



# Configuring VoiceXML Voice Store and Forward

The VoiceXML Voice Store and Forward feature allows streaming-based voice recording and playback features for various media including local memory, HTTP, ESMTTP, and RTSP for 14 different Cisco codecs and two standard audio file formats, .au and .wav. This chapter explains how to configure the VoiceXML Store and Forward feature.

For more information about this and related Cisco IOS voice features, see the following:

- “Overview of Cisco IOS Tcl IVR and VoiceXML Applications” on page 1
- Entire Cisco IOS Voice Configuration Library—including library preface and glossary, other feature documents, and troubleshooting documentation—at [http://www.cisco.com/en/US/docs/ios/12\\_3/vvf\\_c/cisco\\_ios\\_voice\\_configuration\\_library\\_glossary/vcl.htm](http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary/vcl.htm).



## Note

For releases prior to Cisco IOS Release 12.3(14)T, see the previous version of the *Cisco Tcl IVR and VoiceXML Application Guide* at: [http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax\\_c/tcl\\_leg/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/tcl_leg/index.htm)

## Feature History for VoiceXML Voice Store and Forward

Release	Modification
12.2(11)T	This feature was introduced.
12.3(14)T	A new command-line interface structure for configuring Tcl and IVR applications was introduced and affected the commands for configuring this feature.

## Contents

- Prerequisites for VoiceXML Voice Store and Forward, page 2
- Restrictions for VoiceXML Voice Store and Forward, page 2
- Information About VoiceXML Store and Forward, page 3
- How to Configure VoiceXML Voice Store and Forward, page 8
- Where to Go Next, page 32
- Additional References, page 32

When developing and configuring a voice application, refer to this chapter and to the *Cisco VoiceXML Programmer's Guide* or the *Tcl IVR API Version 2.0 Programmer's Guide*.

## Prerequisites for VoiceXML Voice Store and Forward

- You must configure basic VoiceXML application functionality as described in “Configuring Basic Functionality for Tcl IVR and VoiceXML Applications” on page 1.
- You must write a VoiceXML 2.0 document that implements voice store and forward features. To write your own script, refer to the *Cisco VoiceXML Programmer's Guide*.

## Restrictions for VoiceXML Voice Store and Forward

The following sections list restrictions for the VoiceXML Voice Store and Forward feature:

### Recording and Playback Restrictions

- Playback of audio recordings directly from an ESMTTP server is not supported.
- For ESMTTP recording, only a single mailbox address is supported in the mailto URL.
- Final silence detection, which lets the gateway terminate a recording after a defined length of silence, is not supported for RTSP recording. The final silence detection feature is disabled by default; it must be enabled by using the final silence property in the VoiceXML document.
- The **vad** command must be configured in the VoIP dial peer when final silence detection is needed to terminate a voice recording. When using speech recognition, however, the **no vad** command is required, so final silence is not detected for recording on IP call legs.
- Maximum duration of a recording stored in local memory on the Cisco gateway is limited to the amount of available free memory. Memory limits can also be configured on the gateway by using the **ivr record memory session** and **ivr record memory system** commands.
- For RAM recordings submitted using HTTP POST:
  - Only the chunked transfer method is supported.
  - Recordings are sent using enctype of “multipart/form-data.”
  - The “audio/basic” enctype, which was supported in Cisco IOS Release 12.2(2)XB, is not supported in later releases of Cisco IOS software.
- For audio files streamed to an HTTP server:
  - Only the chunked transfer method is supported.
  - Audio files are sent using enctype of “multipart/form-data.”

### Audio File Data-Length Restrictions

Because the recording duration is unknown at the beginning of an HTTP or ESMTTP streaming session, the .au and .wav file headers generated by the Cisco gateway are encoded with a data length of 0 when recording to an HTTP or ESMTTP server; the gateway correctly decodes 0-length files during playback.

The standard .wav format, however, does not allow for a 0-length in the header, so third-party audio players may not be able to play back .wav files encoded by the gateway. Although the .au format allows files with a 0-length, some audio players may not support these files either.

To enable an audio player to play back recordings made by the Cisco gateway to an HTTP or ESMTTP server, you must use a correction utility to modify fields in the audio file headers, as described in the [“Correction Utility for Audio File Headers” section on page 7](#).

The following additional restrictions apply:

- Recordings made to an RTSP server, or to local memory and submitted to HTTP, use the correct data length in the headers and do not require a correction utility.
- The Cool Edit audio player recognizes 0-length .au and .wav files recorded by the gateway using G.711 u-law so no changes to those files are required. Use the correction utility to enable the Cool Edit audio player to play back .au and .wav files that use G.711 a-law.
- Windows Media Player and Vovida RTSP player require the correction utility to playback .au and .wav files using G.711 u-law, or .wav files using G.711 a-law.
- Audio files recorded by third-party audio players might not completely follow the .au or .wav file format standards. Cisco reserves the right not to support non-standard .au or .wav file formats.
- Concatenating two audio files without first correcting the data length fields and the data offsets can result in files with invalid data lengths. Cisco reserves the right not to support invalid data-length audio files.

## Information About VoiceXML Store and Forward

To configure audio files properties for Tcl and VoiceXML applications, you must understand the following concepts:

- [VoiceXML Voice Store and Forward, page 3](#)
- [VoiceXML Audio Recording Scenario, page 4](#)
- [Audio File Formats Supported for Recording and Playback, page 5](#)
- [VoiceXML Recording Locations, page 5](#)
- [Codec Support for Audio Recording, page 5](#)
- [Codec Mappings in Audio Recordings, page 7](#)
- [Correction Utility for Audio File Headers, page 7](#)

## VoiceXML Voice Store and Forward

The VoiceXML Voice Store and Forward feature expands Cisco IOS VoiceXML to include the input and processing of form field entries using recorded audio clips, rather than numeric input only. Audio clips can be captured and then submitted to an external web server using HTTP or RTSP, or to a messaging server using Extended Simple Mail Transfer Protocol (ESMTTP) for additional processing.

This recording feature can be used to collect caller names or addresses for call screening, product registration, or similar e-commerce applications, and for simple voice messaging, or for any voice browser application where alphanumeric input using DTMF is cumbersome or impractical.

The VoiceXML Voice Store and Forward feature supports speech recording and playback with a choice of four different media locations, including:

- Local memory—Voice recordings are stored in local memory on the Cisco gateway, and can be played back or submitted to an HTTP server using the POST method. Intended for temporarily storing short-length speech clips, such as caller name or address, or a short voice message.

- HTTP—Voice recording is streamed directly to an external HTTP server using the URL specified by the user. Recording can be played back in streaming or non-streaming mode.
- RTSP—Voice recording is directly streamed to and from an external RTSP server using the URL specified by the user. Intended for storing indefinite-length audio recordings.
- ESMTP—Voice recording is directly streamed to the ESMTP server as e-mail audio attachments. This option supports the Mailto: URL.

The Voice Store and Forward feature enables dynamic voice messaging by switching a busy or no-answer voice call to a VoiceXML application. The voice gateway can operate in two modes:

- On-ramp mode—Incoming calls are handled by a VoiceXML document that lets callers record voice messages if the called party is busy or there is no answer. The on-ramp gateway stores the voice recordings as audio clips to the selected media location; an external HTTP or RTSP server, internal memory if space is available, or by directly streaming the voice message as an e-mail attachment to an external ESMTP server.

To configure the gateway to record and play back voice messages, see the [“Configuring the On-Ramp Gateway for VoiceXML Voice Store and Forward”](#) section on page 8.

- Off-ramp mode—An external mail server sends an e-mail notification to the off-ramp gateway. The off-ramp gateway extracts the dialed number in the e-mail header and places an outbound call to the corresponding PSTN or IP destination. When the call is answered, the gateway executes the configured VoiceXML application. The VoiceXML application retrieves the audio clip from the external media server and plays the message to the PSTN or IP destination. The gateway does not support the streaming of audio clips directly from the ESMTP server.

To configure the gateway to receive e-mail triggers and make outbound calls to deliver voice messages, see the [“Configuring the Off-Ramp Gateway to Place a Call”](#) section on page 16.

## VoiceXML Audio Recording Scenario

The following is an example call scenario for a VoiceXML application using recording capabilities:

1. The caller dials a number and is connected through the PSTN or the IP network to a Cisco voice gateway that is configured as a VoiceXML-enabled gateway.
2. The Cisco voice gateway associates the dialed number with the appropriate VoiceXML document, residing on a web server.
3. The voice gateway runs the VoiceXML document and responds to the caller’s input by playing the appropriate audio content.
4. The gateway executes the document, which prompts the user to record a voice message.
5. The message is recorded by the gateway and stored in local memory with the selected audio encoding.
6. After the recording is completed, the user can review the message or submit it by either pressing a specified key or hanging up the call.
7. When the user submits the voice mail message, the gateway’s VoiceXML browser submits the voice message in .au or .wav file format to a specified URL using the HTTP POST method.
8. After receiving the message, the web server can store the message in the appropriate mailbox.
9. After successfully storing the voice message, the application instructs the media stream process to delete the local copy of the voice message.

## Audio File Formats Supported for Recording and Playback

Cisco VoiceXML gateways support two standard audio formats for recording and playback: .au (audio/basic) and .wav (audio/wav). The format used to record an audio file is specified by the VoiceXML document at the time of the recording. If it is not defined in the VoiceXML document, the default format type is audio/basic. The gateway uses standard codec numbers for the codec format in the audio header whenever possible, including G.711 u-law, G.711 a-law, and G.726 (32k ADPCM); other codecs use a proprietary format mapping and may therefore not be recognized by third-party audio players. For a listing of the codec numbers used by the Cisco gateway, see the [“Codec Mappings in Audio Recordings” section on page 7](#).

Recordings made by the gateway are open-ended and real-time streaming, so the .au and .wav file headers generated by the Cisco gateway are encoded with a data length of 0. All audio files recorded by the Cisco gateway can be played back by the gateway, however, because the standard .wav format does not allow for a 0 length in the header, some third-party audio players may not be able to play back .wav files encoded by the gateway. The .au file format allows a 0 length, but some audio players may also not support these files.

For example, the Cool Edit audio player can play 0-length .au and .wav files that are recorded by the gateway using the G.711 u-law codec. Other audio players and other codecs require the audio file header to be modified before playback. For example, the Windows Media Player audio player can play G.711 u-law and G.711 a-law audio files if the 0-length field is modified. The Cool Edit audio player can play G.711 a-law files if the header is corrected.

To enable an audio player to play back recordings made by the Cisco gateway, you must use a correction utility to modify specific fields in the audio file headers. For information about how to modify the file headers, see the [“Correction Utility for Audio File Headers” section on page 7](#).

## VoiceXML Recording Locations

The VoiceXML Voice Store and Forward feature supports audio recording and playback using local memory on the Cisco gateway or a choice of external media server locations:

- Local memory—Recording and playback is supported for storing and retrieving audio files. Audio recordings can also be submitted to HTTP servers for permanent storage.
- ESMTP—Recording is supported by directly streaming audio to ESMTP server as e-mail attachment. Payout directly from the ESMTP server is not supported.
- HTTP—Recording is supported by directly streaming audio to HTTP server using the chunked transfer-encoding method; playback is supported using streaming and non-streaming methods.
- RTSP—Recording and payout is supported by directly streaming audio to and from an RTSP server.
- TFTP—Payout supported by retrieving audio file from a TFTP server, using streaming or non-streaming methods. Recording audio to a TFTP server is not supported.

The URL of the recording destination is specified in the VoiceXML document by using the Cisco property `cisco-dest`. For more information, refer to the [Cisco VoiceXML Programmer's Guide](#).

## Codec Support for Audio Recording

[Table 5-1](#) shows the codecs that are supported for audio recording and playback, by platform and minimum required Cisco IOS release.

**Note**

All codecs listed are supported for H.323 and SIP unless otherwise noted.

The GSMFR codec is not supported on the Cisco Integrated Services Routers (ISRs) for media recording.

**Table 5-1**      **Codec Support for Audio Recording by Platform and Minimum Cisco IOS Release**

Codec	Cisco IOS Value	Cisco IOS Release			
		Cisco 3600 Series	Cisco AS5300	Cisco AS5350	Cisco AS5400
G.711 a-law	g711alaw	12.2(11)T	12.2(11)T	12.2(11)T	12.2(11)T
G.711 u-law	g711ulaw	12.2(11)T	12.2(2)XB	12.2(2)XB	12.2(2)XB
G.723.1 Annex-A (5.3 kbps)	g723ar53	12.2(11)T	12.2(11)T	12.2(11)T	12.2(11)T
G.723.1 Annex-A (6.3 kbps)	g723ar63	12.2(11)T	12.2(11)T	12.2(11)T	12.2(11)T
G.723.1 (5.3 kbps)	g723r53	12.2(11)T	12.2(2)XB <sup>1</sup>	12.2(2)XB <sup>1</sup>	12.2(2)XB <sup>1</sup>
G.723.1 (6.3 kbps)	g723r63	12.2(11)T	12.2(2)XB	12.2(2)XB <sup>1</sup>	12.2(2)XB <sup>1</sup>
G.726 (16 kbps) <sup>2</sup>	g726r16	12.2(11)T	12.2(11)T <sup>3</sup>	12.2(11)T <sup>3</sup>	12.2(11)T <sup>3</sup>
G.726 (24 kbps) <sup>2</sup>	g726r24	12.2(11)T	12.2(11)T	12.2(11)T <sup>3</sup>	12.2(11)T <sup>3</sup>
G.726 (32 kbps) <sup>2</sup>	g726r32	12.2(11)T	12.2(11)T	12.2(11)T <sup>3</sup>	12.2(11)T <sup>3</sup>
G.728 (16 kbps)	g728	12.2(11)T	12.2(11)T	Not supported	Not supported
G.729 (8 kbps) <sup>2</sup>	g729r8 (high complexity)	12.2(11)T	12.2(11)T	12.2(11)T	12.2(11)T
G.729 Annex-A (8 kbps)	g729r8 (medium complexity)	12.2(11)T	Not supported	Not supported	Not supported
G.729 Annex-B (8 kbps) <sup>2</sup>	g729br8 (high complexity)	12.2(11)T	12.2(11)T	12.2(11)T	12.2(11)T
G.729A Annex-B (8 kbps)	g729br8 (medium complexity)	12.2(11)T	Not supported	Not supported	Not supported

1. This codec is supported only for H.323 on this platform in Cisco IOS Release 12.2(2)XB; it is supported for SIP in Cisco IOS Release 12.2(11)T.

2. This codec is supported only for audio playback in Cisco IOS Release 12.2(2)XB.

3. This codec is not supported for SIP on this platform.

**Note**

- If the codec for an audio recording is not specified in the VoiceXML document, the default codec used for the recording is G.711 u-law.
- For recording and playback over an IP call leg, the negotiated codec must match the codec specified in the VoiceXML document. If the codec specified for the audio file is different than the codec negotiated for the call, the recording or playback fails and an error is generated.
- To determine specific codec support for your platform and Cisco IOS release, use the **codec** command in dial-peer configuration mode.

## Codec Mappings in Audio Recordings

Table 5-2 lists the codec number that is mapped to each codec in the audio file header, for .au format and .wav format files.

**Table 5-2** Codec Numbers Used in Audio File Headers

.AU Files		.WAV Files	
Number	Codec	Number	Codec
1	PCMULAW	1	MS_PCM
23	32K_ADPCM	2	MS_ADPCM
27	PCMALAW	6	G711_ALAW
39	CISCO_G729	7	G711_MULAW
42	CISCO_G729_b	100	32K_ADPCM
47	CISCO_G723_1r53	5339	CISCO_G729
48	CISCO_G723_1r63	5342	CISCO_G729_b
49	CISCO_G723_1ar53	5347	CISCO_G723_1r53
50	CISCO_G723_1ar63	5348	CISCO_G723_1r63
51	CISCO_G726_r16	5349	CISCO_G723_1ar53
52	CISCO_G726_r24	5350	CISCO_G723_1ar63
53	CISCO_G726_r32	5351	CISCO_G726_r16
55	CISCO_G728	5352	CISCO_G726_r24
		5353	CISCO_G726_r32
		5355	CISCO_G728

## Correction Utility for Audio File Headers

This section describes the header fields that a correction utility must modify to enable an audio player to play back recordings made by the Cisco gateway to an HTTP or ESMTTP server.

### .AU File Format Correction

The data\_size field in the .au header requires correcting. It is located at offset 8 from the beginning of the .au file. The correction utility must find the total size of the source .au file, subtract the .au header size at offset 4 from this file size, and insert the result into the data\_size field at the offset 8 position.

The Cisco gateway puts a value of 24 in offset 4 and a value of 0 in offset 8, but this is not guaranteed. The correction utility should follow the correction logic outlined above.



#### Note

The .au header fields are in big-endian format; byte-swapping is required for a correction utility running on little-endian platforms.



## .WAV File Format Corrections

Three fields in the .wav header require correcting:

- `total_size_minus_8` field—Located at offset 4 from the beginning of the .wav file. The correction utility must find the total size of the source .wav file, subtract 8 from this file size, and insert the result into this field at the offset 4 position.
- `chunk_data_length` field—The correction utility must find the total size of the source .wav file, and subtract one of these values from the file size:
  - 90 bytes for 32k ADPCM (G.726) codec
  - 56 bytes for all other codecs

then insert the result into the field at one of these offset positions:

- Offset 86 for 32k ADPCM (G.726) codec
- Offset 52 for all other codecs
- `total_sample_blocks` field—total number of sample blocks inserted at this position:
  - Offset 74 for 32k ADPCM (G.726) codec
  - Offset 40 for all other codecs

The Cisco gateway puts a value of 0 in offset 4 and a value of 0 in offset 40, but this is not guaranteed. The correction utility should follow the correction logic outlined above.



### Note

The .wav header fields are in little-endian format; byte-swapping is required for a correction utility running on big-endian platforms.

## How to Configure VoiceXML Voice Store and Forward

This section contains the following procedures:

- [Configuring the On-Ramp Gateway for VoiceXML Voice Store and Forward, page 8](#)
- [Configuring the Off-Ramp Gateway to Place a Call, page 16](#)

## Configuring the On-Ramp Gateway for VoiceXML Voice Store and Forward

This section contains the following procedures for configuring the on-ramp gateway to record voice messages to an ESMTTP server:

- [Configuring the Interface Type for Sending Voice Mail, page 9](#) (optional)
- [Configuring the Sending MTA, page 9](#) (required)
- [Configuring the POTS Dial Peer, page 11](#) (required)
- [Verifying the On-Ramp Gateway Configuration, page 12](#) (optional)



## On-Ramp Gateway for VoiceXML Voice Store and Forward

When acting as an on-ramp voice gateway, the Cisco VoiceXML gateway records voice messages from end users, converts them into e-mail attachments, and forwards the MIME e-mail message with the voice mail attachments to an ESMTP server for storage. The voice calls and the delivery of the e-mails with voice mail attachments are controlled by VoiceXML documents running on the Cisco gateway.

Voice mail stored on the ESMTP server can be delivered either as an e-mail message with attachment when the recipient downloads messages or it can be forwarded as an e-mail notification to the off-ramp gateway. In the latter case, the ESMTP server delivers the voice mail to the off-ramp Cisco gateway, which processes the e-mail and initiates an outbound call to the destination. For more information, see the [“Configuring the Off-Ramp Gateway to Place a Call” section on page 16](#).

The Cisco gateway supports the following mail formats:

- Mailto URL in its simplest form, which is the Internet mail address, as described in RFC-2368.
- Content transfer encoding is Base64.
- MIME content type is multipart/voice-message with the audio/\* sub-type. Within the multipart/voice-message level, only one audio/\* sub-type is enclosed.
- Audio formats are audio/basic and audio/wav.

The content type and audio format is set through a VoiceXML property. The sending MTA defines the delivery parameters for the e-mail message to which the recorded audio file is attached. The sending MTA can be specified through properties in the VoiceXML document or it can be configured on the Cisco gateway. A POTS dial peer on the gateway associates the dialed number with the appropriate VoiceXML application.

## Configuring the Interface Type for Sending Voice Mail

**Note**

On platforms with only voice cards, the default interface type is **fax-mail**, so you do not have to configure this command if your gateway has only voice cards. On platforms with a combination of modem and voice cards, the default is **modem**, and you must configure this command.

Specify the voice module in the gateway as the interface through which to send voice messages by using this command in global configuration mode:

```
fax interface-type fax-mail
```

Example: Router(config)# fax interface-type fax-mail

**Note**

After using the **fax interface-type fax-mail** command to change the interface type, you must reload the gateway for the new configuration to be effective.

## Configuring the Sending MTA

The sending MTA defines the elements of the e-mail message to which the recorded audio file is attached. These elements can be configured globally on the gateway or they can be specified through Cisco properties in the VoiceXML document. For information on specifying the MTA agent by using VoiceXML properties, refer to the [Cisco VoiceXML Programmer's Guide](#).

**Note**

Specifying the sending MTA by using VoiceXML properties takes precedence over the gateway configuration. Any value that is configured on the gateway is ignored if the same attribute is specified in the VoiceXML document.

To configure the sending MTA on the on-ramp gateway, enter the following commands in global configuration mode:

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **mta send mail-from hostname *string***
4. **mta send mail-from {username *string* | username *\$s\$*}**
5. **mta send server {host-name | ip-address}**
6. **mta send subject *string***
7. **mta send postmaster *e-mail-address***
8. **mta send origin-prefix *string***
9. **mta send return-receipt-to hostname *string***
10. **mta send return-receipt-to username {*string* | *\$s\$*}**

**DETAILED STEPS**

**Step 1** Enable privileged EXEC mode:

**enable**

Example: Router> enable

Enter your password if prompted.

**Step 2** Enter global configuration mode:

**configure terminal**

Example: Router# configure terminal

**Step 3** Specify the host name portion of the origination e-mail address:

**mta send mail-from hostname *string***

- *string*—Text string that specifies the SMTP host name or IP address. To specify an IP address, enclose it in brackets as follows: [xxx.xxx.xxx.xxx].

Example: Router(config)# mta send mail-from hostname onramp-gateway.com

**Step 4** Specify the user name portion of the origination e-mail address:

**mta send mail-from {username *string* | username *\$s\$*}**

- **username *string***—Text string that specifies the sender's username.
- **username *\$s\$***—Macro that specifies that the username comes from the calling number.

Example: Router(config)# mta send mail-from username mailuser

**Note**

The **mta send mail-from hostname** and **mta send mail-from username** commands configure the RFC 822 From: field and the RFC 821 Mail From field of the e-mail message. These two commands form a complete e-mail address, like mailuser@onramp-gateway.com. The To: address is configured by using the cisco-dest property in the VoiceXML document.

**Step 5** Specify the destination server:

```
mta send server {host-name | ip-address}
```

- *host-name*—Host name of the destination mail server.
- *IP-address*—IP address of the destination mail server.

Example: Router(config)# mta send server mail-server

**Step 6** Define the text that appears in the Subject field of the e-mail message:

```
mta send subject string
```

- *string*—Text string that appears in the subject header of an e-mail message.

Example: Router(config)# mta send subject e-mail subject goes here

**Step 7** Define the origination address to use if the **mta send mail-from** address is not configured:

```
mta send postmaster e-mail-address
```

- *e-mail-address*—Destination address (the mail server postmaster account). This is also the address where all undeliverable e-mail is sent.

Example: Router(config)# mta send postmaster test@mail-server

**Step 8** (Optional) Define additional identifying information to be prepended to the e-mail header.

```
mta send origin-prefix string
```

- *string*—Text string that adds comments to the e-mail prefix header. If this string contains more than one word, the string value should be contained within quotation marks ("x").

Example: Router(config)# mta send origin-prefix "here is origin-prefix"

**Step 9** (Optional) Specify the host name address where MDNs are sent:

```
mta send return-receipt-to hostname string
```

- **hostname** *string*—Text string that specifies the SMTP host name where MDNs will be sent.

Example: Router(config)# mta send return-receipt-to hostname onramp-gateway.com

**Step 10** (Optional) Specify the username address where MDNs are sent:

```
mta send return-receipt-to username {string | $$$}
```

- *string*—Text string that specifies the sender's username where MDNs will be sent.
- **\$\$\$**—Wildcard that specifies that the calling number (ANI) generates the disposition-notification to the e-mail address.

Example: Router(config)# mta send return-receipt-to username \$\$\$



**Note** If the **mta send return-receipt-to** address is not configured, the on-ramp gateway inserts the postmaster address in this field as a default.

## Configuring the POTS Dial Peer

To configure the POTS dial peer, use the following commands beginning in global configuration mode:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *number* **pots**
4. **service** *name*
5. **incoming called-number** *string*

## 6. direct-inward-dial

### DETAILED STEPS

- 
- Step 1** Enable privileged EXEC mode:
- ```
enable
```
- Example: Router> enable  
Enter your password if prompted.
- Step 2** Enter global configuration mode:
- ```
configure terminal
```
- Example: Router# configure terminal
- Step 3** Enter dial-peer configuration mode for a POTS dial peer:
- ```
dial-peer voice number pots
```
- number*—Number tag used to identify this dial peer. Range is from 1 to 2147483647.
- Example: Router(config)# dial-peer voice 900 pots
- Step 4** Associate a specific call application with this dial peer:
- ```
service service-name
```
- service-name*—Name of the VoiceXML application. This is the name of the application that was defined when the application was configured by using the application configuration submodes.
- Example: Router(config-dial-peer)# service vxml\_smtip
- Step 5** Specify the called number that links voice calls to this dial peer:
- ```
incoming called-number string
```
- string*—Sequence of digits representing the full or a partial telephone number (for example, the extension) used to reach the voice gateway. See the [“How Voice Applications are Matched to Called Numbers” section on page 41](#) for more information.
- If DID is enabled, the incoming called number (DNIS) is used to match the destination pattern of outgoing MMoIP dial peers.
- Example: Router(config-dial-peer)# incoming called-number 5551211
- Step 6** (Optional) Specify Direct Inward Dialing (DID) for this dial peer:
- ```
direct-inward-dial
```
- Example: Router(config-dial-peer)# direct-inward-dial
- 

## Verifying the On-Ramp Gateway Configuration

To verify the on-ramp gateway configuration, perform the following steps:

### SUMMARY STEPS

1. `show running-config`
2. `show dial-peer voice number`
3. `show call application voice summary`
4. `debug mmoip send email`

## DETAILED STEPS

**Step 1** Use the **show running-config** command to verify that:

- a. The interface type is set to **fax interface-type fax-mail**, for example:

```
!
fax interface-type fax-mail
```



**Note** If your gateway has only voice cards, **fax-mail** is the default setting so the command does not show up in the configuration output.

- b. The sending MTA is configured with the **mta send** commands, for example:

```
mta send server mapp-smtp
mta send subject subject line here
mta send origin-prefix This is the origin-prefix
mta send postmaster joe@mapp-smtp
mta send mail-from hostname Cisco-5300.cisco.com
mta send mail-from username $$$
mta send return-receipt-to hostname Cisco-5300.cisco.com
!
```

- c. The POTS dial peer is configured to accept inbound calls to the called number in the **incoming called-number** command and links calls to the VoiceXML document in the **service** command, for example:

```
!
dial-peer voice 5550 pots
service smtp_rec
incoming called-number 5550100
!
```

**Step 2** Use the **show dial-peer voice** command to display detailed configuration information about the dial peer, including whether it is operational and the assigned application. The following example output shows that POTS dial peer 5550 is linked to the application **smtp\_rec**.

Router# **show dial-peer voice 5550**

```
VoiceEncapPeer5550
  information type = voice,
  description = '',
  tag = 5550, destination-pattern = '',
  answer-address = '', preference=0,
  CLID Restriction = None
  CLID Network Number = ''
  CLID Second Number sent
  source carrier-id = '', target carrier-id = '',
  source trunk-group-label = '', target trunk-group-label = '',
  numbering Type = 'unknown'
  group = 5550, Admin state is up, Operation state is up,
  incoming called-number = '5550100', connections/maximum = 0/unlimited,
  DTMF Relay = disabled,
  huntstop = disabled,
  in bound application associated: 'smtp_rec'
  out bound application associated: ''
  dnis-map =
  permission :both
  incoming COR list:maximum capability
  outgoing COR list:minimum requirement
  Translation profile (Incoming):
  Translation profile (Outgoing):
```

```

incoming call blocking:
translation-profile = ''
disconnect-cause = 'no-service'
voice-port = ''
type = pots, prefix = '',
forward-digits default
session-target = '', up,
direct-inward-dial = disabled,
digit_strip = enabled,
register E.164 number with GK = TRUE
fax rate = system,    payload size = 20 bytes

Time elapsed since last clearing of voice call statistics never
Connect Time = 0, Charged Units = 0,
Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
Accepted Calls = 0, Refused Calls = 0,
Last Disconnect Cause is "",
Last Disconnect Text is "",
Last Setup Time = 0.

```

- Step 3** Use the **show call application voice summary** command to verify that the VoiceXML application is loaded and running on the gateway, for example:

```
Router# show call application voice summary
```

name	description
session	Basic app to do DID, or supply dialtone.
fax_hop_on	Script to talk to a fax redialer
clid_authen	Authenticate with (ani, dnis)
clid_authen_collect	Authenticate with (ani, dnis), collect if that fails
clid_authen_npw	Authenticate with (ani, NULL)
clid_authen_col_npw	Authenticate with (ani, NULL), collect if that fails
clid_col_npw_3	Authenticate with (ani, NULL), and 3 tries collecting
clid_col_npw_npw	Authenticate with (ani, NULL) and 3 tries without pw
DEFAULT	Default system session application
lib_off_app	Libretto Offramp
callme	flash:call.vxml
*smtp_rec	tftp://10.10.5.3/scripts/smtp_rec.vxml

```

TCL Script Version 2.0 supported.
TCL Script Version 1.1 supported.
Voice Browser Version 2.0 for VoiceXML 1.0 & 2.0 supported.

```



**Tip**

If an asterisk is displayed next to the application name when using the **summary** keyword, it means that the application is configured, but not running. Normally this is because the application was not successfully loaded. For troubleshooting information, see the [“Verifying Loading of Service” section on page 26](#).

- Step 4** Use the **debug mmoip send email** command and send an e-mail to a specified e-mail address to test connectivity between the on-ramp gateway and the e-mail server.

# Troubleshooting Recording Configuration

## SUMMARY STEPS

1. Play the recorded audio file using an audio tool such as Cool Edit.
2. **debug vxml puts**
3. **debug voip application**
4. **show vsp session**

## DETAILED STEPS

- 
- Step 1** Play the recorded audio file using an audio tool such as Cool Edit.
- Step 2** Use the **debug vxml puts** command to verify the duration and size of the recorded audio file.
- You can run a simple VoiceXML document that uses the <cisco-puts> tag to print the duration and size of the audio recording, for example:
- ```
<form id="record_to_http">

    <record name="myrec" beep="true" maxtime="15s" finalsilence="10s" dtmfterm="true"
    type="audio/basic;codec=g711ulaw">
    <prompt><audio src="record.au"/></prompt>
    </record>

    <block>
    <cisco-puts>DURATION:<cisco-putvar namelist="myrec$.duration"/></cisco-puts>
    <cisco-puts>SIZE:<cisco-putvar namelist="myrec$.size"/></cisco-puts> ....
    </block>
</form>
```
- Step 3** Use the **debug voip application** command to verify the start and stop times of the audio recording, for example:
- ```
Router# show dial-peer voice 555

*Mar 1 14:53:26.610: $ mr_record_start: stream 0x63811808 recording is now
startedmc=0x63819DC8
*Mar 1 14:53:36.610: ms_stop_record: mc=0x63819DC8, cid=0x38,
cause=MS_STOP_MAX_TIME, RECORDING, RAM
*Mar 1 14:53:36.610: mr_vsp_flush_stream: mc=0x63819DC8, cid=0x38
*Mar 1 14:53:36.610: ms_stop_record_done: cid=0x38 stopped at 14:53:33.376, dur=10000
(ms),
DISASSOCIATING
*Mar 1 14:53:36.610: msu_recrd_ms_record_complete: callID=0x38(56),
context=0x62E13E9C,
cause=MS_STOP_MAX_TIME, stream_id=1
*Mar 1 14:53:36.610: msu_recrd_ms_record_complete: Recorded MC:0x63819DC8
URL ram:myrecord_6_0_56
*Mar 1 14:53:36.610: msu_call_app: app_cbf=0x610B09E0
```
- Step 4** Use the **show vsp session** command to view the packet statistics for the recording session, for example:
- ```
Router# show vsp session

VSP_STATS: Session Statistics -
  sessions total=4; max_active=1, current=1
  session_duration last=0; max=21000, min=21000 ms
  pre_stream_wait last=0; max=20, min=20 ms
  stream_duration last=0; max=20980, min=20980 ms
  post_stream_wait last=0; max=0, min=0 ms
  stream_size last=240; max=167804, min=240 bytes
```



```

streaming_rate last=0; max=7000, min=7000 bytes/sec
total_packet_count last=0; max=745, min=745 packets
drop_packet_count last=0; max=0, min=0 packets
particle_packet_count last=0; max=745, min=745 packets

VSP: current reference time=63832
      se_st = session start time; st_st = stream start time
      rx_ct = rx codec type; tx_ct = tx codec type
      rx_pt = rx payload type; tx_pt = tx payload type
VSP:<4>[0x649FBE78] (0x649FBE78) MS->HTTP se_st=277460, st_st=277472; rx_ct=g711ulaw,
tx_ct=g711ulaw

```

---

## Configuring the Off-Ramp Gateway to Place a Call

To configure the off-ramp gateway, perform the tasks in the following sections:

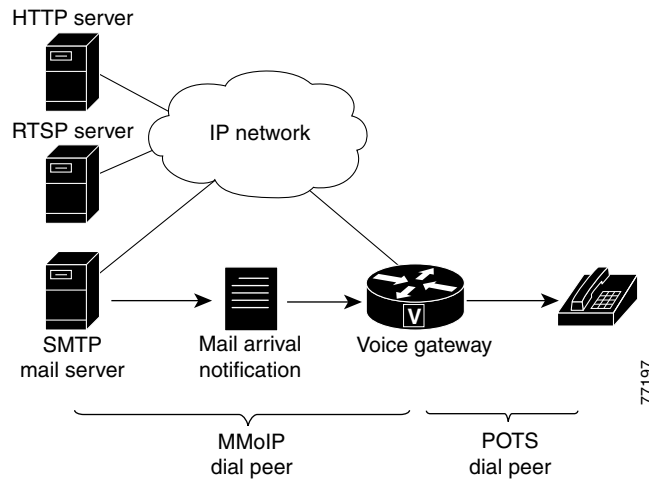
- [Downloading the Tcl Script for Off-Ramp Mail Application, page 19](#) (required)
- [Loading the Mail Application onto the Gateway, page 19](#) (required)
- [Configuring the Interface Type for Receiving Voice Mail, page 21](#) (required)
- [Configuring the Receiving MTA for Voice Store and Forward, page 21](#) (required)
- [Configuring the POTS Dial Peer, page 22](#) (required)
- [Configuring the MMoIP Dial Peer, page 23](#) (required)
- [Verifying the Off-Ramp Gateway Configuration, page 24](#) (optional)

### Off-Ramp Gateway Placing a Call

The Cisco VoiceXML gateway can act as an off-ramp gateway to dial the PSTN or IP network after it receives an e-mail notification of a voice message. The off-ramp gateway:

- Accepts e-mail notifications of voice messages from ESTMP servers.
- Hands e-mail messages over to the off-ramp mail application script, *offramp-mapp.tcl*, which extracts the destination telephone number.
- Places an outgoing call to the destination PSTN or IP voice user.
- Loads the specified VoiceXML application when the destination answers.
- Retrieves audio clips and plays out voice messages to the destination.

The off-ramp gateway uses receiving MTAs to define the parameters of the SMTP server. An MMoIP dial peer specifies the Tcl mail application that hands calls off to the outbound POTS dial peer. A call is placed to the destination telephone number and the specified VoiceXML application is executed when the call is answered.

**Figure 5-1 Voice Store and Forward Off-Ramp Call****Note**

Converting voice mail attachments to audio clips and saving them to an external RTSP or HTTP server is the responsibility of external application servers; it is not within the scope of the Cisco VoiceXML gateway.

## Off-Ramp Mail Trigger Call Scenario

The following steps describe how the off-ramp gateway delivers voice messages:

- Step 1** The gateway receives an e-mail notification from the external MTA. The notification e-mail does not contain a real voice message; it merely acts as a trigger to launch the off-ramp mail application. The voice mail attachments are ignored by the gateway.
- Step 2** If the receipt To: field in the message header begins with `VXML=`, the e-mail message is treated as a voice notification. The following example shows how the receipt To: field in the e-mail message header begins with `VXML=`, followed by the called number:

```
To: VXML=+16505554321@cisco.com
From: "John Smith" <15105553456@cisco.com>
Date: Mon, 26 Sept 2001 10:20:20 -0700 (CDT)
Subject: Voice message from cisco systems
Message-ID: 1234563456789@cisco.com
MIME-Version: 1.0 (Voice 2.0)
X-Mailer: IOS (tm) 5300 Software (C5300-IS-M)
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

If the receipt To: field begins with `FAX=`, it is considered a T.37 fax message and is handed off to the Store and Forward Fax feature.

- Step 3** The gateway extracts the destination telephone number from the e-mail header and matches it to an MMoIP dial peer.
- Step 4** The mail application that is configured in this MMoIP dial peer is executed. This is the mail application that uses the `app_voicemail_offramp.tcl` script provided by Cisco.
- Step 5** The mail application hands off the call to the POTS dial-peer that matches the dialed number, and an outbound call is made to the PSTN using this destination telephone number.

- Step 6** If the call is answered, the gateway executes the VoiceXML document that is configured in the POTS dial peer. If the called number is busy or there is no answer, the VoiceXML script is not executed.
- If DSN is configured and the called number is busy, or there is no answer or some other error occurs, the sender receives a “mail is not deliverable” error message. When there is no answer or the called number is busy, the gateway sends the DSN error status after the SMTP transaction completes (which means that the gateway has received the full e-mail from the mail server). If the gateway can not match an outbound dial peer, the DSN error message is delivered immediately.
- 

## MDNs

A message disposition notification (MDN) indicates that an e-mail message has been opened. A sender requests that an MDN be returned when the receiver opens an e-mail message. The following header is included in the e-mail header of the message:

```
Disposition-Notification-To:
```

The MDN is initiated by the sending e-mail client, and the return receipt is generated by the receiving e-mail client. Most PC-based e-mail software applications, such as Eudora, Netscape Messenger, and Microsoft Outlook, generate MDNs. The MDN is sent by the on-ramp gateway to the user defined in the **mta send return-receipt-to** command.

RFC 2298 requires that the receiver be allowed to prevent the automatic generation of an MDN. Because of this requirement, it is difficult to determine whether or not the user has actually received the e-mail message. For example, the recipient can always choose not to respond to MDN requests, or the recipient software might not understand or accept MDN requests.

## DSNs

A delivery status notification (DSN) is a message or response that is automatically generated and sent to the originator of an e-mail message by the SMTP server, notifying the sender of the status of the e-mail message. The on-ramp DSN request is included as part of the e-mail message sent by the on-ramp gateway if the appropriate DSN property is specified in the VoiceXML document. The on-ramp DSN response is generated by the SMTP server when the e-mail message is accepted. The DSN is sent to the user defined in the **mta send mail-from** command.

The off-ramp DSN is requested by the e-mail client. The DSN response is generated by the off-ramp gateway when it receives a request as part of the e-mail message. DSNs can only be generated if the mail client on the SMTP server is capable of responding to a DSN request. Because the off-ramp gateway generates the DSNs, you must configure both the **mta send mail-from** and **mta send return-receipt to** commands to use the DSN feature, for example:

```
mail from: <user@mail-server.company.com>
rcpt to: <555-1212@company.com> NOTIFY=SUCCESS,FAILURE,DELAY
```

Three different states can be reported back to the sender in the DSN:

- Delayed—Message delivery was delayed.
- Failure—SMTP server was unable to deliver the message to the recipient.
- Success—Message was successfully delivered to the recipient mailbox.

**Note**

For information on specifying DSN properties in a VoiceXML document, refer to the *Cisco VoiceXML Programmer's Guide*.

## Downloading the Tcl Script for Off-Ramp Mail Application

Before configuring the Cisco gateway to act as an off-ramp gateway for placing outbound calls, you must download and install the required Tcl script for the off-ramp mail application. This Tcl script, *app\_voicemail\_offramp.tcl*, hands off calls to the configured VoiceXML application. For a description of how this mail trigger works, see the “Off-Ramp Mail Trigger Call Scenario” section on page 17.

Download the required Tcl script by completing the following steps:

- 
- Step 1** Log in to the Cisco web site and go to <http://www.cisco.com/cgi-bin/tablebuild.pl/tclware>.
- Step 2** Select the Tcl zip file that contains the mail application: *app-voicemail-offramp.2.0.0.0.zip* (or later).
- Step 3** Select the Cisco.com server nearest your physical location and download the file.
- Step 4** Unzip the contents of the file.
- The zip file that you download includes:
- Mail application Tcl script (*app\_voicemail\_offramp.tcl*)
  - ReadMe file
  - Call flow file
- Step 5** Move the application script file (*app\_voicemail\_offramp.tcl*) to a location that can be accessed by your gateway using a URL format.
- Tcl scripts and VoiceXML documents can be stored in any of the following locations: TFTP, FTP, or HTTP servers, in Flash memory of the voice gateway, or on the removable disks of the Cisco 3600 series. The audio files that they reference can be stored in any of these locations, and on RTSP servers. The URL is a standard URL that points to the location of the script. Examples include:
- *flash:myscript.tcl*—The script called *myscript.tcl* is being loaded from Flash memory on the router.
  - *slot0:myscript.tcl*—The script called *myscript.tcl* is being loaded from a device in slot 0 on the router.
  - *tftp://bigserver/myscripts/ticketime.tcl*—The script called *ticketime.tcl* is being loaded from a server called *bigserver* in a directory within the *tftpboot* directory called *myscripts*.
- 

## Loading the Mail Application onto the Gateway

To load the off-ramp mail application and corresponding VoiceXML application into the gateway's memory, enter the following commands:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**

4. **service** *application-name location*
5. **param mail-script** *application-name*
6. **service** *application-name location*
7. **param dsn-script** *application-name*

## DETAILED STEPS

**Step 1** Enable privileged EXEC mode:

**enable**

Example: Router> enable

Enter your password if prompted.

**Step 2** Enter global configuration mode:

**configure terminal**

Example: Router# configure terminal

**Step 3** Enter application configuration mode:

**application**

Example: Router(config)# application

**Step 4** Load the off-ramp mail script and assign it an application name:

```
application
  service application-name location
```

- *application-name*—Name of the off-ramp mail application.
- *location*—Directory and filename of the Tcl script in URL format. For example, flash memory (flash:filename), a TFTP (tftp://../filename) or an HTTP server (http://../filename) are valid locations. This application must use the app\_voicemail\_offramp.tcl script downloaded from Cisco.

Example:

Router(config)# application

Router(config-app)# service off tftp://serv1/app\_voicemail\_offramp.tcl

**Step 5** Load the VoiceXML application that handles calls from the off-ramp mail application:

```
application
  service application-name location
```

- *application-name*—Name of the VoiceXML application to which the off-ramp mail application hands off calls when the destination answers.
- *location*—Directory and filename of the VoiceXML document in URL format. For example, flash memory (flash:filename), a TFTP (tftp://../filename) or an HTTP server (http://../filename) are valid locations.

Example: Router(config)#

Router(config)# application

Router(config-app)# service mapp1 tftp://serv1/mapp-1.vxml

**Step 6** Link the off-ramp mail application to the VoiceXML application:

```
application
  service mail-application-name
  param mail-script application-name
```

- *mail-application-name*—Name of the off-ramp mail application that launches the app\_voicemail\_offramp.tcl script when the gateway receives an e-mail trigger. This is the application named in Step 4.

- *application-name*—Name of the VoiceXML application to which the off-ramp mail application hands off the call when the destination answers. This is the application named in Step 5.

Example:

```
Router(config)#application
Router(config-app)#service offramp
Router(config-app-param)#param mail-script mappl
```

- Step 7** (Optional) Link the off-ramp mail application to the VoiceXML application that handles calls for off-ramp DSN and MDN e-mail messages:

```
application
  service mail-application-name
  param dsn-script application-name
```

- *mail-application-name*—Name of the off-ramp mail application that launches the `app_voicemail_offramp.tcl` script when the gateway receives an e-mail trigger.
- *application-name*—Name of the VoiceXML application to which the off-ramp mail application hands off the call when the destination answers.

Example:

```
Router(config)#application
Router(config-app)#service offramp
Router(config-app-param)#param dsn-script map_dsn
```

## Configuring the Interface Type for Receiving Voice Mail



### Note

On platforms with only voice cards, the default interface type is **fax-mail**, so you do not have to configure this command if your gateway has only voice cards. On platforms with a combination of modem and voice cards, the default is **modem**, and you must configure this command.

Specify the voice module in the gateway as the interface through which to receive voice messages by using this command in global configuration mode:

```
fax interface-type fax-mail
```

Example: Router(config)# fax interface-type fax-mail



### Note

After using the **fax interface-type fax-mail** command to change the interface type, you must reload the gateway for the new configuration to take affect.

## Configuring the Receiving MTA for Voice Store and Forward

To configure the receiving MTA on the off-ramp gateway, enter the following commands in global configuration mode:

1. **enable**
2. **configure terminal**
3. **mta receive aliases** *string*
4. **mta receive maximum-recipients** *number*
5. **mta receive generate-mdn**

## DETAILED STEPS

---

**Step 1** Enable privileged EXEC mode:

**enable**

Example: Router> enable

Enter your password if prompted.

**Step 2** Enter global configuration mode:

**configure terminal**

Example: Router# configure terminal

**Step 3** Define a host name to use as an alias for the off-ramp gateway:

**mta receive aliases** *string*

- *string*—Host name or IP address to use as an alias for the SMTP server.

You can define up to ten different aliases. The Cisco gateway MTA only accepts incoming mail if the destination host name of the incoming mail matches one of the aliases configured by the **mta receive aliases** command. This command does not automatically allow reception for a domain IP address—it must be explicitly added. If you add an IP address, you must enclose the address in brackets as follows: [xxx.xxx.xxx.xxx].

Example 1: Router(config)# mta receive aliases cisco.com

Example 2: Router(config)# mta receive aliases [10.10.1.1]

**Step 4** Define the number of simultaneous SMTP recipients handled by this gateway:

**mta receive maximum-recipients** *number*

- *number*—Maximum number of recipients for all SMTP connections. Range is from 0 to 1024. The default is 0, which means that the off-ramp function is disabled.

This limits the number of resources (modems) allocated for e-mail transmissions.

Example: Router(config)# mta receive maximum-recipients 10

**Step 5** (Optional) Configure the off-ramp gateway to generate an MDN message when requested to do so:

**mta receive generate-mdn**

You may want to enable this feature depending on the type of mailer in use.

Example: Router(config)# mta receive generate-mdn

---

## Configuring the POTS Dial Peer

To configure a POTS dial peer that the off-ramp gateway uses to place calls, use the following commands beginning in global configuration mode:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *number* **pots**
4. **destination-pattern** *string*
5. **port** *controller-number*

### DETAILED STEPS

---

**Step 1** Enable privileged EXEC mode:



**enable**

Example: Router&gt; enable

Enter your password if prompted.

**Step 2** Enter global configuration mode:**configure terminal**

Example: Router# configure terminal

**Step 3** Enter dial-peer configuration mode for a POTS dial peer:**dial-peer voice** *number* **pots**

- *number*—Number tag used to identify this dial peer. Range is from 1 to 2147483647.

Example: Router(config)# dial-peer voice 555 pots

**Step 4** Identify the destination telephone number to which the mail application places calls:**destination-pattern** *string*

- *string*—Sequence of digits representing the full or a partial telephone number (for example, the extension) used to reach the destination.

Example: Router(config-dial-peer)# destination-pattern 5551939

**Step 5** Associate a specific voice port with this dial peer:**port** *controller-number*

This command specifies the T1 controller port through which outgoing voice calls are routed.

- *controller-number*—Number of the T1 or E1 controller.
- **:D**—D channel associated with ISDN PRI.

**Note**

The syntax of the **port** command is platform-specific. For information about the specific syntax for your platform, refer to the [Cisco IOS Voice Command Reference](#), Release 12.3.

Example 1: (Cisco 3600 series) Router(config-dial-peer)# port 1/0/0

Example 2: (Cisco AS5300, AS5350, and AS5400) Router(config-dial-peer)# port 1/0:D

## Configuring the MMoIP Dial Peer

To configure the off-ramp gateway MMoIP dial peer, use the following commands beginning in global configuration mode:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *number* **mmoip**
4. **service** *service-name*
5. **incoming called-number** *string*

### DETAILED STEPS

**Step 1** Enable privileged EXEC mode:**enable**

Example: Router&gt; enable

Enter your password if prompted.

**Step 2** Enter global configuration mode:

**configure terminal**

Example: Router# configure terminal

**Step 3** Enter dial-peer configuration mode for a MMOIP dial peer:

**dial-peer voice** *number* **mmoip**

- *number*—Number tag used to identify this dial peer. Range is from 1 to 2147483647.

Example: Router(config)# dial-peer voice 1000 mmoip

**Step 4** Associate the mail application with this dial peer:

**service** *service-name*

- *service-name*—Name of the mail application. Calls matched to this dial peer are handed off to this application. This application must use the app\_voicemail\_offramp.tcl script provided by Cisco.

Example: Router(config-dial-peer)# service mapp1

**Step 5** Specify the called number that links voice calls to this dial peer:

**incoming called-number** *string*

- *string*—Sequence of digits representing the full or a partial telephone number (for example, the extension) used to reach the voice gateway. See the [“How Voice Applications are Matched to Called Numbers”](#) section on page 41 for more information.

Example: Router(config-dial-peer)# incoming called-number 5551939



**Note**

The value of the **incoming called-number** command in the MMoIP dial peer must match the value of the **destination-pattern** command in the corresponding off-ramp POTS dial peer.

## Verifying the Off-Ramp Gateway Configuration

To verify the off-ramp gateway configuration, perform the following steps:

### SUMMARY STEPS

1. **show running-config**
2. **show dial-peer voice** *number*
3. **show call application voice summary**
4. Send an e-mail to the off-ramp gateway and request a return receipt.
5. **debug mta receive all**

### DETAILED STEPS

**Step 1** Use the **show running-config** command to verify that:

- a. The interface type is set to **fax interface-type fax-mail**, for example:

```
!
fax interface-type fax-mail
```

**Note**

If your gateway has only voice cards, **fax-mail** is the default setting so the command does not show up in the configuration output.

- b. The receiving MTA is configured with the **mta receive alias** and the **mta receive maximum-recipients** commands, for example:

```
mta receive aliases cisco.com
mta receive maximum-recipients 10
!
```

- c. The off-ramp mail application script and the associated VoiceXML document are configured on the gateway with the application configuration submodes, and the two applications are linked with the **param mail-script** command, for example:

```
application
  service offramp tftp://10.10.17.1/scripts/tcl/app_voicemail_offramp.tcl
  param mail-script mapp1
  service tftp://10.10.17.1/scripts/vxml/mapp-1.vxml
!
```

- d. The MMoIP dial peer is configured to use the off-ramp mail application with the **service** command, for example:

```
!
dial-peer voice 555 mmoip
  service offramp
  incoming called-number 5551212
```

**Note**

Be sure that the **information-type fax** command is not configured in the MMoIP dial peer. The information type is set to voice by default and therefore does not display in the configuration output.

- e. The POTS dial peer is configured to make outbound calls to the destination by using the called number in the **destination-pattern** command, for example:

```
!
dial-peer voice 5551 pots
  destination-pattern 5551212
  port 0:D
!
```

- Step 2** Use the **show dial-peer voice** command to display detailed configuration information about the dial peers, including the information type, whether they are operational, and the assigned application. For example, the following output shows that MMoIP dial peer 555 is linked to the application *offramp*.

Router# **show dial-peer voice 555**

```
MultiMediaOverIpPeer555
  information type = voice,
  description = '',
  tag = 555, destination-pattern = '',
  answer-address = '', preference=0,
  CLID Restriction = None
  CLID Network Number = ''
  CLID Second Number sent
  source carrier-id = '', target carrier-id = '',
  source trunk-group-label = '', target trunk-group-label = '',
  numbering Type = 'unknown'
  group = 555, Admin state is up, Operation state is up,
  incoming called-number = '555...', connections/maximum = 0/unlimited,
```

```

DTMF Relay = disabled,
modem transport = system,
huntstop = disabled,
in bound application associated: 'offramp'
out bound application associated: ''
dnis-map =
permission :both
incoming COR list:maximum capability
outgoing COR list:minimum requirement
Translation profile (Incoming):
Translation profile (Outgoing):
incoming call blocking:
translation-profile = ''
disconnect-cause = 'no-service'
type = mmioip, session-target = '',
session-protocol = SMTP,
Image Encoding Type = pass-through,
Image Resolution = pass-through,
Disposition Notification = disabled,
Delivery Status Notification = ,
Time elapsed since last clearing of voice call statistics never
Connect Time = 0, Charged Units = 0,
Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
Accepted Calls = 0, Refused Calls = 0,
Last Disconnect Cause is "",
Last Disconnect Text is "",
Last Setup Time = 0.

```

**Step 3** Use the **show call application voice summary** command to verify that the VoiceXML application is loaded and running on the gateway, for example:

```
Router# show call application voice summary
```

name	description
session	Basic app to do DID, or supply dialtone.
fax_hop_on	Script to talk to a fax redialer
clid_authen	Authenticate with (ani, dnis)
clid_authen_collect	Authenticate with (ani, dnis), collect if that fails
clid_authen_npw	Authenticate with (ani, NULL)
clid_authen_col_npw	Authenticate with (ani, NULL), collect if that fails
clid_col_npw_3	Authenticate with (ani, NULL), and 3 tries collecting
clid_col_npw_npw	Authenticate with (ani, NULL) and 3 tries without pw
DEFAULT	Default system session application
lib_off_app	Libretto Offramp
callme	flash:call.vxml
smtp_rec	tftp://10.10.5.3/scripts/smtp_rec.vxml
offramp	tftp://10.10.17.1/scripts/tcl/app_voicemail_offramp.tcl
mapp1	tftp://10.10.17.1/scripts/vxml/mapp-1.vxml

TCL Script Version 2.0 supported.

TCL Script Version 1.1 supported.

Voice Browser Version 2.0 for VoiceXML 1.0 & 2.0 supported.



#### Tip

If an asterisk is displayed next to the application name when using the **summary** keyword, it means that the application is configured, but not running. Normally this is because the application was not successfully loaded. For troubleshooting information, see the [“Verifying Loading of Service”](#) section on page 26.

- Step 4** Send an e-mail to the off-ramp gateway and request a return receipt. To be accepted by the gateway, the destination e-mail address must use the appropriate MTA receive alias, as configured by using the **mta receive alias** command.
- Step 5** Use the **debug mta receive all** command to view output relating to the activity on the SMTP server (messages exchanged, for example, the handshake) between the e-mail server and the off-ramp gateway.
- 

## Configuration Examples for VoiceXML Voice Store and Forward

This section provides the following gateway configuration examples of Cisco Tcl and VoiceXML applications. It contains comments in places especially relevant to the configuration of the feature.

- [VoiceXML Voice Store and Forward on Cisco AS5300 Example, page 27](#)
- [VoiceXML Voice Store and Forward Off-Ramp on Cisco 3600 Series Example, page 30](#)

### VoiceXML Voice Store and Forward on Cisco AS5300 Example

```
!  
version 12.3  
no service pad  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
service internal  
service udp-small-servers  
service tcp-small-servers  
!  
hostname Cisco-5300  
!  
logging buffered 1000000 debugging  
aaa new-model  
!  
aaa authentication login fax enable  
aaa authorization exec fax group radius  
aaa authorization network fax group radius  
aaa accounting connection fax stop-only group radius  
aaa session-id common  
!  
username access-class nopassword  
!  
resource-pool disable  
!  
ip subnet-zero  
ip tftp source-interface Ethernet0  
ip domain-name cisco.com  
ip host mail-server.cisco.com 1.14.116.1  
ip host demo 223.255.254.254  
ip host rtsp-ws 1.7.153.4  
ip host mapp-smtp.cisco.com 1.7.127.3  
ip host mapp-rtsp.cisco.com 1.7.127.151  
ip name-server 1.14.116.1  
!  
isdn switch-type primary-5ess  
!  
voice service pots  
    fax protocol t38 ls-redundancy 5 hs-redundancy 0
```

```

!
voice service voip
  fax protocol t38 ls-redundancy 5 hs-redundancy 0
  h323
!
fax interface-type fax-mail
mta send server mapp-smtp
mta send subject subject line here
mta send origin-prefix This is the origin-prefix
mta send postmaster joe@mapp-smtp
mta send mail-from hostname Cisco-5300.cisco.com
mta send mail-from username $$
mta send return-receipt-to hostname Cisco-5300.cisco.com
mta send return-receipt-to username $$
mta receive aliases mail-server2
mta receive aliases [1.7.87.4]
mta receive aliases cisco.com
mta receive maximum-recipients 10
mnoip aaa send-id primary gateway
mnoip aaa method fax authentication gateway
mnoip aaa send-authentication enable
mnoip aaa receive-accounting enable
mnoip aaa receive-authentication enable
!
controller T1 0
  framing esf
  clock source line primary
  linecode b8zs
  pri-group timeslots 1-24
!
controller T1 1
  framing sf
  linecode ami
!
controller T1 2
  framing sf
  linecode ami
!
controller T1 3
  framing sf
  clock source line secondary 3
  linecode ami
!
interface Ethernet0
  ip address 1.7.87.4 255.255.0.0
!
interface Serial0:23
  no ip address
  isdn switch-type primary-5ess
  isdn incoming-voice modem
!
interface FastEthernet0
  no ip address
  shutdown
  duplex auto
  speed auto
!
ip default-gateway 1.7.0.1
ip classless
ip route 223.255.254.0 255.255.255.0 1.7.0.1
no ip http server
ip pim bidir-enable
!
!

```

```
priority-list 1 protocol ip high tcp 1720
dialer-list 1 protocol ip permit
!
!
radius-server host 1.7.127.3 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key password
radius-server vsa send accounting
radius-server vsa send authentication
call rsvp-sync
application
  service rtsp_record tftp://demo/vxml/rtsp_record_vxml.vxml
  !
  service rtsp_play tftp://demo/vxml/rtsp_play.vxml
  !
  service http_submit tftp://demo/vxml/http_submit.vxml
  !
  service testcgi tftp://demo/vxml/testcgi.vxml
  !
  service http_play tftp://demo/vxml/http_play.vxml
  !
  service http_record tftp://demo/vxml/http_record.vxml
  !
  service mapp1 tftp://demo/tcl/mapp-1.tcl
    param mail-script offramp-mapp-test
    param authentication enable
    param authen-list fax
    param authen-method envelope-from
  !
  service offramp-mapp-test tftp://demo/vxml/testplay.vxml
  !
voice-port 0:D
!
mgcp modem passthrough voip mode ca
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer voice 101 pots
  description ***** change the application on this dial peer accordingly *****
  service http_play
  incoming called-number 5....
  port 0:D
!
dial-peer voice 208 voip
  shutdown
  description ***** no shut this dial peer for voip call *****
  destination-pattern 5556681
  session target ipv4:1.7.87.2
  codec g711ulaw
!
dial-peer voice 210 voip
  destination-pattern 52924
  session target ipv4:1.7.87.5
!
dial-peer voice 108 pots
  destination-pattern 5556681
  port 0:D
  prefix 95556681
!
dial-peer voice 545 pots
  incoming called-number 529..
  shutdown
  destination-pattern 529..
```



```

direct-inward-dial
port 0:D
prefix 529
!
dial-peer voice 1008 mmoip
description ***** email trigger incoming dial peer *****
service mapp1
incoming called-number 5556681
!
dial-peer voice 555 voip
incoming called-number 5556248
codec g711ulaw
!
dial-peer voice 5556 pots
description ***** outgoing dial peer to the destination number *****
destination-pattern 5556681
port 0:D
prefix 95556681
!
line con 0
exec-timeout 0 0
password itxctest
line aux 0
password password
no exec
line vty 0
exec-timeout 0 0
password password
no exec
line vty 1 3
line vty 4
exec-timeout 0 0
!
end

```

## VoiceXML Voice Store and Forward Off-Ramp on Cisco 3600 Series Example

```

!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Cisco-3640
!
logging rate-limit console 10 except errors
no logging console
enable secret 5 $1$2DFtyu
enable password sample
!
username test password 0 test123
!
!
voice-card 1
codec complexity high
!
ip subnet-zero
!
ip ftp username test
ip ftp password test123
!

```

```
no ip dhcp-client network-discovery
isdn switch-type primary-5ess
!
!
fax interface-type fax-mail
mta send server 10.10.17.1
mta send subject Test message from Cisco-3640
mta send postmaster test@ultra5.cisco.com
mta send mail-from hostname Cisco-3640
mta send mail-from username test
mta receive aliases [10.10.178.11]
mta receive maximum-recipients 120
!
!
controller T1 1/0
    framing esf
    clock source line primary
    linecode b8zs
    pri-group timeslots 1-24
!
controller T1 1/1
    framing esf
    clock source internal
    linecode b8zs
!
!
interface FastEthernet0/0
    ip address 1.2.178.11 255.255.0.0
    duplex auto
    speed auto
!
interface FastEthernet0/1
    no ip address
    shutdown
    duplex auto
    speed auto
!
interface Serial1/0:23
    no ip address
    no logging event link-status
    isdn switch-type primary-5ess
    isdn incoming-voice voice
    isdn T310 4000
    no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.2.0.1
no ip http server
!
!
snmp-server manager
call rsvp-sync
!
application
    service mapp1 tftp://10.10.17.1/docs/scripts/app_voicemail_offramp.tcl
    param mail-script docs-mapp-test
!
    service docs-mapp-test tftp://10.10.17.1/docs/scripts/offramp/mapp_31.vxml
!
voice-port 1/0:23
!
!
mgcp profile default
!
```

```

dial-peer cor custom
!
!
dial-peer voice 1000 pots
 destination-pattern 5551211
 port 1/0:23
 prefix 5551211
!
dial-peer voice 2000 mmoip
 service mapp1
 incoming called-number 5551211
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password sample
 login
line vty 5 15
 password sample
 login
!
exception core-file Cisco-3640-core
exception protocol ftp
exception dump 10.10.17.1
no scheduler allocate
!
end

```

## Where to Go Next

- To configure properties for audio files, see [“Configuring Audio File Properties for Tcl IVR and VoiceXML Applications” on page 1](#).
- To configure properties for speech recognition or speech synthesis, see [“Configuring ASR and TTS Properties” on page 1](#).
- To configure a VoiceXML fax detection application, see [“Configuring Fax Detection for VoiceXML” on page 1](#).
- To configure telephony call-redirect features for voice applications, see [“Configuring Telephony Call-Redirect Features” on page 1](#).
- To configure session interaction for a Tcl IVR 2.0 application, see [“Configuring Tcl IVR 2.0 Session Interaction” on page 1](#).
- To configure support for SIP and TEL URLs, see [“Configuring SIP and TEL URL Support” on page 245](#).
- To monitor and troubleshoot voice applications, see [“Monitoring and Troubleshooting Voice Applications” on page 1](#).

## Additional References

- [“” on page 1](#)—Describes how to access Cisco Feature Navigator; also lists and describes, by Cisco IOS release, Tcl IVR and VoiceXML features for that release

- [“Overview of Cisco IOS Tcl IVR and VoiceXML Applications” on page 1](#)—Describes underlying Cisco IOS Tcl IVR and VoiceXML technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance

