



Cisco Unity Connection Telephony Interface (CUTI) API

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

CHAPTER 1

API Overview 1

- Introduction 1
- Benefits 2
- Technical Details 2
- Getting Started with CUTI 2
- Other CUTI Resources 2
- Troubleshooting 3

CHAPTER 2

Cisco Unity Connection Telephony Interface (CUTI) API -- Using CUTI for Basic Call Operations 5

- About Basic Call Operations 5
- Playing and Recording by Using a Call 6
- Examples 6
- Play Greetings 14
- Save Video Greetings 18
- Example of Data Fields 19

CHAPTER 3

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Users 21

- Voice Name Upload for Users 21
 - Add/Update the Voice Name for Users by Uploading a File from the Desktop 21
 - Add/Update Voice Name Recording the Greeting using Telephony Interface 23
 - Adding Voice Name by Passing Input Stream in the Request 25

CHAPTER 4

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Contacts 27

- Voice Name Upload for Contacts 27
 - Add/Update the Voice Name for Contacts by Uploading a File from the Desktop 27

Add/Update Voice Name Recording the Greeting using Telephony Interface 29

Adding Voice Name by Passing Input Stream in the Request 31

CHAPTER 5

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Distribution Lists 33

Voice Name Upload for Distribution Lists 33

Add/Update the Voice Name for Users by Uploading a File from the Desktop 33

Add/Update Voice Name Recording the Greeting using Telephony Interface 35

Adding Voice Name by Passing Input Stream in the Request 37

CHAPTER 6

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Call Handlers 39

Voice Name Upload for Call Handlers 39

Add/ Update the Voice Name for Call Handler by Uploading a File From the Desktop 39

Add/Update Voice Name Recording the Greeting using Telephony Interface 41

Adding Voice Name by Passing Input Stream in the Request 43

CHAPTER 7

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Directory Handlers 45

Voice Name Upload for Directory Handlers 45

Add/Update the Voice Name for Directory Handler by Uploading a File from the Desktop 45

Add/Update Voice Name Recording the Greeting using Telephony Interface 47

Adding Voice Name by Passing Input Stream in the Request 49

CHAPTER 8

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Interview Handlers 51

Voice Name Upload for Interview Handlers 51

Add/Update the Voice Name for Interview Handler by Uploading a File from the Desktop 51

Add/Update Voice Name Recording the Greeting using Telephony Interface 53

Adding Voice Name by Passing Input Stream in the Request 55



CHAPTER 1

API Overview

Links to Other API pages: Cisco_Unity_Connection_APIS

- [Introduction](#) , on page 1
- [Benefits](#), on page 2
- [Technical Details](#) , on page 2
- [Getting Started with CUTI](#) , on page 2
- [Other CUTI Resources](#) , on page 2
- [Troubleshooting](#) , on page 3

Introduction

The Cisco Unity Connection Telephony Interface (CUTI) API is a web service interface for telephone record and playback on Cisco Unity Connection systems. CUTI is designed to provide a simple, stable method of accessing telephone record and playback functionality on Connection systems through a standards-based interface using XML and HTTPS.

You can use CUTI to do the following:

- Initiate dialout to a phone device
- Play and record greetings, messages, and other audio
- Control playback speed and volume
- Stop and resume playback and record

Note: All the above functions associated with CUTI API support both the IPv4 and IPv6 addresses. However, the IPv6 address works only when Connection platform is configured in Dual (IPv4/IPv6) mode.

For more information see the chapter "Adding or Changing the IPv6 Addresses of Cisco Unity Connection 8.5 and Later Servers" at the following link

http://www.cisco.com/en/US/docs/voice_ip_comm/connection/9x/upgrade/guide/9xcucrug051.html.

Note: With Cisco Unity Connection 9.1(1), the single sign-on feature is enabled for all the Connection Rest APIs. For more information, see the "Single Sign-On in Cisco Unity Connection" chapter in Security Guide for Cisco Unity Connection 9.x

http://www.cisco.com/en/US/docs/voice_ip_comm/connection/9x/security/guide/9xcucsec061.html

Benefits

With CUTI, you have the ability to record audio as needed by using a phone, providing additional functionality besides uploading .wav files. You can use CUTI to do the following:

- As part of a User or Admin interface, record audio as part of a name, greeting, or message
- Play secure messages
- Record messages using phone as a media device

Technical Details

CUTI was developed using the latest advances in web-based interfaces: it is a REST interface that standardizes operations such as add, delete, and modify. The XML comes with standard XML schema definitions that are annotated with information about what is in them.

As a web-based interface, CUTI is independent of operating system and programming language, and does not require any client libraries to use.

Getting Started with CUTI

In order to begin developing with the Cisco Unity Connection CUTI API, you will need to obtain the following:

Hardware

- Cisco Media Convergence Server (MCS) for Cisco Unity Connection version 8.0 and later
- For detailed hardware requirements, see the Cisco Unity Connection 8.x Supported Platforms List

Software

- Cisco Unity Connection Software Ordering
- Not for Resale Kits (Must be eligible to purchase)
- Select Unified Communications System Release Kit

Discounts for some of the required hardware and software may be available for participants in the Cisco Technology Developer Program.

We recommend that all developers have an up-to-date Cisco Developer Services support agreement. This provides the developer with access to professional support and assistance for application development.

Other CUTI Resources

Additional information about CUTI is also available on the Cisco Developer Network ([link to CDN](#)). Note, however, that the documentation here on the DocWiki is the most up-to-date documentation available for CUTI.

To participate in the CUTI forum, see the CUTI forum on CDN

Troubleshooting

See the following for information on troubleshooting all Connection APIs:

[Troubleshooting \(applies to all Connection APIs\)](#)



CHAPTER 2

Cisco Unity Connection Telephony Interface (CUTI) API -- Using CUTI for Basic Call Operations

Links to Other API pages: [Cisco_Unity_Connection_APIS](#)

- [About Basic Call Operations](#) , on page 5
- [Playing and Recording by Using a Call](#) , on page 6
- [Examples](#), on page 6
- [Play Greetings](#), on page 14
- [Save Video Greetings](#) , on page 18
- [Example of Data Fields](#) , on page 19

About Basic Call Operations

The basic call operations are making a call, checking call-connected status, and hanging up. The call object is used to convey information about a call (see API XSD's). Cisco Unity Connection 10.5(2) and later facilitates you to make a video call. A user will be able to establish a video call, if all the required pre-checks are satisfied, which are as follows:

- Mapping video service accounts: subscriber needs a video service configured with the video service account.
- Enabling Class of Service (COS) Settings for the subscriber

When making a call, the following fields are used:

- **number**: the only field that does not have a default value. Users with the system administrator role can call any number, but end users can call only those numbers that do not violate their associated restriction table, based on their class of service.
- **maximumRings**: defaults to 4.
- **mediaSwitchObjectId**: for end users, defaults to the phone system that is associated with the user. For users who do not have an associated phone system (for example, administrators), the phone system that is marked as the default TRAP phone system is used.
- **calltype**: indicates whether the call type is audio or video. If you do not specify the call type, then by default the call will be placed as an audio call only.

When a call has been placed, a client can get server information about the call, most importantly the connected status.

The following are HTTP commands for basic call operations:

POST http://<server>/calls:

Create a call (make a phone call) by using the information in a call object.
Returns a URI for the new call.

GET http://<server>/calls/<call id>: Get basic call information. Returns a call object.

DELETE http://<server>/calls/<call id>: Hangs up a call.

Playing and Recording by Using a Call

Files on the server can be played and recorded by phone (using the call control operations) by using the CallControl object (see API XSD's).

Note: Video record and play do not support the pause and resume operations.

Recording or Playing message can be paused and resumed (using the call control operations). The fields of the CallControl object are described below:

- **opType**: PLAY, RECORD, PAUSE, RESUME or STOP
- **resourceType**: MESSAGE, BROADCASTMESSAGE, STREAM, HANDLER, or URI. STREAM is generally a temporary resource created by a record operation.
- **resourceID**: Can be a message identifier, a broadcast message identifier or a stream identifier depending on the resource type.
- **Speed**, **volume** and **startPosition** are all optional parameters. Speed and volume are a percentage and default to 100. The start position indicates a time in milliseconds and defaults to 0.
- **lastResult**: The result of the last call control operation.
- **folderType**: Reserved. Only messages in the inbox can be played.
- **sessionId**: This is a reference of the video recording on the MediaSense Server using Record API. MediaSense uniquely identifies the recording using the sessionId.

A call control object is passed to a call by using a POST to the resource URI of the call:



Note

Max time limit for a paused call to be resumed will be idle time out.

```
POST http://<server>/vmrest/calls/<call id>
```

Examples

The examples below show how to: make a call, check call status, record by phone, play the recording, pause and resume the recording, hang up, and send the recording as a message. Each step is shown as the HTTP request and response.

Make the call

```
POST /vmrest/calls HTTP/1.1
Content-Type: application/json

Accept: application/json

User-Agent: Java/1.6.0_17

Host: cuc-install-67.cisco.com

Connection: keep-alive

Authorization: Basic
Y2NtYWRtaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 36

{"number":"2119","maximumRings":"4"}
```

Response:

```
HTTP/1.1 201 Created
Set-Cookie:
JSESSIONIDSSO=CB2BE1D9771621DBEA24B479C1A9C10A; Path=/
Set-Cookie:
JSESSIONID=EB7C33B4DC0AAE91CB002D799C8734BA; Path=/vmrest
Location:
[http://cuc-install-67.cisco.com/vmrest/calls/vmrest/calls/1]
Content-Type: application/json
Transfer-Encoding: chunked

Date: Fri, 15 Jan 2010 15:14:11 GMT
Server:
```

Make the Video call

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

```
<Call>
  <number>1096</number>
  <maximumRings>5</maximumRings>
  <callType>Video</callType>
</Call>
```

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/calls/66
CallType:video
```

JSON Example

```
POST /vmrest/calls HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic
Y2NtYWRtaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 36
{"number":"1097","maximumRings":"4","callType":"video"}
```

Response:

```
HTTP/1.1 201 Created
Set-Cookie:
JSESSIONIDSSO=CB2BE1D9771621DBEA24B479C1A9C10A; Path=/
Set-Cookie:
JSESSIONID=EB7C33B4DC0AAE91CB002D799C8734BA; Path=/vmrest
Location:
[http://cuc-install-67.cisco.com/vmrest/calls/vmrest/calls/40/calltype:video]
Content-Type: application/json
```

Video Call Downgrade Scenarios

If a Video call gets downgraded to audio, user receives the following response:

Scenario 1

If a user does not have Video Service Account and try to make a video call, the following response is received:

```
Response Code: 201
/vmrest/calls/66
CallType: Audio
CallDowngradeReason: Video Service Account does not exist
```

Scenario 2

If a user does not have Video Service activated on Unity Connection and try to make a video call, the following response is received:

```
Response Code: 201
/vmrest/calls/66
CallType: Audio
CallDowngradeReason: Video Service not found
```

Scenario 3

If a user does not have Video enabled in Class of Service and try to make a video call, the following response is received:

```
Response Code: 201
/vmrest/calls/66
CallType: Audio
CallDowngradeReason: Not allowed to Playback and Record Video Greetings
```

Check call status

```
GET /vmrest/calls/1 HTTP/1.1
Accept: application/json

User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com

Connection: keep-alive
authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
```

Response:

```

HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=849167D3D61D2F30BA278BBF949FDF0F; Path=/
Set-Cookie: JSESSIONID=28A69142E490EC63A56690F9811BDDF9; Path=/vmrest
Content-Type: application/json
{"id":"1","connected":"true","nativeConnectionId":"2"}

```

Check Video call status

```
GET https://<connection-server>/vmrest/calls/<call-id>
```

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

```
Response Code: 200
```

```

<Call>
  <number>1007</number>
  <id>1105</id>
  <connected>true</connected>
  <callType>video</callType>
  <nativeConnectionId>16</nativeConnectionId>
</Call>

```

JSON Example

```

GET /vmrest/calls/1 HTTP/1.1
Accept: application/json

User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com

Connection: keep-alive
authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==

```

Response:

```

HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Expires: Wed, 31 Dec 1969 16:00:00 PST
Set-Cookie: JSESSIONIDSSO=849167D3D61D2F30BA278BBF949FDF0F; Path=/
Set-Cookie: JSESSIONID=28A69142E490EC63A56690F9811BDDF9; Path=/vmrest
Content-Type: application/json
{"number":"1097","id":"40","connected":"true","callType":"video","nativeConnectionId":"45"}

```

Record by phone

```

POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 15
{"op":"RECORD"}

```

Response:

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=8279B639488165EFFAC8D330FD2A1281; Path=/
Set-Cookie: JSESSIONID=AA008CFCC9B329EA961CD1E660384785; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:13 GMT
Server:
{"op": "RECORD", "resourceType": "STREAM", "resourceId": "cf1cb014-6394-4279-ab5a-74a6d680e440.wav",
"lastResult": "0", "speed": "100", "volume": "100", "startPosition": "0"}
```

Video Recording by phone

```
POST https://<connection-server>/vmrest/calls/<call-id>
```

```
<CallControl>
<op>RECORD</op>
</CallControl>
```

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

```
Response Code: 200
```

```
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>9d59b180-6147-4796-9488-014ef73203a0.wav</resourceId>
  <sessionId>3514637e02b471</sessionId>
  <lastResult>0</lastResult>
</CallControl>
```

JSON Example

```
POST /vmrest/calls/<call-id> HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 15
{"op": "RECORD"}
```

Response:

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=8279B639488165EFFAC8D330FD2A1281; Path=/
Set-Cookie: JSESSIONID=AA008CFCC9B329EA961CD1E660384785; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:13 GMT
Server:
1({"op": "RECORD", "resourceType": "STREAM", "resourceId": "a899ac7-9ee2-4c75-b3a7-e6a2a5d6d2dc.wav", "sessionId": "1e14628d8a9ac1", "lastResult": "0"})
```

Pause the recording

```
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: ucbu-aricent-vm5.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
{"op":"PAUSE"}
```

Response:

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=6A4C99C02E79D696A2DEC46C24A0B2DA; Path=/; Secure
Set-Cookie: JSESSIONID=1E642B010F5AD5630DE922B1378679DF; Path=/vmrest; Secure
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 01 Jun 2012 06:33:49 GMT
Server:

{"op":"PAUSE","lastResult":"0","speed":"100","volume":"100","startPosition":"0"}
```

Resume the recording

```
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: ucbu-aricent-vm5.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
{"op":"RESUME"}
```

Response:

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=687E20DEFCC984797342E37A656A1F32; Path=/; Secure
Set-Cookie: JSESSIONID=48D31DBFCBAEF586F2A9398AB11204CC; Path=/vmrest; Secure
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 01 Jun 2012 06:34:25 GMT
Server:

{"op":"RESUME","lastResult":"0","speed":"100","volume":"100","startPosition":"0"}
```

Play the recording

```
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"STREAM","resourceId":"cf1cb014-6394-4279-ab5a-74a6d680e440.wav",
"lastResult":"0","speed":"100","volume":"100","startPosition":"0"}
```

Response:

```

HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:

{"op":"PLAY","resourceType":"STREAM","resourceId":"cf1cb014-6394-4279-ab5a-74a6d680e440.wav",
"lastResult":"0","speed":"100","volume":"100","startPosition":"0"}

```

Play Video recording

In a video call,

- For STREAM or URI resource type, if the Session ID is available, then the video recording will be played, else the audio recording will be played from the Audio resource-ID along with an image.
- For Message or Broadcast Message resource type, the audio recording will be played from the Audio resource-ID along with an image.
- If both sessionID and resource ID are not available, an error message is send to the user.

```
POST https://<connection-server>/vmrest/calls/<call-id>
```

```

Request:
<CallControl>
  <op>PLAY</op>
  <resourceType>STREAM</resourceType>
  <resourceId>1218f93a-2ce2-4987-9589-f8e06bd54a14.wav</resourceId>
  <sessionId>2f61469411d7131</sessionId>
  <lastResult>0</lastResult>
</CallControl>

```

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

```

Response: 200 Ok
<CallControl>
  <op>PLAY</op>
  <resourceType>STREAM</resourceType>
  <resourceId>1218f93a-2ce2-4987-9589-f8e06bd54a14.wav</resourceId>
  <sessionId>2f61469411d7131</sessionId>
  <lastResult>0</lastResult>
</CallControl>

```

JSON Example

```

Request:
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"STREAM","resourceId":"1218f93a-2ce2-4987-9589-f8e06bd54a14.wav","sessionId":"2f61469411d7131","lastResult":"0"}

```



```

Response:
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:

1(("op":"PLAY","resourceType":"STREAM","resourceId":"1218f93a-2ae2-4987-9589-f8e06cd54a14.wav","sessionId":"2f61469411d7131","lastResult":"0"))

```

Hang up

```

DELETE /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==

```

Response:

```

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=93B286C6584533F9EE793A01E5804465; Path=/
Set-Cookie: JSESSIONID=0CCFFFCAD1C11A45EAD1E8F04B22D28D; Path=/vmrest
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:

```

Send the recording as a message

```

POST /vmrest/messages?userobjectid=84c14db9-7439-4326-a2e2-e516aa192dff HTTP/1.1
Content-Type: multipart/form-data; boundary=Boundary_1_16617866_1263568442453
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 731

--Boundary_1_16617866_1263568442453
Content-Type: application/json
{"Subject":"send message test","ArrivalTime":"0","FromSub":"false"}
--Boundary_1_16617866_1263568442453
Content-Type: application/json

{"Recipient":{"Type":"TO","Address":
{"UserGuid":"84c14db9-7439-4326-a2e2-e516aa192dff","DisplayName":"Operator"}
}}
--Boundary_1_16617866_1263568442453
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><CallControl><op>PLAY</op><resourceType>STREAM</resourceType><resourceId>cf1cb014-6394-4279-
a6fa-746d80e440.wav</resourceId><lastResult>0</lastResult><speed>100</speed><volume>100</volume><startPosition>0</startPosition></CallControl>

--Boundary_1_16617866_1263568442453--

```

Response:

```

HTTP/1.1 202 Accepted
Set-Cookie: JSESSIONIDSSO=3D875CAE50B6B4947DEDCE5EDA119922; Path=/
Set-Cookie: JSESSIONID=842B1A6E53DAE329F6DA7525951668B4; Path=/vmrest
Content-Type: application/json
Content-Length: 0
Date: Fri, 15 Jan 2010 15:14:16 GMT
Server:

```

Send the recording as a Video message

```

POST /vmrest/messages?userobjectid=84c14db9-7439-4326-a2e2-e516aa192dff HTTP/1.1
Content-Type: multipart/form-data;boundary=Boundary_1_16617866_1263568442453
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 731

--Boundary_1_16617866_1263568442453
Content-Type: application/json
{"Subject":"send message test","ArrivalTime":"0","FromSub":"false"}
--Boundary_1_16617866_1263568442453
Content-Type: application/json

{"Recipient":{"Type":"TO","Address":
{"UserGuid":"84c14db9-7439-4326-a2e2-e516aa192dff","DisplayName":"Operator"}
}}
--Boundary_1_16617866_1263568442453
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><CallControl><op>PLAY</op><resourceType>STREAM</resourceType><resourceId>cf1cb014-6394-4279-
afa-746680e440.wav</resourceId><sessionId><lastResult>0</lastResult><speed>100</speed><volume>100</volume><startPosition>0</startPosition></CallControl>

--Boundary_1_16617866_1263568442453--

```

Play Greetings

Play Audio Greetings

Audio Greeting of users, user templates, call handler can be played using CUTI API. a) For a user with mailbox following should be the request:

```

Request:
POST https://<connection-server>/vmrest/calls/<call-id>
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
</CallControl>

```

```

Response:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
  <lastResult>0</lastResult>
</CallControl>
Response code:200    Ok

```

JSON Example

```

Request
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"HANDLER","resourceId":"0bc5155b-0989-4523-a75a-948cc691b99e.wav","lastResult":0,"speed":100,"volume":100,"startPosition":0}

```

```

Response: HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:
1({"op":"PLAY","resourceType":"HANDLER","resourceId":"0bc5155b-0989-4523-a75a-948cc691b99e.wav","lastResult":0,"speed":100,"volume":100,"startPosition":0})

```

For a System administrator following should be the request:

In this case greeting of handler whose objectId "handlerObjectId" is passed in request will be played.

```
POST https://<connection-server>/vmrest/calls/<call-id>?handlerObjectId=<handlerObjectId>
```

```

Request:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
</CallControl>

```

```

Response:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
  <lastResult>0</lastResult>
</CallControl>
Response code:200    Ok

```

JSON Example

```
Request:
POST /vmrest/calls/1 ? handlerObjectId=<handlerObjectId>
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-2ce2-4987-9589-f8e06bd54a14.wav"}
```

```
Response:
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:
1({"op":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-2ce2-4987-9589-f8e06bd54a14.wav","lastResult":"0","speed":"100","volume":"100","startPosition":"0"})
```

Play Video Greetings

Video greeting of users, user templates, handler can be played using CUTI API. a) For a user with video account following should be the request:

```
Request:
POST https://<connection-server>/vmrest/calls/<call-id>
```

```
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <sessionId>55d146ced50d1</sessionId>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <lastResult>0</lastResult>
</CallControl>
```

```
Response:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <sessionId>55d146ced50d1</sessionId>
  <lastResult>0</lastResult>
</CallControl>
Response code:200   Ok
```

JSON Example

```
Request:
POST /vmrest/calls/1 HTTP/1.1
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-20e2-4987-9589-f8e060d54a14.wav","sessionId":"2f61469411d7131"}
```

```
Response:
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:

1({"op":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-20e2-4987-9589-f8e060d54a14.wav","sessionId":"2f61469411d7131","lastResult":"0"})
```

For a System administrator following should be the request: In this case video greeting of handler, whose objectId "handlerObjectId" is passed in request will be played.

```
POST https://<connection-server>/vmrest/calls/<call-id>?handlerObjectId=<handlerObjectId>

Request:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <sessionId>55d146ced50d1</sessionId>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <lastResult>0</lastResult>
</CallControl>
Response:
<CallControl>
  <op>PLAY</op>
  <resourceType>HANDLER</resourceType>
  <resourceId>0bc5155b-0989-4523-a75a-948cc691b99e.wav</resourceId>
  <sessionId>55d146ced50d1</sessionId>
  <lastResult>0</lastResult>
</CallControl>
Response code:200   Ok
```

JSON Example

```
Request:
POST /vmrest/calls/1 ? handlerObjectId=<handlerObjectId>
Content-Type: application/json
Accept: application/json
User-Agent: Java/1.6.0_17
Host: cuc-install-67.cisco.com
Connection: keep-alive
Authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==
Content-Length: 159

{"op":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-20e2-4987-9589-f8e060d54a14.wav","sessionId":"2f61469411d7131"}
```

```

Response:
HTTP/1.1 200 OK
Set-Cookie: JSESSIONIDSSO=CD7B1064CEB9385B6FE9279ECBDB40E3; Path=/
Set-Cookie: JSESSIONID=201BD26E840AA2CEEC242A9CCD0FE8F6; Path=/vmrest
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 15 Jan 2010 15:14:15 GMT
Server:

1(("q":"PLAY","resourceType":"HANDLER","resourceId":"1218f93a-2e2-4987-9589-f8e06cc54e14.wav","sessionId":"2f61469411d7131","lastResult":"0"))

```

Add new Resource Type: HANDLER in section "Playing and Recording by Using a Call"

Save Video Greetings

Unity Connection allows you to save video greetings using both GET and PUT requests.

Example of GET Request

```

GET http://<connection-server>/vmrest/handlers/callhandlers/<call handler object
ID>/greetings/<greeting type>/greetingstreamfiles/<language code>/video

```

The following is the response of the above GET command and the output may vary depending on your inputs.

```

Response: 200
<CallControl>
  <resourceId>aad91d6d-aeca-4a72-8069-b656efb3041f.wav</resourceId>
  <sessionId>570146ed1504cb1</sessionId>
</CallControl>

```

JSON Example

```

Request
GET
vmrest/handlers/callhandlers/30600b21-1a4c-47a3-a078-8078984e5376/greetings/Standard/greetingstreamfiles/1033/video
Accept: application/json
User-Agent: Java/1.6.0_17
Host: <connection-server>
Connection: keep-alive
authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg=

```

```

Response
HTTP/1.1 201
Content-Type: application/json
Date: Fri, 15 Jan 2010 15:14:11 GMT
Server:
{ "resourceId" : " aad91d6d-aeca-4a72-8069-b656efb3041f.wav", "sessionId" : "570146ed1504cb1"}

```

Example of PUT Request

```

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

```

```

<CallControl>
  <resourceId>aad91d6d-aeca-4a72-8069-b656efb3041f.wav</resourceId>
  <sessionId>570146ed1504cb1</sessionId>
</CallControl>

```

Response: 201 OK

JSON Example

```
Request
PUT
vmrest/handlers/callhandlers/30600b21-1a4c-47a3-a078-8078984e5376/greetings/Standard/greetingstreamfiles/1033/video
Content-Type: application/json
Accept: application/json
Host: <connection-server>
Connection: keep-alive
authorization: Basic Y2NtYWRTaW5pc3RyYXRvcjplY3NidWxhYg==

{ "resourceId" : " aad91d6d-aeca-4a72-8069-b656efb3041f.wav", "sessionId" : "570146ed1504cb1" }
```

```
Response :
HTTP/1.1 201
Content-Type: application/json
Date: Fri, 15 Jan 2010 15:14:11 GMT
Server:
```

Example of Data Fields

Field Name	Data Type	Operation	Description
number	Integer	Read/Write Only	The user extension where the caller want to place a call.
maximumRings	Integer	Read/Write	The maximum number of phone rings when a call is placed and if the call is not picked, the caller will get ring-no-answer message. Default value: 4.
mediaSwitchObjectid	String	Read Only	The unique identifier of the MediaSwitch object that Cisco Unity Connection uses for placing the call to the subscriber phone.
callType	String	Read/Write	Displays whether the call type is audio or video.
callId	Integer	Read Only	The unique identifier of the call that Cisco Unity Connection uses for placing the call to the subscriber phone.
connected	Boolean	Read Only	Displays whether the subscriber is connected to a call or not.
nativeConnectionId	Integer	Read Only	The unique identifier of the call created by Unity Connection.
sessionId	String	Read/Write	This is a reference of the video recording on the MediaSense Server using Record API. MediaSense uniquely identifies the recording using the sessionId.

Example of Data Fields



CHAPTER 3

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Users

Links to Other API pages: [Cisco_Unity_Connection_APIs](#)

- [Voice Name Upload for Users, on page 21](#)

Voice Name Upload for Users

This API is used to upload voice name to the Users. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/Update the Voice Name for Users by Uploading a File from the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type: application/json
Connection: keep-alive
```

Add/Update the Voice Name for Users by Uploading a File from the Desktop

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

The content has been accepted and copied into the temporary file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type: application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3 Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>
Request Body:
<Distributionlists>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</Distributionlists>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/distributionlists/<distributionlistobjectid>
Accept: application/json
Content-Type: application/json
Connection: keep-alive
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```

```
Response Code: 201
```

Use the below URL in the browser to listen to the voice name.

<https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

```
GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}
```

Response Code: 200

Use the following request to record the voice name:

```
POST https://<connection-server>/vmrest/users/<userobjectId>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>
```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```
POST https://<connection-server>/vmrest/ users/<userobjectId>
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD"
}
```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the user handler voice name.

```
PUT https://<connection-server>/vmrest/users/<uerobjectid>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
}
```

Response Code: 204

JSON Example

```
POST https://<connection-server>/vmrest/ users/<userobjectid>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the interview handler: Paste the URL in the browser and listen to the voice name. <https://<connection-server>/vmrest/Users/<userobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be: <https://<connection-server>/vmrest/users/<userobjectid>/voicename> The request body should be like this: `put.setRequestBody(new FileInputStream(file3));` where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the user handler: <https://<connection-server>/vmrest/Users/<userobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request



CHAPTER 4

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Contacts

- [Voice Name Upload for Contacts, on page 27](#)
- [Add/Update the Voice Name for Contacts by Uploading a File from the Desktop, on page 27](#)
- [Add/Update Voice Name Recording the Greeting using Telephony Interface, on page 29](#)
- [Adding Voice Name by Passing Input Stream in the Request, on page 31](#)

Voice Name Upload for Contacts

This API is used to upload voice name to the Contacts. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/Update the Voice Name for Contacts by Uploading a File from the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note Rename the .wav file which is to be uploaded as the greeting to the temporary file name generated as part of the response from step1. For eg : If the file on desktop has name "greeting.wav" and the temporary file name is 0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav(got as part of the response of step1), then the file should be renamed to 0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav before step 2 is performed. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3 Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/contacts/<contactobjectid>
Request Body:
<Contacts>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</Contacts>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/contacts/<contactobjectid>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
Request Body:
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```


Response Code: 201

Use the below URL in the browser to listen to the voice name.

<https://<connection-server>/vmrest/contacts/<contactobjectid>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

It's a three step process to record a new file then modify the current stream with this new stream

Step 1 Call Connection

In the first step, the integration between Unity Connection and Call Manager must be setup so that a call can be setup. Refer to the document at the below link to check how to make the call.

[http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_\(CUTI\)_API_-_Using_CUTI_for_Basic_Call_Operations #](http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_(CUTI)_API_-_Using_CUTI_for_Basic_Call_Operations #)

Step 2 Recording

Once the phone is answered, the second step is to record the greeting. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds. The minimum and maximum length of the recording using telephony integration can be seen in Connection Administration under the Advanced >> Telephony settings.

- Maximum Recording Time in Milliseconds – default value 1200000
- Minimum Recording Duration in Milliseconds -default value 1000

The same can be fetched using APIs using the following URL:

<https://<connection-server>/vmrest/configurationvalues>

Check for the values

The request is as follows:

```
<ConfigurationValue>
  <Type>3</Type>
  <LastModifiedTime>2013-01-21 07:21:53.49</LastModifiedTime>
  <LastModifiedByComponent>CUADMIN</LastModifiedByComponent>
  <FullName>System.Telephony.RecordingMinimumLengthMs</FullName>
  <Value>1000</Value>
  <UserSetting>true</UserSetting>
  <MinVal>0</MinVal>
  <MaxVal>5000</MaxVal>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>
```

Configuration values can be modified using Connection Administration..

JSON Example

```

GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}

```

Use the following request to record the voice name:

```

POST https://<connection-server>/vmrest/contacts/<contactobjectid>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>

```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```

POST https://<connection-server>/vmrest/contacts/<contactobjectid>
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD"
}

```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the contact voice name.

```
PUT https://<connection-server>/vmrest/contacts/<ContactobjectId>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
```

Response Code: 204

JSON Example

```
POST https://<connection-server>/vmrest/contacts/<contactobjectId>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the interview handler: Paste the URL in the browser and listen to the voice name. <https://<connection-server>/vmrest/Contacts/<contactobjectId>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be:

<https://<connection-server>/vmrest/contacts/<contactobjectId>/voicename>

The request body should be like this:

```
put.setRequestBody(new FileInputStream(file3));
```

where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the contacts:

<https://<connection-server>/vmrest/Contacts/<contactobjectId>/voicename>

Adding Voice Name by Passing Input Stream in the Request



CHAPTER 5

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Distribution Lists

Links to Other API pages: [Cisco_Unity_Connection_APIs](#)

- [Voice Name Upload for Distribution Lists, on page 33](#)

Voice Name Upload for Distribution Lists

This API is used to upload voice name to the distribution lists. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/Update the Voice Name for Users by Uploading a File from the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

JSON Example

Add/Update the Voice Name for Users by Uploading a File from the Desktop

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

The content has been accepted and copied into the temporary file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3 Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>
Request Body:
<Distributionlists>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</Distributionlists>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/distributionlists/<distributionlistobjectid>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```

```
Response Code: 201
```

Use the below URL in the browser to listen to the voice name.

<https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

It's a three step process to record a new file then modify the current stream with this new stream

Step 1 Call Connection

In the first step, the integration between Unity Connection and Call Manager must be setup so that a call can be setup. Refer to the document at the below link to check how to make the call.

[http://docwiki.cisco.com/wiki/Cisco_Utility_Connection_Telephony_Interface_\(CUTI\)_API_-_Using_CUTI_for_Basic_Call_Operations](http://docwiki.cisco.com/wiki/Cisco_Utility_Connection_Telephony_Interface_(CUTI)_API_-_Using_CUTI_for_Basic_Call_Operations)

Step 2 Recording

Once the phone is answered, the second step is to record the greeting. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds. The minimum and maximum length of the recording using telephony integration can be seen in Connection Administration under the Advanced >> Telephony settings.

- Maximum Recording Time in Milliseconds – default value 1200000
- Minimum Recording Duration in Milliseconds -default value 1000

The same can be fetched using APIs using the following URL:

<https://<connection-server>/vmrest/configurationvalues>

Check for the values

The request is as follows:

```
<ConfigurationValue>
  <Type>3</Type>
  <LastModifiedTime>2013-01-21 07:21:53.49</LastModifiedTime>
  <LastModifiedByComponent>CUADMIN</LastModifiedByComponent>
  <FullName>System.Telephony.RecordingMinimumLengthMs</FullName>
  <Value>1000</Value>
  <UserSetting>true</UserSetting>
  <MinVal>0</MinVal>
  <MaxVal>5000</MaxVal>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>
```

Response Code: 200

Configuration values can be modified using Connection Administration..

JSON Example

Add/Update Voice Name Recording the Greeting using Telephony Interface

```
GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}
```

Response Code: 200

Use the following request to record the voice name:

```
POST https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>
```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```
POST https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD"
}
```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the distribution list voice name.


```
PUT https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
```

Response Code: 204

JSON Example

```
POST https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the interview handler: Paste the URL in the browser and listen to the voice name.

<https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be:

<https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename>

The request body should be like this:

```
put.setRequestBody(new FileInputStream(file3));
```

where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the distribution list:

<https://<connection-server>/vmrest/distributionlists/<distributionlistobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request



CHAPTER 6

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Call Handlers

Links to Other API pages: [Cisco_Unity_Connection_APIs](#)

- [Voice Name Upload for Call Handlers, on page 39](#)

Voice Name Upload for Call Handlers

This API is used to upload voice name to the Call Handlers. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/ Update the Voice Name for Call Handler by Uploading a File From the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

Add/ Update the Voice Name for Call Handler by Uploading a File From the Desktop

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

The content has been accepted and copied into the temporary file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3 Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>
Request Body:
<CallHandler>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</CallHandler>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/handlers/callhandlers/<callhandlerobjectid>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
Request Body:
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```

```
Response Code: 201
```

Use the below URL in the browser to listen to the voice name.

<https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectId>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

It's a three step process to record a new file then modify the current stream with this new stream

Step 1 Call Connection

In the first step, the integration between Unity Connection and Call Manager must be setup so that a call can be setup. Refer to the document at the below link to check how to make the call.

[http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_\(CUTI\)_API_-_Using_CUTI_for_Basic_Call_Operations](http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_(CUTI)_API_-_Using_CUTI_for_Basic_Call_Operations)

Step 2 Recording

Once the phone is answered, the second step is to record the greeting. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds. The minimum and maximum length of the recording using telephony integration can be seen in Connection Administration under the Advanced >> Telephony settings.

- Maximum Recording Time in Milliseconds – default value 1200000
- Minimum Recording Duration in Milliseconds -default value 1000

The same can be fetched using APIs using the following URL:

<https://<connection-server>/vmrest/configurationvalues>

Check for the values

The request is as follows:

```
<ConfigurationValue>
  <Type>3</Type>
  <LastModifiedTime>2013-01-21 07:21:53.49</LastModifiedTime>
  <LastModifiedByComponent>CUADMIN</LastModifiedByComponent>
  <FullName>System.Telephony.RecordingMinimumLengthMs</FullName>
  <Value>1000</Value>
  <UserSetting>true</UserSetting>
  <MinVal>0</MinVal>
  <MaxVal>5000</MaxVal>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>
```

Response Code: 200

Configuration values can be modified using Connection Administration..

JSON Example

Add/Update Voice Name Recording the Greeting using Telephony Interface

```
GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}
```

Response Code: 200

Use the following request to record the voice name:

```
POST https://<connection-server>/vmrest/calls/<CallId>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>
```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```
POST https://<connection-server>/vmrest/calls/<CallId>
Accept application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "op": "RECORD"
}
```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the call handler voice name.

```
PUT https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
```

Response Code: 204

JSON Example

```
POST https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the call handler: Paste the URL in the browser and listen to the voice name.

<https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be:

<https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/voicename>

The request body should be like this:

```
put.setRequestBody(new FileInputStream(file3));
```

where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the call handler:

<https://<connection-server>/vmrest/handlers/callhandlers/<callhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request



CHAPTER 7

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Directory Handlers

Links to Other API pages: [Cisco_Unity_Connection_APIS](#)

- [Voice Name Upload for Directory Handlers, on page 45](#)

Voice Name Upload for Directory Handlers

This API is used to upload voice name to the Directory Handler. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/Update the Voice Name for Directory Handler by Uploading a File from the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

The content has been accepted and copied into the temporary file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3 Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>
Request Body:
<DirectoryHandler>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</DirectoryHandler>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
Request Body:
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```

```
Response Code: 201
```

Use the below URL in the browser to listen to the voice name.

<https://<connection-server>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

It's a three step process to record a new file then modify the current stream with this new stream

Step 1 Call Connection

In the first step, the integration between Unity Connection and Call Manager must be setup so that a call can be setup. Refer to the document at the below link to check how to make the call.

[http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_\(CUTI\)_API_-_Using_CUTI_for_Basic_Call_Operations](http://docwiki.cisco.com/wiki/Cisco_Unity_Connection_Telephony_Interface_(CUTI)_API_-_Using_CUTI_for_Basic_Call_Operations)

Step 2 Recording

Once the phone is answered, the second step is to record the greeting. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds. The minimum and maximum length of the recording using telephony integration can be seen in Connection Administration under the Advanced >> Telephony settings.

- Maximum Recording Time in Milliseconds – default value 1200000
- Minimum Recording Duration in Milliseconds -default value 1000

The same can be fetched using APIs using the following URL:

<https://<connection-server>/vmrest/configurationvalues>

Check for the values

The request is as follows:

```
<ConfigurationValue>
  <Type>3</Type>
  <LastModifiedTime>2013-01-21 07:21:53.49</LastModifiedTime>
  <LastModifiedByComponent>CUADMIN</LastModifiedByComponent>
  <FullName>System.Telephony.RecordingMinimumLengthMs</FullName>
  <Value>1000</Value>
  <UserSetting>true</UserSetting>
  <MinVal>0</MinVal>
  <MaxVal>5000</MaxVal>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>
```

Response Code: 200

Configuration values can be modified using Connection Administration..

JSON Example

Add/Update Voice Name Recording the Greeting using Telephony Interface

```
GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}
```

Response Code: 200

Use the following request to record the voice name:

```
POST https://<connection-server>/vmrest/calls/<CallId>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>
```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```
POST https://<connection-server>/vmrest/calls/<CallId>
Accept application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "op": "RECORD"
}
```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the call handler voice name.

```
PUT https://<connectionserver>/
vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
```

Response Code: 204

JSON Example

```
POST https://<connectionserver>/
vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the directory handler: Paste the URL in the browser and listen to the voice name.

<https://<connection-server>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be:

<https://<connection-server>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename> The request body should be like this:

```
put.setRequestBody(new FileInputStream(file3));
```

where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the directory handler:

<https://<connection-server>/vmrest/handlers/directoryhandlers/<directoryhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request



CHAPTER 8

Cisco Unity Connection Provisioning Interface (CUTI) API -- Voice Name Upload for Interview Handlers

- [Voice Name Upload for Interview Handlers, on page 51](#)

Voice Name Upload for Interview Handlers

This API is used to upload voice name to the Interview Handlers. There are three ways of uploading the voice name:

- 1. Upload a .wav file from the desktop.
- 2. Record using CUTI(Cisco Unity Telephony Interface) and then upload the recording.
- 3. Pass the .wav file as the input stream to upload the voice name.

Add/Update the Voice Name for Interview Handler by Uploading a File from the Desktop

It is a 3 step process:

Step 1

A placeholder for the WAV file must be created with a POST request. This is a temporary file placeholder 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 is automatically cleaned.

The request is as follows:

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

```
Response Code: 201
```

The content will be the name of the newly created temporary .wav file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 201
0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav
```

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.

Note The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds.

The request is as follows:

```
PUT https://<connection-server>/vmrest/voicefiles/<temporary file name>
```

```
Response Code: 204
```

The content has been accepted and copied into the temporary file.

JSON Example

```
POST https://<connection-sever>/vmrest/voicefiles/<temporary file name>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
```

```
Response Code: 204
```

Step 3

Add the file to the voice name. The request is as follows:

```
PUT https://<connection-server>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>
Request Body:
<InterviewHandler>
  <VoiceName>0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav</VoiceName>
</InterviewHandler>
```

```
Response Code: 201
```

JSON Example

```
POST https://<connection-sever>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>
Accept: application/json
Content-Type; application/json
Connection: keep-alive
Request Body:
{
  "VoiceName": "0ad46c53-44eb-4da6-988c-4f07fab6bdd.wav"
}
```

```
Response Code: 201
```

Use the below URL in the browser to listen to the voice name.

<https://<connectionserver>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename>

Add/Update Voice Name Recording the Greeting using Telephony Interface

It's a three step process to record a new file then modify the current stream with this new stream

Step 1 Call Connection

In the first step, the integration between Unity Connection and Call Manager must be setup so that a call can be setup. Refer to the document at the below link to check how to make the call.

[http://docwiki.cisco.com/wiki/Cisco_Untity_Connection_Telephony_Interface_\(CUTI\)_API_-_Using_CUTI_for_Basic_Call_Operations sp](http://docwiki.cisco.com/wiki/Cisco_Untity_Connection_Telephony_Interface_(CUTI)_API_-_Using_CUTI_for_Basic_Call_Operations sp)

Step 2 Recording

Once the phone is answered, the second step is to record the greeting. The length of the greeting can be set under the System Settings > General Configuration settings. Here you can enter the maximum length for system call handler greetings. The range is 1 to 1,200 seconds and default setting is 90 seconds. The minimum and maximum length of the recording using telephony integration can be seen in Connection Administration under the Advanced >> Telephony settings.

- Maximum Recording Time in Milliseconds – default value 1200000
- Minimum Recording Duration in Milliseconds -default value 1000

The same can be fetched using APIs using the following URL:

<https://<connection-server>/vmrest/configurationvalues>

Check for the values

The request is as follows:

```
<ConfigurationValue>
  <Type>3</Type>
  <LastModifiedTime>2013-01-21 07:21:53.49</LastModifiedTime>
  <LastModifiedByComponent>CUADMIN</LastModifiedByComponent>
  <FullName>System.Telephony.RecordingMinimumLengthMs</FullName>
  <Value>1000</Value>
  <UserSetting>true</UserSetting>
  <MinVal>0</MinVal>
  <MaxVal>5000</MaxVal>
  <RequiresRestart>false</RequiresRestart>
</ConfigurationValue>
```

Response Code: 200

Configuration values can be modified using Connection Administration..

JSON Example

```
GET https://<connection-server>/vmrest/configurationvalues
Accept : application/json
Connection : keep-alive
{
  "@total": "2",
  "ConfigurationValue":
  [
    {
      "Type": "1",
      "FullName": "System",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    },
    {
      "Type": "1",
      "FullName": "System.Notifier",
      "UserSetting": "false",
      "MinVal": "0",
      "MaxVal": "0",
      "RequiresRestart": "false"
    }
  ]
}
```

Response Code: 200

Use the following request to record the voice name:

```
POST https://<connection-server>/vmrest/calls/<CallId>
Request Body:
<CallControl>
  <op>RECORD</op>
</CallControl>
```

Response Code: 201

Make a note of the output obtained, that will be the input for uploading a wave file.

JSON Example

```
POST https://<connection-server>/vmrest/calls/<CallId>
Accept application/json
Content-Type: application/json
Connection: keep-alive
Request Body:
{
  "op": "RECORD"
}
```

Response Code: 201

Step 3 Upload the .WAV File

The third step is to upload the wave file to the call handler voice name.

```
PUT https://<connectionserver>/
vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename
Request Body:
<CallControl>
  <op>RECORD</op>
  <resourceType>STREAM</resourceType>
  <resourceId>67ed783c-203f-454b-a0e6-57b77820c831.wav</resourceId>
  <lastResult>0</lastResult>
  <speed>100</speed>
  <volume>100</volume>
  <startPosition>0</startPosition>
</CallControl>
```

Response Code: 204

JSON Example

```
POST https://<connectionserver>/
vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename
Accept : application/json
Content-Type : application/json
Connection : keep-alive
Request Body:
{
  "op": "RECORD",
  "resourceType": "STREAM",
  "resourceId": "67ed783c-203f-454b-a0e6-57b77820c831.wav",
  "lastResult": "0", "speed": "100",
  "volume": "100",
  "startPosition": "0"
}
```

Response Code: 204

Use the following URL to listen to the voice name associated with the interview handler: Paste the URL in the browser and listen to the voice name.

<https://<connection-server>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request

The voice name can also be updated using the input stream. An input stream can be created from the .wav file and passed as the request body. The URL for this should be:

<https://<connection-server>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename>

The request body should be like this:

```
put.setRequestBody(new FileInputStream(file3));
```

where the PUT request is created to upload the file3 .wav file. Make sure the content type for the request should be passed as "audio/wav". Use the following URL from the browser to listen to the voice name associated with the interview handler:

<https://<connection-server>/vmrest/handlers/interviewhandlers/<interviewhandlerobjectid>/voicename>

Adding Voice Name by Passing Input Stream in the Request