call handler greeting, set it to PHGreeting instead.

The TargetHandlerObjectId is the object ID of the call handler that you want the key to transfer the caller to.

## Go to an Interview Handler

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <MenuEntry>
    <Action>2</Action>
    <TargetConversation>PHInterview</TargetConversation>
    <TargetHandlerObjectId>interview handler object id</TargetHandlerObjectId>
  </MenuEntry>
```

The TargetConversation should be set to PHInterview and the TargetHandlerObjectId is the object ID of the interview handler that you want to the caller input key to go to.

## Go to a Directory Handler

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <MenuEntry>
    <Action>2</Action>
    <TargetConversation>AD</TargetConversation>
    <TargetHandlerObjectId>object id of directory handler</TargetHandlerObjectId>
  </MenuEntry>
```

The TargetConversation should be set to AD and the TargetHandlerObjectId is the object ID of the directory handler you want to the caller input key to go to.

## Go to the Broadcast Message Administrator Conversation

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
  <Action>2</Action>
  <TargetConversation>BroadcastMessageAdministrator</TargetConversation>
</MenuEntry>
```

## Go to the User System Transfer Conversation

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
  <Action>2</Action>
  <TargetConversation>SubSysTransfer</TargetConversation>
</MenuEntry>
```

## Go to the Greetings Administrator Conversation

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
  <Action>2</Action>
  <TargetConversation>GreetingsAdministrator</TargetConversation>
</MenuEntry>
```

## Go to the Sign-In Conversation

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
  <Action>2</Action>
  <TargetConversation>SubSignIn</TargetConversation>
</MenuEntry>
```

## Go to the User System Transfer Conversation

```
PUT  https://<server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/menuentries/<key>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MenuEntry>
  <Action>2</Action>
  <TargetConversation>SubSysTransfer</TargetConversation>
</MenuEntry>
```

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

## About Custom Recordings

Cisco Unity Connection allows you to play the customized recordings after a message is sent. You can also play the custom recordings after a greeting is played. A greeting is played before the callers are allowed to leave a message for a user or a call handler.

**Note** In Cisco Unity Connection 9.0, the **Post Greeting Recordings** option is changed to the **Custom Recordings option**.

## Listing Custom Recording

**Example 1**

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

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

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:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<PostGreetingRecordings total="2">
<PostGreetingRecording>
  <URI>/vmrest/postgreetingrecordings/b7cfa709-9f06-4cd2-94b3-8e5910d5b041</URI>
  <ObjectId>b7cfa709-9f06-4cd2-94b3-8e5910d5b041</ObjectId>
  <DisplayName>Rec-2</DisplayName>
  <PostGreetingRecordingStreamFilesURI>/vmrest/postgreetingrecordings/b7cfa709-
9f06-4cd2-94b3-8e5910d5b041/postgreetingrecordingstreamfiles</PostGreetingRecordingStreamFilesURI>

  </PostGreetingRecording>
<PostGreetingRecording>
  <URI>/vmrest/postgreetingrecordings/db44de42-7274-4d1c-8837-e710b0649c13</URI>
  <ObjectId>db44de42-7274-4d1c-8837-e710b0649c13</ObjectId>
  <DisplayName>Rec-3</DisplayName>
  <PostGreetingRecordingStreamFilesURI>/vmrest/postgreetingrecordings/db44de42-7274-
4d1c-8837-e710b0649c13/postgreetingrecordingstreamfiles</PostGreetingRecordingStreamFilesURI>

</PostGreetingRecording>
</PostGreetingRecordings>
```

```
RESPONSE Code: 200
```

**Example 2**

The following is an example of the GET request that lists the custom recording as represented by <objectId>:

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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<PostGreetingRecording>
  <URI>/vmrest/postgreetingrecordings/b7cfa709-9f06-4cd2-94b3-8e5910d5b041</URI>
  <ObjectId>b7cfa709-9f06-4cd2-94b3-8e5910d5b041</ObjectId>
  <DisplayName>Rec-2</DisplayName>

<PostGreetingRecordingStreamFilesURI>/vmrest/postgreetingrecordings/b7cfa709-9f06-4cd2-94b3-8e5910d5b041/postgreetingrecordingstreamfiles</PostGreetingRecordingStreamFilesURI>

</PostGreetingRecording>
```

```
RESPONSE Code: 200
```

# Listing Languages for a Custom Recording

**Example 1**

The following is an example of the GET request that lists all the languages/stream files for a custom recording:

```
https://<connection_server>/vmrest/postgreetingrecordings/<objectId>/postgreetingrecordingstreamfiles
```

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:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<PostGreetingRecordingStreamFiles>
<PostGreetingRecordingStreamFile>

<URI>/vmrest/postgreetingrecordings/b7cfa709-9f06-4cd2-94b3-8e5910d5b041/postgreetingrecordingstreamfiles/1033/audio</URI>


<PostGreetingRecordingObjectId>b7cfa709-9f06-4cd2-94b3-8e5910d5b041</PostGreetingRecordingObjectId>

  <LanguageCode>1033</LanguageCode>
  <StreamFile>58b0c574-62e6-4276-8e25-cb1227570492.wav</StreamFile>

<PostGreetingStreamFileLanguageURI>/vmrest/postgreetingrecordings/b7cfa709-9f06-4cd2-94b3-8e5910d5b041/postgreetingrecordingstreamfiles/1033</PostGreetingStreamFileLanguageURI>

</PostGreetingRecordingStreamFile>
</PostGreetingRecordingStreamFiles>
```

```
RESPONSE Code: 200
```

**Example 2**

The following is an example of the GET request that lists a particular languages/stream files for a custom recording represented by <language-code>:

```
https://<connection_server>/vmrest/postgreetingrecordings/<objectId>/postgreetingrecordingstreamfiles/<language-code>
```

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:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<PostGreetingRecordingStreamFile>

<PostGreetingRecordingObjectId>b7cfa709-9f06-4cd2-94b3-8e5910d5b041</PostGreetingRecordingObjectId>

  <LanguageCode>1033</LanguageCode>
  <StreamFile>58b0c574-62e6-4276-8e25-cb1227570492.wav</StreamFile>
</PostGreetingRecordingStreamFile>
```

```
RESPONSE Code: 200
```

# Modifying Custom Recording

**Example 1**

The following is an example of the PUT request that modifies the custom recording as represented by <objectId>:

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

```
<PostGreetingRecording>
     <DisplayName>custom-rec</DisplayName>
</PostGreetingRecording>
```

```
RESPONSE Code: 204
```

**Example 2**

The following is an example of the PUT request that modifies a stream file for a custom recording as represented by <language-code>:

```
https://<connection_server>/vmrest/postgreetingrecordings/<object-id>/postgreetingrecordingstreamfiles/<language-code>
```

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:

```
<PostGreetingRecordingStreamFile>
   <StreamFile>82dd9884-5cee-42f0-8537-bdc5e622de1f.wav</StreamFile>
</PostGreetingRecordingStreamFile>
```

```
RESPONSE Code: 204
```

# Deleting Custom Recording

The following is an example of the DELETE request that deletes a User Template as represented by <usertemplateid>:

```
https://<connection_server>/vmrest/postgreetingrecordings/<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 custom.

| Field Name | Writable? | DB Value | Explanation/ Comments |
|---|---|---|---|
| ObjectId | Read-only | 36 Characters | ObjectId of the Device |
| DisplayName | Read/Write | 1-64 Characters | Friendly name for the Device like "Mobile Phone" |
| PostGreetingRecordingStreamFilesURI | Read-only | | List all the languages of custom recordings. |
| PostGreetingRecordingObjectId | Read-only | 36 Characters | List all the custom recordings |
| LanguageCode | Read-only | 4 characters | List all the locales/languages |
| StreamFile | Read/Write | 40 characters | Record Greetings |