



# Interactive Voice Response Version 2.0 on Cisco VoIP Gateways

---

## Feature History

Release	Modification
12.1(3)T	This feature was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
CSCds35184	Configuration requirement added to comply with DDTS CSCds35184.
12.2(2)XB	This feature was implemented on the Cisco AS5850 platform.

This feature module describes the new infrastructure to the interactive voice response (IVR) and Tool Command Language (TCL) verbs; the new infrastructure increases scalability and provides enhanced functionality to the Cisco Voice over IP (VoIP) gateways. The Cisco VoIP gateway software now provides IVR post and prepaid application (or the Debit Card feature), and announcement support under both Media Gateway Control Protocol (MGCP) and H.323 protocols. It also utilizes Real Time Streaming Protocol (RTSP), which enables an external RTSP server to play announcements and interact with voice mail servers.

This document includes the following sections:

- Feature Overview, page 2
- Supported Platforms, page 7
- Supported Standards, MIBs, and RFCs, page 8
- Prerequisites, page 8
- Configuration Tasks, page 9
- Configuration Examples, page 15
- Command Reference, page 23
- Debug Commands, page 35
- Glossary, page 59

# Feature Overview

IVR Version 2.0 is the fourth release of IVR and TCL scripting on Cisco IOS VoIP gateways. The Cisco IVR feature (first made available in Cisco IOS Release 12.0(3)T and 12.0(7)T) provides IVR capabilities using TCL scripts.

IVR systems provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words, or more commonly dual tone multifrequency (DTMF) signalling. For example, when a user makes a call with a debit card, an IVR application is used to prompt the caller to enter a specific type of information, such as an account number. After playing the voice prompt, the IVR application collects the predetermined number of touch tones (digit collection), forwards the collected digits to a server for storage and retrieval, and then places the call to the destination phone or system. Call records can be kept and a variety of accounting functions performed.

The IVR application (or script) is a voice application designed to handle calls on a *voice gateway*, which is a router that is equipped with VoIP features and capabilities.

The IVR feature allows an IVR script to be used during call processing. The scripts interact with the IVR software to perform the various functions. Typically, IVR scripts contain both executable files and audio files that interact with the system software.

IVR Version 2.0 is made up of several separate components that are described individually in the sections that follow. These new features are as follows:

- MGCP scripting package implementation
- RTSP client implementation
- IVR prompt playout and digit collection on IP call legs
- New TCL verbs to utilize RTSP and MGCP scripting features

These features add scalability and enable the IVR scripting functionality on VoIP legs. In addition, support for RTSP enables VoIP gateways to play messages from RTSP-compliant announcement servers.

## MGCP Scripting

IVR Version 2.0 infrastructure is greatly enhanced with the addition of support for MGCP using the application package model. MGCP defines application packages to run scripts on the media gateways. These application packages initiate scripts on the gateway and receive return values after execution completes. MGCP scripting allows external call agents (CAs) to instruct a media gateway to run an IVR TCL script in order to perform a specific task and return the end result. For example, you can request and collect the PIN and account number from a caller.

Two Cisco VoIP features previously released that can be implemented are the Debit Card for Packet Telephony and IVR. Both features use the TCL scripting language. The TCL scripts that run with MGCP are written in TCL IVR API Version 2.0 and are able to receive calls through hand off. MGCP scripts can run any TCL command.

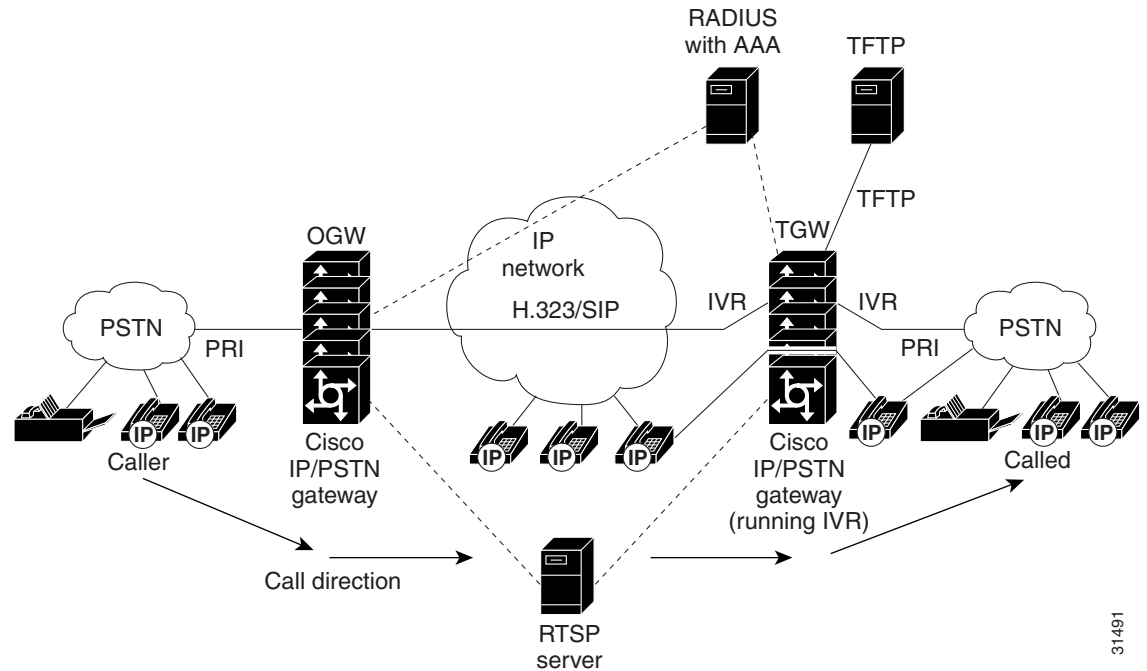
**Note**

For more information about MGCP, see “Media Gateway Control Protocol for the Cisco AS5300 Voice/Gateway” documentation at the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xr/mgcp1206.htm>

Figure 1 displays the CA controlling the IVR scripts. MGCP is the protocol that is running on the CA. The RTSP server is configured to interact with the gateways that have IVR scripts installed and running. The RADIUS server running authentication, authorization, and accounting (AAA) also interacts with the gateways.

**Figure 1** MGCP Control of IVR Scripts



31491

## RTSP Client Implementation

RTSP is an application-level protocol used for control over the delivery of data with real-time properties. RTSP provides an extensive framework to enable control and to perform on-demand delivery of real-time data. For example, RTSP is used to control the delivery of audio streams from an audio server.

By implementing an RTSP client on VoIP gateways, an application running on the gateway can connect calls with audio streams from an external audio server. It also has the following benefits:

- Reduces the CPU load
- Allows larger prompts to be played
- Saves memory on the gateway because no packetization is involved
- Allows use of an external audio server. This external audio server removes the limitation on the number of prompts that can be played out and on the size of the prompt.

## IVR Prompts Played on IP Call Legs

IVR Version 2.0 scripts can be configured for incoming POTS or VoIP call legs to play announcements to the user or collect user input (digits). With IVR Version 2.0 the prompts can be triggered from both the PSTN side of the call leg and the IP side of the call leg. This feature enables the audio files (or prompts) to be played out over the IP network.

IVR scripts played toward a VoIP call leg are subject to the following conditions:

- G.711 mu-law encoding must be used when playing prompts.
- G.711 mu-law encoding must also be used for the duration of these calls, even after prompt playback has completed.
- No DSP is on the IP leg, so the script cannot initiate a tone.
- When an IVR script is used to collect digits on a VoIP call leg, DTMF relay must be used. The following DTMF relay methods are supported:
  - For H.323 protocol configured on the call leg, use DTMF relay method:
    - Cisco Proprietary RTP
    - DTMF Relay via H245 Alphanumeric IE
    - DTMF Relay via H245 Signal IE
  - For SIP protocol configured on the call leg, use Cisco Proprietary RTP



**Note**

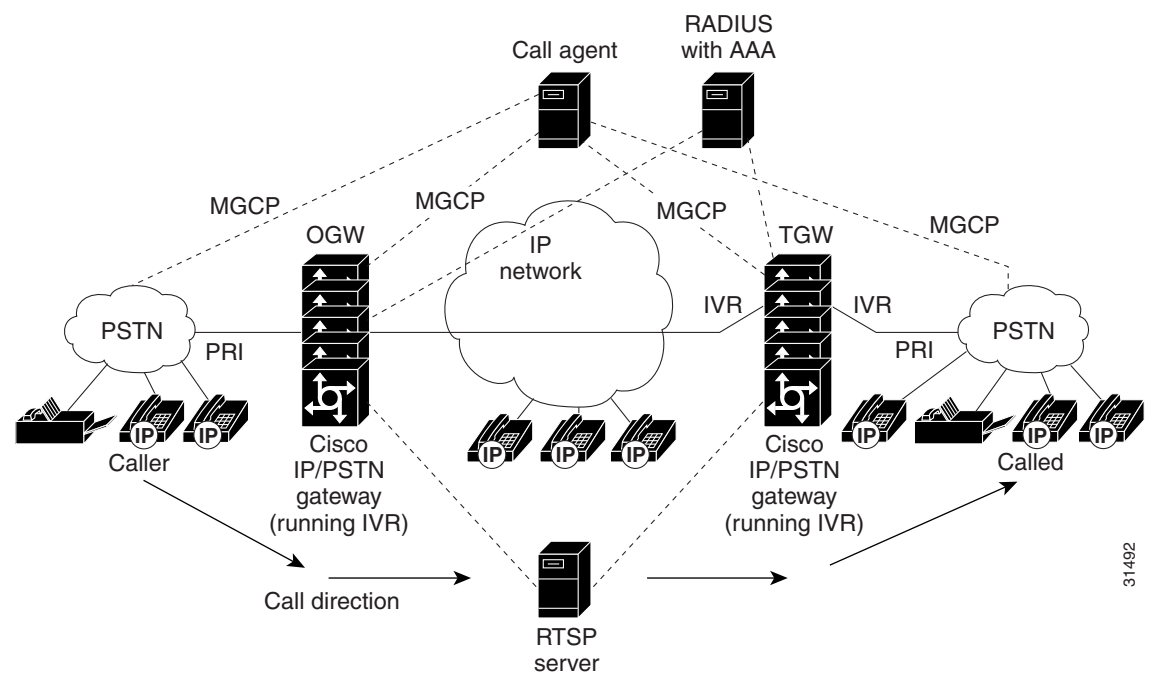
For additional information about the DTMF relay refer to the “Multiservice Application Configuration Guide” in the Cisco IOS 12.1 documentation set at the following URL:

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi\\_c/mcprt1/mcdvoip.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/multi_c/mcprt1/mcdvoip.htm)

With Cisco IOS Release 12.1(3)T, the system can accept calls initiated from the IP side using G.711 and terminate the call to the terminating gateway using the same codec.

Figure 2 displays the IVR application on the gateways controlling the scripts. IP phones can also originate a call to a gateway running an IVR script.

**Figure 2** IVR Control of Scripts on an IP Call Leg



31492

## TCL Verbs

The IVR Version 2.0 feature delivers a new set of TCL verbs and TCL scripts that replace the previous TCL Version 1.0 verbs and scripts. These TCL verbs are used to utilize the RTSP audio servers. The TCL verbs are used to develop TCL scripts that interact with the IVR application. These scripts are used to pass events to the Media Gateway Controller, which is a call agent.

The IVR 2.0 feature is not backward compatible with the IVR 1.0 scripts. The MGCP scripting package can only be implemented using the new IVR 2.0 TCL verbs.


**Note**

For in-depth information about the TCL 2.0 verb set and how to develop scripts, refer to the *TCL IVR API Version 2.0 Programmer's Guide* at the following URL:

[http://www.cisco.com/univercd/cc/td/doc/product/access/acs\\_serv/vapp\\_dev/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/index.htm)

The IVR scripts use the TCL verbs to interact with the gateway during call processing in order to collect the required digits, for example, to request the PIN or account number for the caller. The TCL scripts are the default scripts for all Cisco voice features using IVR, and are secured with a proprietary Cisco locking mechanism or signature. Only Cisco internal technical support personnel are authorized to open and modify these scripts. When a TCL script is activated, the system verifies the Cisco signature level. If the script is inconsistent with the authorized signature level, the script does not load, and the console screen displays an error message.

Refer to the *TCL IVR API Version 2.0 Programmer's Guide* for information about developing TCL scripts. These TCL scripts are configured to control calls coming into or going out of the gateway. The programmer's guide also contains an annotated example of a TCL IVR script and includes instructions for testing and loading TCL IVR scripts.


**Note**

Ensure that you have the correct version of TCL scripts loaded that support this feature. These TCL scripts can be downloaded from the following Cisco Connection Online (CCO) location: <http://www.cisco.com/cgi-bin/tablebuild.pl/tclware>.

## Benefits

- Adds scalability to the IVR infrastructure
- Reduces the CPU load
- Allows larger prompts to be played
- Saves memory on the gateway because no packetization is involved
- Allows use of an external audio server

## Restrictions

The following list of restrictions applies to the combination of features described in the “Feature Overview” section:

- TCL IVR 1.0 verbs and TCL IVR 2.0 verbs cannot be mixed in a script. You either write a script using 1.0 verbs/API or using 2.0 verbs/API.
- H.245-alphanumeric DTMF relay does not accurately report the duration of a key press, for example, holding down the pound (#) key for longer than 1 second to register the “long pound” feature. Doing so only reports a duration of 200 milliseconds. Therefore, if an IVR script is configured on the terminating gateway, Cisco Proprietary RTP or DTMF Relay via H245 Alphanumeric IE must be used.
- RTSP multicast sessions are not supported by the Cisco IOS RTSP client.
- Cisco Proprietary RTP, DTMF Relay via H245 Alphanumeric IE, or DTMF Relay via H245 Signal IE must be configured and negotiated on the VoIP call leg to collect digits over a VoIP call leg.
- RTSP is not recommended for dynamic prompt playouts.

## Related Features and Technologies

- VoIP
- AAA Functionality on the VoIP Gateway
- RADIUS and H.323 Gateway-Specific Accounting
- MGCP

## Related Documents

- *Service Provider Features for Voice over IP*, Cisco IOS Release 12.0(3)T
- *Cisco IOS Multiservice Applications Configuration Guide*:
  - “Configuring Gateways”
  - “Configuring Interactive Voice Response for Cisco Access Platforms”
  - “Configuring Debit Card for Packet Telephony”
- *Media Gateway Control Protocol for the Cisco AS5300 Voice/Gateway*, Cisco IOS Release 12.1(1)T
- *Session Initiation Protocol for Voice over IP on Cisco Access Platforms*, Cisco IOS Release 12.1(1)T
- *TCL IVR API 1.0 Programmer’s Guide*
- *TCL IVR Version 2.0 Programmer’s Guide*, Cisco IOS Release 12.1(3)T

## Supported Platforms

- Cisco AS5300 universal access server
- Cisco AS5850 universal gateway

### Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at [cdbadmin@cisco.com](mailto:cdbadmin@cisco.com). If you want to establish an account on Cisco.com, go to <http://www.cisco.com/register> and follow the directions to establish an account.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

<http://www.cisco.com/go/fn>

## Supported Standards, MIBs, and RFCs

### Standards

No new or modified standards are supported by this feature.

### MIBs

- CISCO-CALL-APPLICATION-MIB

The CISCO-CALL-APPLICATION-MIB supports the management of TCL scripts on the gateway using Simple Network Management Protocol (SNMP). SNMP management support includes monitoring which scripts are loaded, loading a script, and unloading a script.

To obtain lists of MIBs supported by platform and Cisco IOS release and to download MIB modules, go to the Cisco MIB web site on Cisco Connection Online (CCO) at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

### RFCs

- RFC 1889, *Real Time Protocol*
- RFC 2326, *Real Time Streaming Protocol*
- RFC 2327, *Session Description Protocol*

## Prerequisites

Before you can configure your access server platform with this feature, complete the following tasks:

- Ensure that your access platform has a minimum of 16 MB Flash memory and 64 MB DRAM memory.
- Establish a working IP network. For more information about configuring IP, refer to the Cisco IOS 12.1 documentation set. See specifically the *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS Multiservice Applications Configuration Guide*.
- Configure VoIP. For more information about configuring VoIP, refer to the *Voice over IP Software Configuration Guide* for the appropriate access platform.

- Download the TCL scripts required for the IVR feature, which are at the following CCO location:  
<http://www.cisco.com/cgi-bin/tablebuild.pl/tclware>
- Ensure that the VCWare version 7.xx is upgraded on the access server. Download the VCWare at the following CCO location:  
<http://www.cisco.com/cgi-bin/tablebuild.pl/vcware>
- If you are playing out prompts from an RTSP server, you can download a version from Real Networks at the following URL:  
<http://www.real.com>

This free evaluation copy can serve up to 25 simultaneous RTSP sessions. Ensure that the machine running Windows NT or UNIX has sufficient CPU power and memory to serve the requisite number of RTSP sessions.

**Note**

---

A Cisco IP/TV server (version 3.0 or later) can be used as an alternative RTSP server.

---

## Configuration Tasks

Before starting the software configuration tasks for the IVR Version 2.0 features, complete the following preinstallation tasks:

- Download the TCL Scripts and audio files to be used with this feature from the CCO Software Support Center.
- Store the TCL scripts and audio files on a TFTP server configured to interact with your gateway access server.
- Create the IVR/TCL application script to use with the **call application voice** command when configuring IVR using TCL scripts. You create this application first and store it on a server or location where it can be retrieved by the access server.

**Note**

---

When configuring the language priority using the **call application voice language** command, keep in mind that the software is hardcoded for configuring the digit 1 to represent the primary language, and the digit 2 to represent the secondary language.

---

See the following sections for configuration tasks for the IVR Version 2.0 features. Each task in the list is identified as either optional or required:

- Configuring the Application (Required)
- Configuring IVR on the Inbound POTS Dial Peer or Configuring IVR on the Inbound VoIP Dial Peer (Required)
- Configuring MGCP Scripting (Optional)

**Note**


---

When an IVR script is used to detect a “long #” from a caller connected to the H.323 call leg, either Cisco Proprietary RTP or DTMF Relay via H245 Signal IE must be used. DTMF Relay via H245 Alphanumeric IE does not report the actual duration of the digit, causing long pound (#) detection to fail.

---

## Configuring the Application

You must configure the application that interacts with the dial peer before you configure the dial peer. The dial peer collects digits from the caller and uses the application you have created. Use the **call application voice** command as shown in the table below. Each command line is optional depending on the type of action desired or the digits to collect.

	Command	Purpose
Step 1	Router# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Router(config)# <b>call application voice</b> <i>name url</i>	<p>Defines the name of the application to be used with your IVR script, and indicates the location of the appropriate IVR script to be used with this application.</p> <p>URL designates where to look for the file and the protocol to use to access it.</p> <p>Examples:</p> <pre>flash:scripts/session.tcl tftp://dirt/sarvi/scripts/session.tcl ftp://sarvi-ultra/scripts/session.tcl slot0:scripts/tcl/session..tcl</pre> <p> <b>Note</b> You can only configure a URL if the application named <i>name</i> has <i>not</i> been configured.</p>
Step 3	Router(config)# <b>call application voice</b> <i>name language</i> <i>digit language</i>	<p>The number that identifies the language used by the audio files. For example, enter <b>1</b> for English language or <b>2</b> for Spanish language audio files. Any number can represent any language.</p> <p>Enter: <b>language</b> <i>digit language</i>.</p> <p>For example: <code>call application voice test language 1 en</code></p> <p>Parameter values are:</p> <ul style="list-style-type: none"> <li><i>digit</i> — 0 through 9</li> <li><i>language</i>—Two characters representing the language. For example, en for English, sp for Spanish, and ch for Mandarin. Enter <b>aa</b> to represent all.</li> </ul>
Step 4	Router(config)# <b>call application voice</b> <i>name</i> <b>pin-length</b> <i>number</i>	Defines the number of characters in the PIN for the designated application. Values are 0 through 10.
Step 5	Router(config)# <b>call application voice</b> <i>name</i> <b>retry-count</b> <i>number</i>	Defines the number of times a caller is permitted to reenter the PIN for the designated application. Values are 1 through 5.

	Command	Purpose
Step 6	Router(config)# <b>call application voice name</b> <b>uid-length number</b>	Defines the number of characters allowed to be entered for the user ID for the designated application.  Values are 1 through 20.
Step 7	Router(config)# <b>call application voice name</b> <b>set-location language category location</b>	Defines the location, language, and category of the audio files for the designated application.  For example, <b>set-location en 1 tftp://server/dir/</b>  Parameter values are: <ul style="list-style-type: none"> <li>category - 0 through 4, 0 represents all 1-4 categories. Category 1 represents numbers, category 2 represents units, category 3 represents time &amp; date, category 4 represents all others.</li> <li>location - A url represents the location of this category.</li> </ul>

## Configuring IVR on the Inbound POTS Dial Peer



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

	Command	Purpose
Step 1	Router(config)# <b>aaa new-model</b>	Enables AAA security and accounting services.
Step 2	Router(config)# <b>gw-accounting h323</b>	Enables gateway-specific H.323 accounting.
Step 3	Router(config)# <b>aaa authentication login h323 radius</b>	Defines a method list called H.323 where RADIUS is defined as the only method of login authentication.
Step 4	Router(config)# <b>aaa accounting connection h323 start-stop radius</b>	Defines a method list called H.323 where RADIUS is used to perform connection accounting, providing start-stop records.
Step 5	Router(config)# <b>radius-server host ip-address</b> <b>auth-port number acct-port number</b>	Identifies the RADIUS server and the ports that will be used for authentication and accounting services.
Step 6	Router(config)# <b>radius-server key key</b>	Specifies the password used between the gateway and the RADIUS server.
Step 7	Router(config)# <b>dial-peer voice number pots</b>	Enters dial-peer configuration mode to configure the incoming POTS dial peer.  The <i>number</i> value of the <b>dial-peer voice pots</b> command is a tag that uniquely identifies the dial peer.
Step 8	Router(dial-peer)# <b>application name</b>	Associates the IVR application with the incoming POTS dial peer. Enter the selected IVR/TCL application name.

	Command	Purpose
Step 9	Router(config-dial-peer)# <b>destination-pattern</b> <i>pattern</i>	<p>Enters the telephone number associated with this dial peer. The <i>pattern</i> is a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0–9 and the letters A–D. The following special characters can be entered in the string:</p> <ul style="list-style-type: none"> <li>• The plus symbol (+) can be used to indicate an E.164 standard number. On the Cisco MC3810, the plus symbol (+) is not a valid character in the string.</li> <li>• The asterisk character (*) and the pound sign (#) that appear on standard touch-tone dial pads can be used in any dial string. However, these characters cannot be used as leading characters in a string (for example, *650).</li> <li>• The period (.) can be used as a trailing character, and is used as a wildcard character. Multiple periods as trailing characters indicate multiple wildcard digits, such as for the 789... wildcard.</li> <li>• The comma (,) can be used only in prefixes, and is used to insert a 1-second pause or a delay.</li> </ul> <p>The timer (T) character can be used to configure variable length dial plans.</p>
Step 10	Router(config-dial-peer)# <b>session target</b> <i>ipv4:1.13.7x.x</i>	Specifies the session target IP address.

## Configuring IVR on the Inbound VoIP Dial Peer

	Command	Purpose
Step 1	Router # <b>configure terminal</b>	Enters global configuration mode.
Step 2	Router(config)# <b>dial-peer voice</b> <i>4401</i> <i>voip</i>	Enters the dial-peer configuration mode and identifies the call leg.
Step 3	Router(config-dial-peer)# <b>application</b> <i>application-name</i>	Specifies the name of the application and script to use.
Step 4	Router(config-dial-peer)# <b>destination-pattern</b> <i>pattern</i>	Enters the destination pattern.
Step 5	Router(config-dial-peer)# <b>session protocol</b> <i>sipv2</i>	<p>Specifies the session protocol.</p> <p>The default session protocol is H.323. Configure <b>sipv2</b> to enable SIP.</p>
Step 6	Router(config-dial-peer)# <b>session target</b> <i>ipv4:1.13.7x.x</i>	Specifies the session target IP address.

	Command	Purpose
Step 7	Router(config-dial-peer)# <b>dtmf-relay cisco-rtp</b>	<p>Specifies the DTMF relay method.</p> <p>Other options include:</p> <ul style="list-style-type: none"> <li>• <b>cisco-rtp</b>: Available for H.323 and SIP.</li> <li>• <b>h245-alphanumeric</b> and <b>h245-signal</b>: Available for H.323 only.</li> <li>• <b>no dtmf-relay</b>: Default.</li> </ul> <p> <b>Note</b> If digit collection from this VoIP call leg is needed, the <b>dtmf-relay</b> keyword is required.</p>
Step 8	Router(config-dial-peer)# <b>codec g711ulaw</b>	<p>Specifies the voice codec to use.</p> <p> <b>Note</b> If the configured application will be playing prompts to the VoIP call leg, the <b>g711ulaw</b> keyword is required.</p>

## Configuring MGCP Scripting

The MGCP script package needs to be enabled to perform MGCP scripting. Enable the script in global configuration mode by entering the **mgcp package-capability script package** command.

The example MGCP configuration shown in this section is for DS0s on T1 lines. The configuration tasks are as follows:

- Enabling the MGCP service on the DS0 groups
- Enabling the other MGCP packages
- Configuring the call agent address and other MGCP parameters

To configure MGCP scripting, perform the following steps in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>mgcp</b>	Starts the MGCP daemon.
Step 2	Router(config)# <b>mgcp request timeout</b> <i>timeout</i>	Specifies how long the gateway should wait for a response to a request.
Step 3	Router(config)# <b>mgcp request retries</b> <i>count</i>	Specifies the number of times to retry sending the <b>mgcp</b> command.
Step 4	Router(config)# <b>mgcp call-agent</b> { <i>ipaddr</i>   <i>hostname</i> } [ <i>port</i> ]	Configures the address of the call agent.

	Command	Purpose
Step 5	Router(config)# <b>mgcp max-waiting-delay</b> <i>value</i>	Configures the maximum waiting delay to be used in a restart in progress (RSIP) message as restart instructions for the call agent.
Step 6	Router(config)# <b>mgcp restart-delay</b> <i>value</i>	Configures the restart delay value to be used in an RSIP message as graceful teardown instructions for the gateway connection.
Step 7	Router(config)# <b>mgcp vad</b>	Configure voice activity detection.
Step 8	Router(config)# <b>mgcp package-capability</b> { <b>as-package</b>   <b>dtmf-package</b>   <b>gm-package</b>   <b>rtp-package</b>   <b>trunk-package</b> }	Specifies an MGCP package capability.
Step 9	Router(config)# <b>mgcp default-package</b> { <b>as-package</b>   <b>dtmf-package</b>   <b>gm-package</b>   <b>rtp-package</b>   <b>trunk-package</b> }	Configures the default package capability type.
Step 10	Router(config)# <b>mgcp quality-threshold</b> { <b>hwm-jitter-buffer</b> <i>value</i>   <b>hwm-latency</b> <i>value</i>   <b>hwm-packet-loss</b> <i>value</i>   <b>lwm-jitter-buffer</b> <i>value</i>   <b>lwm-latency</b> <i>value</i>   <b>lwm-packet-loss</b> <i>value</i> }	Configures the jitter buffer size, packet-loss threshold, and latency threshold.
Step 11	Router(config)# <b>mgcp playout</b> { <b>adaptive</b> <i>init-value min-value max-value</i> }   { <b>fixed</b> <i>init-value</i> }	Tunes the jitter buffer packet size used for MGCP connections.
Step 12	Router(config)# <b>mgcp codec</b> <i>type</i> [ <b>packetization-period</b> <i>value</i> ]	Configures the default codec type.
Step 13	Router(config)# <b>mgcp ip-tos</b> { <b>high-reliability</b>   <b>high-throughput</b>   <b>low-cost</b>   <b>low-delay</b>   <b>precedence</b> <i>value</i> }	Enables the IP type of service for MGCP connections.
Step 14	Router(config)# <b>controller t1 slot#</b>	Uses the controller configuration mode for the T1 controller in the specified slot.
Step 15	Router(config-controller)# <b>framing</b> <i>type</i>	Configures the framing type.
Step 16	Router(config-controller)# <b>clock source</b> <i>type</i>	Configures the clock source.
Step 17	Router(config-controller)# <b>linecode</b> <i>type</i>	Configures the line coding.
Step 18	Router(config-controller)# <b>ds0-group</b> <i>n timeslots range</i> <b>type</b> <i>signaling-type</i> <b>service mgcp</b>	Configures the DS0s to support MGCP.

## Verifying the Configuration

- Step 1** Enter the **show call application voice summary** command to verify that the newly created applications are listed. The example output follows:

```
Router# show call application voice summary

name                description
-----
DEFAULT             NEW::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
SESSION             Default system session application
hotwo               tftp://hostname/scripts/nb/nb_handoffTwoLegs.tcl
hoone               tftp://hostname/scripts/nb/nb_dohandoff.tcl
hodest              tftp://hostname/scripts/nb/nb_handoff.tcl
clid                tftp://hostname/scripts/tcl_ivr/clid_authen_collect.tcl
db102               tftp://hostname/scripts/1.02/debitcard.tcl
*hw                 tftp://171.69.184.xxx/tr_hello.tcl
*hw1                tftp://san*tr_db
tftp://171.69.184.235/tr_debitcard.answer.tcl

TCL Script Version 2.0 supported.
TCL Script Version 1.1 supported.

Router#
```



**Note** In the output shown, an asterisk (\*) in an application indicates that this application was not loaded successfully. Use the **show call application voice name** command to view information for a particular application.

- Step 2** Enter the **show dial-peer voice peer tag** command and verify that the application associated with the dial peer is correct.
- Step 3** Enter the **show running-config** command to display the entire configuration.

## Configuration Examples

Use the **show running-config** command to display the entire gateway configuration. Figure 3 shows the type of topology used in the configuration for the example.

**Figure 3** Example Configuration Topology



In this example configuration, GW1 is running IVR for phone A, and GW2 is running IVR for phone B. This section provides the following configuration examples:

- GW1 IVR Configuration Example
- GW2 IVR Configuration Example
- MGCP Scripting Configuration Example

## GW1 IVR Configuration Example

The following output is the result of using the **show running-config** command:

```

GW1
Router# show running-config

Building configuration...

Current configuration:
!
! Last configuration change at 08:39:29 PST Mon Jan 10 2000 by lab
!
version 12.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GW1
!
logging buffered 100000 debugging
aaa new-model
aaa authentication login default local group radius
aaa authentication login h323 group radius
aaa authentication login con none
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
enable password xxx
!
username lab password 0 lab
!
!
resource-pool disable
!
!
clock timezone PST -8
ip subnet-zero
ip host baloo 1.14.124.xxx
ip host dirt 223.255.254.254
ip host rtspserver3 1.14.1xx.2
ip host rtspserver1 1.14.1xx.2
!
mgcp package-capability trunk-package
mgcp default-package trunk-package
isdn switch-type primary-net5
isdn voice-call-failure 0
!
!
```

```
tftp://dirt/hostname/WV/en_new/
call application voice debit_card tftp://dirt/Router/scripts.new/app_debitcard.tcl
call application voice debit_card uid-len 6
call application voice debit_card language 1 en
call application voice debit_card language 2 ch
call application voice debit_card set-location ch 0 tftp://dirt/hostname/WV/ch_new/
call application voice debit_card set-location en 0 tftp://dirt/hostname/WV/en_new/
call application voice debit_card_rtsp tftp://dirt/IVR 2.0/scripts.new/app_debitcard.tcl
call application voice debit_card_rtsp uid-len 6
call application voice debit_card_rtsp language 1 en
call application voice debit_card_rtsp language 2 ch
call application voice debit_card_rtsp set-location ch 0 rtsp://rtspserver1:554/
call application voice debit_card_rtsp set-location en 0 rtsp://rtspserver1:554/

mta receive maximum-recipients 0
!
!
controller E1 0
  clock source line primary
  pri-group timeslots 1-31
!
controller E1 1
!
controller E1 2
!
controller E1 3
!
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
!
interface Ethernet0
  ip address 1.14.128.35 255.255.255.xxx
  no ip directed-broadcast
  h323-gateway voip interface
  h323-gateway voip id gk1 ipaddr 1.14.128.19 1xxx
  h323-gateway voip h323-id gw1@cisco.com
  h323-gateway voip tech-prefix 5#
!
interface Serial0:15
  no ip address
  no ip directed-broadcast
  isdn switch-type primary-net5
    isdn incoming-voice modem
  fair-queue 64 256 0
  no cdp enable
!
!
interface FastEthernet0
  ip address 16.0.0.1 255.255.xxx.0
  no ip directed-broadcast
  duplex full
  speed auto
  no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.14.128.33
ip route 1.14.xxx.0 255.xxx.255.xxx 16.0.0.2
ip route 1.14.xxx.16 255.xxx.255.240 1.14.xxx.33
no ip http server
!
!
```

```
radius-server host 1.14.132.2 auth-port 1645 acct-port 1646
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!
voice-port 0:D
  cptone DE
!
!
dial-peer voice 200 voip
  incoming called-number 53
  destination-pattern 34.....
  session target ipv4:16.0.0.2
  dtmf-relay h245-alphanumeric
  codec g711ulaw
!
dial-peer voice 102 pots
  application debit_card_rtsp
  incoming called-number 3450072
  shutdown
  destination-pattern 53.....
  port 0:D
!
!
dial-peer voice 202 voip
  shutdown
  destination-pattern 34.....
  session protocol sipv2
  session target ipv4:16.0.0.2
  dtmf-relay cisco-rtp
  codec g711ulaw
!
dial-peer voice 101 pots
  application debit_card
  incoming called-number 3450070
  destination-pattern 53.....
  port 0:D
!
!
gateway

!
line con 0
  exec-timeout 0 0
  transport input none
line aux 0
line vty 0 4
  password xxx
!
ntp clock-period 17180740
ntp server 1.14.42.23
end

GW1#
```

## GW2 IVR Configuration Example

The following output is the result of using the **show running-config** command:

```
GW2#
Router# show running-config

Building configuration...

Current configuration:
!
! Last configuration change at 08:41:12 PST Mon Jan 10 2000 by lab
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname GW2
!
logging buffered 100000 debugging
aaa new-model
aaa authentication login default local group radius
aaa authentication login h323 group radius
aaa authentication login con none
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
!
username lab password xxx
username 111119 password xxx
!
!
resource-pool disable
!
!
!
!
clock timezone PST -8
ip subnet-zero
ip host radiusserver2 1.14.132.2
ip host radiusserver1 1.14.138.11
ip host baloo 1.14.124.254
ip host rtspserver2 1.14.136.2
ip host dirt 223.255.254.254
ip host rtspserver3 1.14.126.2
!
mgcp package-capability trunk-package
mgcp default-package trunk-package
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
!
call application voice clid_authen_sky
tftp://dirt/hostname/sky_scripts/clid_authen_collect_cli_sky.tcl

call application voice rtsp_demo tftp://dirt/hostname/sky_scripts/rtsp_demo.tcl
tftp://dirt/hostname/WV/en_new/
call application voice debit_card tftp://dirt/IVR 2.0/scripts.new/app_debitcard.tcl
call application voice debit_card uid-len 6
call application voice debit_card language 1 en
call application voice debit_card language 2 ch
call application voice debit_card set-location ch 0 tftp://dirt/hostname/WV/ch_new/
```

```

call application voice debit_card set-location en 0 tftp://dirt/hostname/WV/en_new/
call application voice clid_authen_rtsp tftp://dirt/IVR
2.0/scripts.new/app_clid_authen_collect_cli_rtsp.tcl

call application voice clid_authen_rtsp location rtsp://rtspserver2:554/
call application voice clid_authen1 tftp://dirt/IVR
2.0/scripts.new/app_clid_authen_collect_cli_rtsp.tcl
call application voice clid_authen1 location tftp://dirt/hostname/WV/en_new/
call application voice clid_authen1 uid-len 6
call application voice clid_authen1 retry-count 4
mta receive maximum-recipients 0
!
!
controller T1 0
  framing esf
  clock source line primary
  linecode b8zs
  pri-group timeslots 1-24
!
controller T1 1
  clock source line secondary 1
!
controller T1 2
!
controller T1 3
!
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
!
interface Ethernet0
  ip address 1.14.xxx.4 255.255.xxx.240
  no ip directed-broadcast
  h323-gateway voip interface
  h323-gateway voip id gk2 ipaddr 1.14.xxx.18 1719
  h323-gateway voip h323-id gw2@cisco.com
  h323-gateway voip tech-prefix 3#
!
interface Serial0:23
  no ip address
  no ip directed-broadcast
  isdn switch-type primary-5ess
  isdn incoming-voice modem
  fair-queue 64 256 0
  no cdp enable
!
interface FastEthernet0
  ip address 16.0.0.2 255.xxx.255.0
  no ip directed-broadcast
  duplex full
  speed 10
  no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.14.xxx.5
ip route 1.14.xxx.32 255.255.xxx.240 16.0.0.1
no ip http server
!
!
radius-server host 1.14.132.2 auth-port 1645 acct-port 1646
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication

```

```
!  
voice-port 0:D  
!  
dial-peer voice 100 voip  
  application debit_card  
  incoming called-number 34  
  shutdown  
  destination-pattern 53.....  
  session target ras  
  dtmf-relay h245-alphanumeric  
  codec g711ulaw  
!  
dial-peer voice 200 pots  
  incoming called-number 30001  
  destination-pattern 3450070  
  port 0:D  
  prefix 50070  
!  
dial-peer voice 101 voip  
  application debit_card  
  incoming called-number 34.....  
  shutdown  
  session protocol sipv2  
  session target ipv4:16.0.0.1  
  dtmf-relay cisco-rtp  
  codec g711ulaw  
!  
dial-peer voice 102 voip  
  incoming called-number 34.....  
  destination-pattern 53.....  
  session target ipv4:16.0.0.1  
  dtmf-relay h245-alphanumeric  
  codec g711ulaw  
!  
gateway  
!  
line con 0  
  exec-timeout 0 0  
  transport input none  
line aux 0  
line vty 0 4  
  password xxx  
!  
ntp clock-period 17180933  
ntp server 1.14.42.23  
end  
  
GW2#
```

## MGCP Scripting Configuration Example

The following example displays only the MGCP specific portion of the configuration:

```
!  
mgcp  
mgcp request timeout 10000  
mgcp request retries 1  
mgcp call-agent 1.14.138.11  
mgcp restart-delay 10  
mgcp codec g723ar63 packetization-period 30  
mgcp vad
```

```
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability trunk-package
mgcp package-capability rtp-package
mgcp package-capability as-package
mgcp package-capability script-package
mgcp default-package trunk-package
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
!
mta receive maximum-recipients 0
!
!
controller T1 0
  framing esf
  clock source line primary
  linecode b8zs
  ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 1
  framing esf
  clock source line secondary 1
  linecode b8zs
  ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 2
  framing esf
  linecode b8zs
  ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 3
  framing esf
  linecode b8zs
  ds0-group 0 timeslots 1-24 type none service mgcp
!
!
end
```

# Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.1(3) T command reference publications.

- **call application cache reload time**
- **call application voice**
- **call application voice load**
- **rtsp client session history duration**
- **rtsp client session history records**
- **show rtsp client session**

# call application cache reload time

To configure the router to reload the Media Gateway Control Protocol scripts from cache on a regular interval, use the **call application cache reload time** global configuration command. To set the value to the default use the **no** form of this command.

**call application cache reload time** *bg-minutes*

**no call application cache reload time**

## Syntax Description

*bg-minutes*

Specifies the number of minutes to wake up the background process. This background process checks the time elapsed since the script has been used, and if the script is current:

- If the script has not been used in the last “unload time,” it will unload the script and quit. The unload time is not configurable.
- If the script has been used, the background process will load the script from the URL. It compares the scripts and if they do not match, begins using the new script for new calls.

## Defaults

30 minutes

## Command Modes

Global configuration

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Examples

The following example displays the **call application cache reload time** command configured to specify 30 minutes before a background process is wakened:

```
call application cache reload time 30
```

## Related Commands

Command	Description
<b>call application voice load</b>	Allows reload of an application that was loaded via the MGCP scripting package.
<b>show call application voice</b>	Displays all TCL or MGCP scripts that are loaded.

# call application voice

To set the parameter values to be used with the IVR feature, use the **call application voice** command global configuration. To set the value to the default use the **no** form of this command.

**call application voice** *name* {*url* | {*param*} *value1*[*value2*]}

**no call application voice** *name* {*url* | {*param*} *value1* [*value2*]}

## Syntax Description


<i>name</i>	The name of the application to which the parameters are being passed. You can only configure the <i>parameter</i> if the application named <i>name</i> has been configured.
<i>url</i>	Designates where to look for the file and the protocol to use to access it. Examples: <pre>flash:scripts/session.tcl tftp://dir/sarvi/scripts/session.tcl ftp://sarvi-ultra/scripts/session.tcl slot0:scripts/tcl/session..tcl</pre>
	 <b>Note</b> You can only configure a URL if the application named <i>name</i> has <i>not</i> been configured.
<i>parameter</i>	Name of the parameter that is being passed. See Table 1.
<i>value1</i>	Value of the first parameter being passed.
<i>value2</i>	Value of the second parameter being passed.

Table 1 Valid Parameters

<i>language</i>	<p>The number that identifies the language used by the audio files. For example, enter <b>1</b> for English language or <b>2</b> for Spanish language audio files. Any number can represent any language.</p> <p>Enter: <b>language digit language</b>.</p> <p>For example: <code>call application voice test language 1 en</code></p> <p>Parameter values are:</p> <ul style="list-style-type: none"> <li><i>number</i>—Tag that uniquely identifies an audio file. Enter 1 to indicate the primary language, and to 2 indicate the secondary language. Valid entries are from 0 - 9. (The digits 1 and 2 are hard coded to represent primary and secondary languages.)</li> <li><i>language</i>—Two characters representing the language. For example, en for English, sp for Spanish, and ch for Mandarin. Enter <b>aa</b> to represent all.</li> </ul>
<i>number</i>	<p>The designated operator telephone number of the service provider (or any other number designated by the service provider). This is the number that calls are terminated to when debit time allowed has run out or the debit amount is exceeded.</p>
<i>pin-length</i>	<p>The number of characters in the PIN number (0–10).</p>
<i>retry-count</i>	<p>The number of times the caller is permitted to enter digits.</p> <p>Enter <b>retry-count number</b> of retries (1–5).</p>
<i>set-location</i>	<p>The location where the audio files are stored. This command connects both the <i>category</i> and <i>language</i> arguments that are used with the <b>call application voice</b> command.</p> <p>Enter: <b>set-location language category audio files URL or directory</b></p> <p>Parameter values are:</p> <ul style="list-style-type: none"> <li><i>location</i>: Where audio files are stored. Enter the TFTP server location.</li> <li><i>language</i>: en = English, ch = Mandarin, sp = Spanish</li> <li><i>category</i>: 0 = numbers, 1 = units, 2 = days, months, and so on, 3 = others</li> <li>audio file URL or directory in the TFTP server</li> </ul> <p>The audio files can be divided into category groups (0–4). For example, audio files representing the days and months can be category 1, audio files representing units of currency can be category 2, audio files representing units of time: seconds, minutes, and hours can be category 3.</p> <p>For example: <code>set-location en 1 tftp://keyer/debitaudio/</code></p>
<i>uid-length</i>	<p>The number of characters allowed to be entered for the user ID or PIN. (1–20)</p>
<i>warning-time</i>	<p>The number of seconds the user is warned before the allowed calling time has run out.</p> <p>Enter <b>warning-time number</b> of seconds. (10–600)</p>
<i>word</i>	<p>Specifies a generic word that will be passed into the script (opaque strings to Cisco IOS software).</p>

**Defaults**

Each parameter must be defined for the system interaction with the voice prompt. See the command syntax description for individual parameter defaults.

**Command Modes**

Dial-peer configuration.

**Command History**

Release	Modification
12.0(4)XH	This command was introduced.
12.0(7)T	The URL, <i>uid-length the parameter</i> , and <i>pin-length</i> arguments were added.
12.1(3)T	The generic <i>word</i> argument was added for an AV pair.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

**Usage Guidelines**

The capability for a generic AV pair is added in Cisco IOS Release 12.1(3)T.

This command is used with the Debit Card feature and when an application is created to interact with IVR and TCL scripts.

The set of parameters, or keywords, is displayed only if the URL for the defined application is a TCL script, determined with the extension.tcl on the URL.

**Examples**

- The following example shows syntax entered to configure the URL location, using category 2, English language) audio files:

```
call application voice debitcard set-location en 2
tftp://keyer/faryaman/debitcard
```

- The following example shows syntax entered to configure the URL for all categories, all languages audio files:

```
call application voice debitcard set-location aa 0
tftp://keyer/debitaudio/
```

- The following example shows a sequence of commands entered during configuration of an application:

```
Router(config)# call application voice prepaid tftp://keyer/debitcard.tcl
```

```
Router(config)# call application voice prepaid uid-len 4
```

```
Router(config)# call application voice prepaid language 1 en
```

```
Router(config)# call application voice prepaid language 2 sp
```

```
Router(config)# call application voice prepaid set-location en 0 tftp://keyer/
```

```
Router(config)# call application voice prepaid pin-len 4
```

```
Router(config)# call application voice prepaid retry-count 3
```

```
Router(config)# call application voice prepaid uid-len 10
```

## call application voice

- The following example shows syntax entered to call the application named my\_app that is located on the TFTP server host named joe. The name of the TCL script is script.tcl.

```
call app voice my_app tftp://server/joe/script.tcl
call app voice my_app num-calls-allowed unlimited
```

### Related Commands

Command	Description
<b>call application cache reload time</b>	Configures the interval for reloading MGCP scripts.
<b>call application voice load</b>	Allows reload of an application that was loaded via the MGCP scripting package.
<b>show call application voice</b>	Displays all TCL or MGCP scripts that are loaded.

# call application voice load

To reload the selected tool command line script from the URL, use the **call application voice load** privileged EXEC command.

**call application voice load** *name*

<b>Syntax Description</b>	<i>name</i>	Defines the TCL script to use for the call. Enter the name of the TCL or MGCP script you want this dial peer to use.
---------------------------	-------------	--

**Defaults** None. TCL or scripts are not loaded.

**Command Modes** Privileged EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.0(7)T	This command was introduced.
	12.1(3)T	Dynamic script loading of MGCP scripts.
	12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
	12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

**Usage Guidelines** The software checks the signature lock to ensure it is a Cisco- supported TCL script. call application voice load *name*

If the TCL script does not have a valid Cisco supported signature, the software fails to load the script and generates the following error message:

```
00:02:54: %IVR-3-BAD_IVR_SIG: Script signature is invalid
```

**Examples** The following example shows syntax entered to load a MGCP script package:

```
call application voice load mgcp-script-pkg
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>call application voice</b>	Creates and then calls the application that will interact with the IVR feature.
	<b>show call application voice</b>	Displays a list of the voice applications that are configured.
	<b>call application cache reload time</b>	Configures the interval for reloading MGCP scripts.

## rtsp client session history duration

To specify the length of time to keep Real Time Streaming Protocol session history records in memory, use the **rtsp client session history duration** global configuration command. To set the value to the default use the **no** form of this command.

**rtsp client session history duration** *number*

**no rtsp client session history duration** *number*

### Syntax Description

<i>number</i>	Specifies the number of minutes to keep the record.
---------------	---

### Defaults

10 minutes

### Command Modes

Global configuration

### Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

### Examples

The following example sets the RTSP session history to 500 minutes:

```
rtsp client session history duration 500
```

### Related Commands

Command	Description
<b>call application voice load</b>	Allows reload of an application that was loaded via the MGCP scripting package.
<b>rtsp client session history records</b>	Specifies the number of RTSP client session history records kept during the session.
<b>show call application voice</b>	Displays all TCL or MGCP scripts that are loaded.
<b>show rtsp client session</b>	Displays cumulative information about the RTSP session records.

# rtsp client session history records

To configure the number of records to keep in the RTSP client session history, use the **rtsp client session history records** global configuration command. To set the value to the default use the **no** form of this command.

**rtsp client session history records** *number*

**no rtsp client session history records** *number*

<b>Syntax Description</b>	<i>number</i>	Specifies the number of records to retain a session history record. Values = 1–100,000.
---------------------------	---------------	--

<b>Defaults</b>	50 records
-----------------	------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.1(3)T	This command was introduced.
	12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
	12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

<b>Examples</b>	The following example sets the RTSP client history to 500 records: <pre>rtsp client session history record 500</pre>
-----------------	---

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>show call application voice</b>	Displays all TCL or MGCP scripts that are loaded.
	<b>call application voice load</b>	Allows reload of an application that was loaded via the MGCP scripting package.
	<b>rtsp client session history duration</b>	Specifies the length of time the RTSP is kept during the session.

# show rtsp client session

To display cumulative information about Real Time Streaming Protocol session records, use the **show rtsp client session** privileged EXEC command. To set the value to the default use the **no** form of this command.

**show rtsp client session** {**history** | **active**} [**detailed**]

**no show rtsp client session** {**history** | **active**} [**detailed**]



## Note

“Session” refers to a session between the application and the RTSP client. Each call leg that is configured to use RTSP streaming has a session.

## Syntax Description

<b>history</b>	Displays cumulative information about the session, packet statistics, and general call information such as call ID, session ID, individual RTSP stream URLs, packet statistics, and play duration.
<b>active</b>	If <b>detailed</b> option is not specified, the session information and stream information for the stream that is currently active.
<b>detailed</b>	If the <b>detailed</b> flag is set, the <b>show</b> command displays the session information and stream information in detail for all streams that are associated with the session.

## Defaults

Active (current) stream information is displayed.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Usage Guidelines

Use of this command displays cumulative information about the session is displayed, packet statistics, and general call information such as call ID, session ID, and so on.

A call leg could play several prompts in a session; the “Play Time” refers to the play time associated with a stream or, in other words, a prompt; the cumulative play time is the sum total of all streams (or prompts) played out in a session.

The command output is a stream block that contains information about the stream, (URL, packet statistics, current state of the stream, play duration, call ID, session ID, individual RTSP stream URLs, and packet statistics).

**Examples**

The following output is displayed when the **show rtsp client session active** command is used during an active session:

```
Router# show rtsp client session active

RTSP Session ID:0x8      Current Status:RTSP_STATUS_PLAYING
Associated CallID:0xF
Active Request:RTSP_API_REQ_PLAY
Control Protocol:TCP      Data Protocol:RTP

Total Packets Transmitted:0 (0 bytes)
Total Packets Received:708 (226560 bytes)

Cumulative Elapsed Play   Time:00:00:28.296
Cumulative Elapsed Record Time:00:00:00.000

      Session ID:0x8      State:ACTIVE
      Local IP Address:1.13.79.45      Local Port 16660
      Server IP Address:1.13.79.6      Server Port 11046
      Stream URL:rtsp://rtsp-cisco.cisco.com:554/chinna.au/streamid=0

      Packets Transmitted:0 (0 bytes)
      Packets Received:708 (226560 bytes)

      Elapsed Play   Time:00:00:28.296
      Elapsed Record Time:00:00:00.000
      ReceiveDelay:85      LostPackets:0

Router#
```

The following output is displayed when the **show rtsp client session history detailed** command is used:

```
Router# show rtsp client session history detailed

RTSP Session ID:0x8
Associated CallID:0xF
Control Protocol:TCP      Data Protocol:RTP

Total Packets Transmitted:0 (0 bytes)
Total Packets Received:2398 (767360 bytes)

Cumulative Elapsed Play   Time:00:01:35.916
Cumulative Elapsed Record Time:00:00:00.000

      Session ID:0x8      State:INACTIVE
      Local IP Address:1.13.79.45      Local Port 16660
      Server IP Address:1.13.79.6      Server Port 11046
      Stream URL:rtsp://rtsp-cisco.cisco.com:554/chinna.au/streamid=0

      Packets Transmitted:0 (0 bytes)
      Packets Received:2398 (767360 bytes)

      Play   Time:00:01:35.916
      Record Time:00:00:00.000
      OnTimeRcvPlayout:93650
      GapFillWithSilence:0
      GapFillWithPrediction:70
      GapFillWithInterpolation:0
      GapFillWithRedundancy:0
      HighWaterPlayoutDelay:85
      LowWaterPlayoutDelay:64
      ReceiveDelay:85      LostPackets:0
      EarlyPackets:2      LatePackets:12

rtsp-5#
Router#
```

## ■ show rtsp client session

Related Commands	Command	Description
	<b>rtsp client session history duration</b> <i>number</i>	Specifies the length of time the RTSP is kept during the session.
	<b>rtsp client session history records</b> <i>number</i>	Specifies the number of RTSP client session history records during the session.

# Debug Commands

This section documents the new **debug** commands related to the RTSP and all other commands used with feature are documented in the Cisco IOS Release 12.1(3)T.

- **debug rtsp**
- **debug rtsp api**
- **debug rtsp client session**
- **debug rtsp pmh**
- **debug rtsp socket**
- **debug voip ivr**

# debug rtsp

To show the status of the Real Time Streaming Protocol client server, use the **debug rtsp** command. To disable the display of output use the **no** form of this command.

```
debug rtsp type [all | api | pmh | session | socket]
```

```
[no] debug rtsp type [all | api | pmh | session | socket]
```

## Syntax Description

<b>all</b>	Optional displays debug messages for all RTSP client debug trace.
<b>api</b>	Optional displays debug output for the RTSP client API.
<b>pmh</b>	Optional displays debug output for the RTSP Protocol Message Handler.
<b>session</b>	Optional displays debug output for the RTSP client session information.
<b>socket</b>	Optional displays debug output for the RTSP client socket data.

## Defaults

Debug is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Related Commands

Command	Description
<b>debug rtsp api</b>	Displays debug output for the RTSP client API.
<b>debug rtsp client session</b>	Displays debug output for the RTSP client data.
<b>debug rtsp socket</b>	Displays debug output for the RTSP client socket data.

# debug rtsp api

To display information about the Real Time Streaming Protocol API messages passed down to the RTSP client, use the **debug rtsp api** command. To disable the output, use the **no** form of this command.

**debug rtsp api**

**[no] debug rtsp api**

## Syntax Description

This command has no arguments or keywords.

## Defaults

Debug is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Examples

The following example displays the output from entering the **debug rtsp api** command:

```
router# debug rtsp api

RTSP client API debugging is on
router#
Jan  1 00:23:15.775:rtsp_api_create_session:sess_id=0x61A07C78,
      evh=0x60D6B62C context=0x61A07B28
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C2970C
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C2970C
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C2970C
router#
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C29A4C
router#
Jan  1 00:23:22.099:rtsp_api_free_msg_buffer:msg=0x61C29A4C
Jan  1 00:23:22.115:rtsp_api_request:msg=0x61C2A40C
Jan  1 00:23:22.115:rtsp_api_free_msg_buffer:msg=0x61C2A40C
Router#
```

## Related Commands

Command	Description
<b>debug rtsp client session</b>	Displays debug output for the RTSP client data.
<b>debug rtsp pmh</b>	Displays debug messages for the PMH.

<b>Command</b>	<b>Description</b>
<b>debug rtsp socket</b>	Displays debug output for the RTSP client socket data.

# debug rtsp client session

To display debug messages about the Real Time Streaming Protocol client or the current session, use the **debug rtsp [client | session]** command. To disable the output, use the **no** form of this command.

**debug rtsp [client | session]**

**[no] debug rtsp [client | session]**

## Syntax Description

<b>client</b>	(Optional) Displays client information and stream information for the stream that is currently active.
<b>session</b>	(Optional) Displays cumulative information about the session, packet statistics, and general call information such as call ID, session ID, individual RTSP stream URLs, packet statistics, and play duration.

## Defaults

Debug is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

**Examples**

The following example displays the debug messages of the RTSP session:

```

Router# debug rtsp session

RTSP client session debugging is on
router#
Jan  1 00:08:36.099:rtsp_get_new_scb:
Jan  1 00:08:36.099:rtsp_initialize_scb:
Jan  1 00:08:36.099:rtsp_control_process_msg:
Jan  1 00:08:36.099:rtsp_control_process_msg:received MSG request of TYPE 0
Jan  1 00:08:36.099:rtsp_set_event:
Jan  1 00:08:36.099:rtsp_set_event:api_req_msg_type=RTSP_API_REQ_PLAY
Jan  1 00:08:36.103:rtsp_set_event:url:[rtsp://rtsp-cisco.cisco.com:554/en_welcome.au]
Jan  1 00:08:36.103:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:36.103:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_IDLE
                    rtsp_event = RTSP_EV_PLAY_OR_REC
Jan  1 00:08:36.103:act_idle_event_play_or_rec_req:
Jan  1 00:08:36.103:rtsp_resolve_dns:
Jan  1 00:08:36.103:rtsp_resolve_dns:IP Addr = 1.13.79.6:
Jan  1 00:08:36.103:rtsp_connect_to_svr:
Jan  1 00:08:36.103:rtsp_connect_to_svr:socket=0, connection_state = 2
Jan  1 00:08:36.103:rtsp_start_timer:timer (0x62128FD0)starts - delay (10000)
Jan  1 00:08:36.107:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.107:rtsp_stop_timer:timer(0x62128FD0) stops
Jan  1 00:08:36.107:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:36.107:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_IDLE
                    rtsp_event = RTSP_EV_SVR_CONNECTED
Jan  1 00:08:36.107:act_idle_event_svr_connected:
Jan  1 00:08:36.107:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.783:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.783:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:36.783:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_SVR_DESC_OR_ANNOUNCE_RESP
Jan  1 00:08:36.783:act_ready_event_desc_or_announce_resp:
Jan  1
00:08:36.783:act_ready_event_desc_or_announce_resp:RTSP_STATUS_DESC_OR_ANNOUNCE_RESP_OK
Jan  1 00:08:37.287:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:37.287:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:37.287:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_SVR_SETUP_RESP
Jan  1 00:08:37.287:act_ready_event_setup_resp:
Jan  1 00:08:37.287:act_ready_event_setup_resp:Remote RTP Port=13344
Jan  1 00:08:37.287:rtsp_rtp_stream_setup:scb=0x62128F08, callID=0x7 record=0
Jan  1 00:08:37.287:rtsp_rtp_stream_setup:Starting RTCP session.
                    Local IP addr = 1.13.79.45, Remote IP addr = 1.13.79.6,
                    Local RTP port = 18748, Remote RTP port = 13344 CallID=8
Jan  1 00:08:37.291:xmit_func = 0x0 vdbptr = 0x61A0FC98
Jan  1 00:08:37.291:rtsp_control_main:CCAPI Queue Event
Jan  1 00:08:37.291:rtsp_rtp_associate_done:ev=0x62070E08, callID=0x7
Jan  1 00:08:37.291:rtsp_rtp_associate_done:scb=0x62128F08
Jan  1 00:08:37.291:rtsp_rtp_associate_done:callID=0x7, pVdb=0x61F4FBC8,
Jan  1 00:08:37.291:                    spi_context=0x6214145C
Jan  1 00:08:37.291:                    disposition=0, playFunc=0x60CA2238,
Jan  1 00:08:37.291:                    codec=0x5, vad=0, mediaType=6,
Jan  1 00:08:37.291:                    stream_assoc_id=1
Jan  1 00:08:37.291:rtsp_rtp_modify_session:scb=0x62128F08, callID=0x7
Jan  1 00:08:37.291:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:37.291:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_ASSOCIATE_DONE
Jan  1 00:08:37.291:act_ready_event_associate_done:
Jan  1 00:08:37.291:rtsp_get_stream:
Jan  1 00:08:37.783:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:37.783:rtsp_process_async_event:SCB=0x62128F08

```

```

Jan 1 00:08:37.783:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
      rtsp_event = RTSP_EV_SVR_PLAY_OR_REC_RESP
Jan 1 00:08:37.783:act_ready_event_play_or_rec_resp:
Jan 1 00:08:37.783:rtsp_start_timer:timer (0x62128FB0)starts - delay (4249)
rtsp-5#
Jan 1 00:08:42.035:rtsp_process_timer_events:
Jan 1 00:08:42.035:rtsp_process_timer_events:PLAY OR RECORD completed
Jan 1 00:08:42.035:rtsp_process_async_event:SCB=0x62128F08
Jan 1 00:08:42.035:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_PLAY_OR_REC
      rtsp_event = RTSP_EV_PLAY_OR_REC_TIMER_EXPIRED
Jan 1 00:08:42.035:act_play_event_play_done:
Jan 1 00:08:42.035:act_play_event_play_done:elapsed play time = 4249 total play time =
4249
Jan 1 00:08:42.035:rtsp_send_teardown_to_svr:
Jan 1 00:08:42.487:rtsp_control_main:SOCK= 0 Event=0x1
Jan 1 00:08:42.487:rtsp_process_async_event:SCB=0x62128F08
Jan 1 00:08:42.487:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_PLAY_OR_REC
      rtsp_event = RTSP_EV_SVR_TEARDOWN_RESP
Jan 1 00:08:42.487:act_play_event_teardown_resp:
Jan 1 00:08:42.487:rtsp_server_closed:
Jan 1 00:08:42.487:rtsp_send_resp_to_api:
Jan 1 00:08:42.487:rtsp_send_resp_to_api:sending RESP=RTSP_STATUS_PLAY_COMPLETE
Jan 1 00:08:42.491:rtsp_rtp_teardown_stream:scb=0x62128F08, callID=0x7
Jan 1 00:08:42.491:rtsp_rtp_stream_cleanup:scb=0x62128F08, callID=0x7
Jan 1 00:08:42.491:rtsp_update_stream_stats:scb=0x62128F08, stream=0x61A43350,
Jan 1 00:08:42.491:call_info=0x6214C67C, callID=0x7
Jan 1 00:08:42.491:rtsp_update_stream_stats:rx_bytes = 25992
Jan 1 00:08:42.491:rtsp_update_stream_stats:rx_packetes = 82
Jan 1 00:08:42.491:rtsp_reinitialize_scb:
Jan 1 00:08:42.503:rtsp_control_process_msg:
Jan 1 00:08:42.503:rtsp_control_process_msg:received MSG request of TYPE 0
Jan 1 00:08:42.503:rtsp_set_event:
Jan 1 00:08:42.503:rtsp_set_event:api_req_msg_type=RTSP_API_REQ_DESTROY
Jan 1 00:08:42.503:rtsp_session_cleanup:
Jan 1 00:08:42.503:rtsp_create_session_history:scb=0x62128F08, callID=0x7
Jan 1 00:08:42.503:rtsp_insert_session_history_record:current=0x6214BDC8, callID=0x7
Jan 1 00:08:42.503:rtsp_insert_session_history_record:count = 3
Jan 1 00:08:42.503:rtsp_insert_session_history_record:starting history record
deletion_timer of10 minutes
Jan 1 00:08:42.503:rtsp_session_cleanup:deleting session:scb=0x62128F08
Router#

```

**Related Commands**

Command	Description
<b>debug rtsp api</b>	Displays debug output for the RTSP client API.
<b>debug rtsp client session</b>	Displays debug output for the RTSP client data.
<b>debug rtsp pmh</b>	Displays debug messages for the PMH.
<b>debug rtsp socket</b>	Displays debug output for the RTSP client socket data.

# debug rtsp pmh

To display debug information about the Protocol Message Handler (PMH), use the **debug rtsp pmh** command. To disable the output, use the **no** form of this command.

**debug rtsp pmh**

**[no] debug rtsp pmh**

## Syntax Description

This command has no arguments or keywords.

## Defaults

Debug is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Usage Guidelines

Use the **debug rtsp pmh** debug command for the following instances:

- To view packets sent by the gateway (Real time Streaming Protocol client) to the RTSP server. For example:

```
Mar 1 02:25:11.447:SendBuf:DESCRIBE rtsp://rtsp-cisco.cisco.com/en_welcome.au
RTSP/1.0
CSeq:0
```

- To view packets sent by the RTSP server to the gateway. For example:

```
Mar 1 02:25:11.947:#####
Mar 1 02:25:11.947:Mesg_line           :RTSP/1.0 200 OK
Mar 1 02:25:11.951:Content_length      :459
Mar 1 02:25:11.951:Header list
Mar 1 02:25:11.951:Content-length:459
Mar 1 02:25:11.951:Content-type:application/sdp
Mar 1 02:25:11.951:Content-base:rtsp://rtsp-cisco.cisco.com/en_welcome.au/
Mar 1 02:25:11.951:X-TSPort:7802
Mar 1 02:25:11.951>Last-Modified:Thu, 07 Oct 1999 13:51:28 GMT
Mar 1 02:25:11.951>Date:Mon, 10 Jan 2000 16:40:59 GMT
Mar 1 02:25:11.951:CSeq:0
```

**Examples**

The following example output displays the result from entering the **debug rtsp pmh** command:

```
Router# debug rtsp pmh

RTSP client Protocol Message Handler debugging is on
Router#
Jan  1 00:22:34.087:rtsp_pmh_update_play_req_url:
Jan  1 00:22:34.087:rtsp_pmh_parse_url:
Jan  1 00:22:34.087:Input-Url:rtsp://rtsp-cisco.cisco.com:554/en_welcome.au
Jan  1 00:22:34.087:Hostname:rtsp-cisco.cisco.com
Jan  1 00:22:34.087:Port      :554
Jan  1 00:22:34.087:Path      :en_welcome.au
Jan  1 00:22:34.091:rtsp_pmh_build_desc_req:
Jan  1 00:22:34.091:rtsp_pmh_add_req_line:
Jan  1 00:22:34.091:RequestLine:(DESCRIBE rtsp://rtsp-cisco.cisco.com:554/en_welcome.au
RTSP/1.0
)
Jan  1 00:22:34.091:SendBuf:DESCRIBE rtsp://rtsp-cisco.cisco.com:554/en_welcome.au
RTSP/1.0
CSeq:0

Jan  1 00:22:34.091:last_req = 0
Jan  1 00:22:34.739:rtsp_pmh_parse_svr_response:
Jan  1 00:22:34.739:rtsp_pmh_create_mesg:
Jan  1 00:22:34.739:#####
Jan  1 00:22:34.739:Mesg_line      :RTSP/1.0 200 OK
Jan  1 00:22:34.739:Content_length  :482
Jan  1 00:22:34.739:Header list
Jan  1 00:22:34.739:Content-length:482
Jan  1 00:22:34.739:Content-type:application/sdp
Jan  1 00:22:34.739:Content-base:rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/
Jan  1 00:22:34.739>Last-Modified:Thu, 07 Oct 1999 13:51:28 GMT
Jan  1 00:22:34.739:X-TSPort:7802
Jan  1
00:22:34.739:vsrc:http://rtsp-cisco.cisco.com:8080/viewsource/template.html?nuyhtgywkgz6mc
9AbhC4gn5gBsqp4eAlv1yeC3d4ngEt5o5gwuw4t6x05jbhcv66ngE8xg8f
Jan  1
00:22:34.739:Set-Cookie:cbid=ekeghhiljgekgihheoqohpptrrrjrktlufkegkioihgjfdlplrnqogpqlrpsk
qnuffgjcmcl;path=/;expires=Thu,31-Dec-2037 23:59:59 GMT
Jan  1 00:22:34.739>Date:Mon, 10 Apr 2000 15:39:17 GMT
Jan  1 00:22:34.739:CSeq:0
Jan  1 00:22:34.739:Message Body
Jan  1 00:22:34.739:v=0
o=- 939300688 939300688 IN IP4 1.13.79.6
s=<No title>
i=<No author> <No copyright>
a=StreamCount:integer;1
t=0 0
m=audio 0 RTP/AVP 0
a=control:streamid=0
a=rtpmap:0 L8/8000/1
a=length:npt=3.249000
a=range:npt=0-3.249000
a=mimetype:string;"audio/x-pn-au"
a=StartTime:integer;0
a=AvgBitRate:integer;64000
a=AvgPacketSize:integer;320
a=Preroll:integer;0
a=MaxPacketSize:integer;320
a=MaxBitRate:integer;64000
a=OpaqueData:buffer;"AQABAEfAAA="
a=StreamName:string;"audio/x-pn-au"
```

```

Jan 1 00:22:34.739:#####
Jan 1 00:22:34.739:rtsp_pmh_process_resp_headers:
Jan 1 00:22:34.739:rtsp_pmh_get_header_value:
Jan 1 00:22:34.739:rtsp_pmh_process_resp_headers:Cseq=1
Jan 1 00:22:34.739:rtsp_pmh_get_resp_line:
Jan 1 00:22:34.739:rtsp_pmh_process_resp_headers:Response Status
Jan 1 00:22:34.739:rtsp_pmh_process_resp_headers:Status Code:200
Jan 1 00:22:34.739:rtsp_pmh_process_resp_headers:Reason Phrase:OK
Jan 1 00:22:34.743:rtsp_pmh_parse_mesg_body:
Jan 1 00:22:34.743:rtsp_pmh_process_resp_headers:Response
URL:rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0
Jan 1 00:22:34.743:rtsp_pmh_process_resp_headers:RealServer Duration
Jan 1 00:22:34.743:rtsp_pmh_process_resp_headers:IP/TV Duration
Jan 1 00:22:34.743:rtsp_pmh_get_range_from_npt:
Jan 1 00:22:34.743:rtsp_pmh_get_range_from_npt:Duration:3249 msec
Jan 1 00:22:34.743:rtsp_pmh_update_resp_status:
Jan 1 00:22:34.743:rtsp_pmh_update_resp_status:Control Not active
Jan 1 00:22:34.743:#####
Jan 1 00:22:34.743:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:34.743:Content_length      :482
Jan 1 00:22:34.743:Header list
Jan 1 00:22:34.743:Content-length:482
Jan 1 00:22:34.743:Content-type:application/sdp
Jan 1 00:22:34.743:Content-base:rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/
Jan 1 00:22:34.743>Last-Modified:Thu, 07 Oct 1999 13:51:28 GMT
Jan 1 00:22:34.743:X-TSPort:7802
Jan 1
00:22:34.743:vsrc:http://rtsp-cisco.cisco.com:8080/viewsource/template.html?nuyhtgywkgz6mc
9AbhC4gn5gBsqp4eA1vlveC3d4ngEt5o5gwuw4t6x05jbhcv66ngE8xg8f
Jan 1
00:22:34.743:Set-Cookie:cbid=ekeghhiljgekigihheoqhpptrrjrktlufkegkioihgjfdlplrnqogpoqlrpsk
qnuffgjcml;path=/;expires=Thu,31-Dec-2037 23:59:59 GMT
Jan 1 00:22:34.743>Date:Mon, 10 Apr 2000 15:39:17 GMT
Jan 1 00:22:34.743:CSeq:0
Jan 1 00:22:34.743:Message Body
Jan 1 00:22:34.743:v=0
o=- 939300688 939300688 IN IP4 1.13.79.6
s=<No title>
i=<No author> <No copyright>
a=StreamCount:integer;1
t=0 0
m=audio 0 RTP/AVP 0
a=control:streamid=0
a=rtpmap:0 L8/8000/1
a=length:npt=3.249000
a=range:npt=0-3.249000
a=mimetype:string;"audio/x-pn-au"
a=StartTime:integer;0
a=AvgBitRate:integer;64000
a=AvgPacketSize:integer;320
a=Preroll:integer;0
a=MaxPacketSize:integer;320
a=MaxBitRate:integer;64000
a=OpaqueData:buffer;"AQABAEfAAA="
a=StreamName:string;"audio/x-pn-au"

Jan 1 00:22:34.743:#####
Jan 1 00:22:34.743:rtsp_pmh_free_mesg:
Jan 1 00:22:34.743:rtsp_pmh_build_setup_req:
Jan 1 00:22:34.743:rtsp_pmh_add_req_line:
Jan 1 00:22:34.743:RequestLine:(SETUP
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
)
Jan 1 00:22:34.747:rtsp_pmh_build_setup_req:SendBuf:SETUP

```

```
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
CSeq:1
Transport:rtp/avp;unicast;client_port=18084

Jan 1 00:22:35.243:rtsp_pmh_parse_svr_response:
Jan 1 00:22:35.243:rtsp_pmh_create_mesg:
Jan 1 00:22:35.243:#####
Jan 1 00:22:35.243:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:35.243:Content_length       :0
Jan 1 00:22:35.243:Header list
Jan 1
00:22:35.243:Transport:rtp/avp;unicast;client_port=18084-18085;server_port=23192-23193
Jan 1 00:22:35.243:Session:24457-1
Jan 1 00:22:35.243>Date:Mon, 10 Apr 2000 15:39:17 GMT
Jan 1 00:22:35.243:CSeq:1
Jan 1 00:22:35.243:Message Body
Jan 1 00:22:35.243:#####
Jan 1 00:22:35.243:rtsp_pmh_process_resp_headers:
Jan 1 00:22:35.243:rtsp_pmh_get_header_value:
Jan 1 00:22:35.243:rtsp_pmh_process_resp_headers:Cseq=2
Jan 1 00:22:35.243:rtsp_pmh_get_resp_line:
Jan 1 00:22:35.243:rtsp_pmh_process_resp_headers:Response Status
Jan 1 00:22:35.243:rtsp_pmh_process_resp_headers:Status Code:200
Jan 1 00:22:35.243:rtsp_pmh_process_resp_headers:Reason Phrase:OK
Jan 1 00:22:35.243:rtsp_pmh_get_header_value:
Jan 1 00:22:35.247:rtsp_pmh_get_header_value:
Jan 1 00:22:35.247:rtsp_pmh_process_resp_headers:RTP PORT= 23192
Jan 1 00:22:35.247:rtsp_pmh_process_resp_headers:RTP PORT= 23192
Jan 1 00:22:35.247:rtsp_pmh_update_resp_status:
Jan 1 00:22:35.247:rtsp_pmh_update_resp_status:Control Not active
Jan 1 00:22:35.247:#####
Jan 1 00:22:35.247:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:35.247:Content_length       :0
Jan 1 00:22:35.247:Header list
Jan 1
00:22:35.247:Transport:rtp/avp;unicast;client_port=18084-18085;server_port=23192-23193
Jan 1 00:22:35.247:Session:24457-1
Jan 1 00:22:35.247>Date:Mon, 10 Apr 2000 15:39:17 GMT
Jan 1 00:22:35.247:CSeq:1
Jan 1 00:22:35.247:Message Body
Jan 1 00:22:35.247:#####
Jan 1 00:22:35.247:rtsp_pmh_free_mesg:
Jan 1 00:22:35.247:rtsp_pmh_build_play_req:
Jan 1 00:22:35.247:rtsp_pmh_add_req_line:
Jan 1 00:22:35.247:RequestLine:(PLAY
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
)
Jan 1 00:22:35.247:rtsp_pmh_build_play_req:SendBuf:PLAY
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24457-1
CSeq:2

Jan 1 00:22:35.735:rtsp_pmh_parse_svr_response:
Jan 1 00:22:35.735:rtsp_pmh_create_mesg:
Jan 1 00:22:35.739:#####
Jan 1 00:22:35.739:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:35.739:Content_length       :0
Jan 1 00:22:35.739:Header list
Jan 1 00:22:35.739>Date:Mon, 10 Apr 2000 15:39:18 GMT
Jan 1 00:22:35.739:CSeq:2
Jan 1 00:22:35.739:Message Body
Jan 1 00:22:35.739:#####
```

debug rtsp pmh

```

Jan 1 00:22:35.739:rtsp_pmh_process_resp_headers:
Jan 1 00:22:35.739:rtsp_pmh_get_header_value:
Jan 1 00:22:35.739:rtsp_pmh_process_resp_headers:Cseq=3
Jan 1 00:22:35.739:rtsp_pmh_get_resp_line:
Jan 1 00:22:35.739:rtsp_pmh_process_resp_headers:Response Status
Jan 1 00:22:35.739:rtsp_pmh_process_resp_headers:Status Code:200
Jan 1 00:22:35.739:rtsp_pmh_process_resp_headers:Reason Phrase:OK
Jan 1 00:22:35.739:rtsp_pmh_update_resp_status:
Jan 1 00:22:35.739:rtsp_pmh_update_resp_status:Control Not active
Jan 1 00:22:35.739:#####
Jan 1 00:22:35.739:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:35.739:Content_length       :0
Jan 1 00:22:35.739:Header list
Jan 1 00:22:35.739>Date:Mon, 10 Apr 2000 15:39:18 GMT
Jan 1 00:22:35.739:CSeq:2
Jan 1 00:22:35.739:Message Body
Jan 1 00:22:35.739:#####
Jan 1 00:22:35.739:rtsp_pmh_free_mesg:
Jan 1 00:22:40.011:rtsp_pmh_build_teardown_req:
Jan 1 00:22:40.011:rtsp_pmh_add_req_line:
Jan 1 00:22:40.011:RequestLine:(TEARDOWN
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
)
Jan 1 00:22:40.011:SendBuf:TEARDOWN
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24457-1
CSeq:3

Jan 1 00:22:40.443:rtsp_pmh_parse_svr_response:
Jan 1 00:22:40.443:rtsp_pmh_create_mesg:
Jan 1 00:22:40.443:#####
Jan 1 00:22:40.443:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:40.443:Content_length       :0
Jan 1 00:22:40.443:Header list
Jan 1 00:22:40.443>Date:Mon, 10 Apr 2000 15:39:23 GMT
Jan 1 00:22:40.443:CSeq:3
Jan 1 00:22:40.443:Message Body
Jan 1 00:22:40.443:#####
Jan 1 00:22:40.443:rtsp_pmh_process_resp_headers:
Jan 1 00:22:40.443:rtsp_pmh_get_header_value:
Jan 1 00:22:40.443:rtsp_pmh_process_resp_headers:Cseq=4
Jan 1 00:22:40.443:rtsp_pmh_get_resp_line:
Jan 1 00:22:40.443:rtsp_pmh_process_resp_headers:Response Status
Jan 1 00:22:40.443:rtsp_pmh_process_resp_headers:Status Code:200
Jan 1 00:22:40.443:rtsp_pmh_process_resp_headers:Reason Phrase:OK
Jan 1 00:22:40.443:rtsp_pmh_update_resp_status:
Jan 1 00:22:40.443:rtsp_pmh_update_resp_status:Control Not active
Jan 1 00:22:40.443:#####
Jan 1 00:22:40.447:Mesg_line           :RTSP/1.0 200 OK
Jan 1 00:22:40.447:Content_length       :0
Jan 1 00:22:40.447:Header list
Jan 1 00:22:40.447>Date:Mon, 10 Apr 2000 15:39:23 GMT
Jan 1 00:22:40.447:CSeq:3
Jan 1 00:22:40.447:Message Body
Jan 1 00:22:40.447:#####
Jan 1 00:22:40.447:rtsp_pmh_free_mesg:
Router#

Jan 1 00:14:20.483:rtsp_tcp_socket_connect:
Jan 1 00:14:20.483:rtsp_tcp_socket_connect:Socket = 0
Jan 1 00:14:20.483:           Dest_addr = 1.13.79.6  Dest_Port=554
Jan 1 00:14:20.487:rtsp_send_req_to_svr:Socket = 0 send_buf = DESCRIBE

```

```
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au RTSP/1.0
CSeq:0

len = 76
Jan 1 00:14:20.491:rtsp_send_req_to_svr:bytes_sent = 76

Jan 1 00:14:20.491:rtsp_read_svr_resp:Socket = 0
Jan 1 00:14:20.491:rtsp_read_svr_resp:NBYTES = -1
Jan 1 00:14:21.155:rtsp_read_svr_resp:Socket = 0
Jan 1 00:14:21.159:rtsp_read_svr_resp:NBYTES = 996
Jan 1 00:14:21.223:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan 1 00:14:21.227:rtsp_read_svr_resp:RESP received OK
Jan 1 00:14:21.227:rtsp_send_req_to_svr:Socket = 0 send_buf = SETUP
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
CSeq:1
Transport:rtp/avp;unicast;client_port=18074

len = 130
Jan 1 00:14:21.227:rtsp_send_req_to_svr:bytes_sent = 130

Jan 1 00:14:21.663:rtsp_read_svr_resp:Socket = 0
Jan 1 00:14:21.663:rtsp_read_svr_resp:NBYTES = 159
Jan 1 00:14:21.663:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:1
Date:Mon, 10 Apr 2000 15:31:04 GMT
Session:24455-1
Transport:rtp/avp;unicast;client_port=18074-18075;server_port=15562-15563

Jan 1 00:14:21.663:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan 1 00:14:21.663:rtsp_read_svr_resp:RESP received OK
Jan 1 00:14:21.663:rtsp_send_req_to_svr:Socket = 0 send_buf = PLAY
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24455-1
CSeq:2

len = 101
Jan 1 00:14:21.667:rtsp_send_req_to_svr:bytes_sent = 101

Jan 1 00:14:22.155:rtsp_read_svr_resp:Socket = 0
Jan 1 00:14:22.155:rtsp_read_svr_resp:NBYTES = 65
Jan 1 00:14:22.155:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:2
Date:Mon, 10 Apr 2000 15:31:04 GMT

Jan 1 00:14:22.155:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan 1 00:14:22.155:rtsp_read_svr_resp:RESP received OK
rtsp-5#
Jan 1 00:14:26.411:rtsp_send_req_to_svr:Socket = 0 send_buf = TEARDOWN
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24455-1
CSeq:3

len = 105
Jan 1 00:14:26.411:rtsp_send_req_to_svr:bytes_sent = 105

Jan 1 00:14:26.863:rtsp_read_svr_resp:Socket = 0
Jan 1 00:14:26.863:rtsp_read_svr_resp:NBYTES = 65
Jan 1 00:14:26.863:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:3
Date:Mon, 10 Apr 2000 15:31:09 GMT
```

**debug rtsp pmh**

```
Jan 1 00:14:26.863:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan 1 00:14:26.863:rtsp_read_svr_resp:RESP received OK
Jan 1 00:14:26.863:rtsp_close_svr_connection:closing socket 0
Router#
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>debug rtsp api</b>	Displays debug output for the RTSP client API.
<b>debug rtsp client session</b>	Displays debug output for the RTSP client data.
<b>debug rtsp socket</b>	Displays debug output for the RTSP client socket data.

# debug rtsp socket

To display debug messages about the packets received or sent on the TCP or User Datagram Protocol sockets, use the **debug rtsp socket** command. To disable the output, use the **no** form of this command.

**debug rtsp socket**

**[no] debug rtsp socket**

---

## Syntax Description

This command has no arguments or keywords.

---

## Defaults

Debug is not enabled.

---

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

---

## Usage Guidelines

Each RTSP session has a TCP port for control and UDP (RTP) port for delivery of data. The control connection (TCP socket) is used to exchange a set of messages (request from the RTSP client and the response from the server) for playing a prompt. The **debug rtsp socket** command enables the user to debug the message exchanges being done on the TCP control connection.

---

**Examples**

The following example displays output from the **debug rtsp socket** command:

```
Router# show debug rtsp socket

Jan  1 00:14:20.483:rtsp_tcp_socket_connect:
Jan  1 00:14:20.483:rtsp_tcp_socket_connect:Socket = 0
Jan  1 00:14:20.483:          Dest_addr = 1.13.79.6  Dest_Port=554
Jan  1 00:14:20.487:rtsp_send_req_to_svr:Socket = 0 send_buf = DESCRIBE
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au RTSP/1.0
CSeq:0

len = 76
Jan  1 00:14:20.491:rtsp_send_req_to_svr:bytes_sent = 76

Jan  1 00:14:20.491:rtsp_read_svr_resp:Socket = 0
Jan  1 00:14:20.491:rtsp_read_svr_resp:NBYTES = -1
Jan  1 00:14:21.155:rtsp_read_svr_resp:Socket = 0
Jan  1 00:14:21.159:rtsp_read_svr_resp:NBYTES = 996
Jan  1 00:14:21.223:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan  1 00:14:21.227:rtsp_read_svr_resp:RESP received OK
Jan  1 00:14:21.227:rtsp_send_req_to_svr:Socket = 0 send_buf = SETUP
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
CSeq:1
Transport:rtp/avp;unicast;client_port=18074

len = 130
Jan  1 00:14:21.227:rtsp_send_req_to_svr:bytes_sent = 130
```

```

Jan  1 00:14:21.663:rtsp_read_svr_resp:Socket = 0
Jan  1 00:14:21.663:rtsp_read_svr_resp:NBYTES = 159
Jan  1 00:14:21.663:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:1
Date:Mon, 10 Apr 2000 15:31:04 GMT
Session:24455-1
Transport:rtp/avp;unicast;client_port=18074-18075;server_port=15562-15563

Jan  1 00:14:21.663:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan  1 00:14:21.663:rtsp_read_svr_resp:RESP received OK
Jan  1 00:14:21.663:rtsp_send_req_to_svr:Socket = 0 send_buf = PLAY
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24455-1
CSeq:2

len = 101
Jan  1 00:14:21.667:rtsp_send_req_to_svr:bytes_sent = 101

Jan  1 00:14:22.155:rtsp_read_svr_resp:Socket = 0
Jan  1 00:14:22.155:rtsp_read_svr_resp:NBYTES = 65
Jan  1 00:14:22.155:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:2
Date:Mon, 10 Apr 2000 15:31:04 GMT

Jan  1 00:14:22.155:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan  1 00:14:22.155:rtsp_read_svr_resp:RESP received OK
rtsp-5#
Jan  1 00:14:26.411:rtsp_send_req_to_svr:Socket = 0 send_buf = TEARDOWN
rtsp://rtsp-cisco.cisco.com:554/en_welcome.au/streamid=0 RTSP/1.0
Session:24455-1
CSeq:3

len = 105
Jan  1 00:14:26.411:rtsp_send_req_to_svr:bytes_sent = 105

Jan  1 00:14:26.863:rtsp_read_svr_resp:Socket = 0
Jan  1 00:14:26.863:rtsp_read_svr_resp:NBYTES = 65
Jan  1 00:14:26.863:rtsp_read_svr_resp:rcv_buf = RTSP/1.0 200 OK
CSeq:3
Date:Mon, 10 Apr 2000 15:31:09 GMT

Jan  1 00:14:26.863:rtsp_read_svr_resp:rtsp_pmh_parse_svr_response complete
Jan  1 00:14:26.863:rtsp_read_svr_resp:RESP received OK
Jan  1 00:14:26.863:rtsp_close_svr_connection:closing socket 0
Router#

```

**Related Commands**

Command	Description
<b>debug rtsp api</b>	Displays debug output for the RTSP client API.
<b>debug rtsp client session</b>	Displays debug output for the RTSP client data.
<b>debug rtsp pmh</b>	Displays debug messages for the PMH.

# debug voip ivr

To display debug messages for Voice over IP IVR interactions, use the **debug voip ivr** command. To disable the debug output, use the **no** form of this command.

**debug voip ivr**

[no] **debug voip ivr** *type*

## Syntax Description

<b>all</b>	Displays all IVR messages.
<b>applib</b>	Displays IVR API libraries being processed.
<b>callsetup</b>	Displays IVR call setup being processed.
<b>digitcollect</b>	Displays IVR digits collected during the call.
<b>dynamic</b>	Displays IVR dynamic prompt play debug.
<b>error</b>	Displays IVR errors.
<b>script</b>	Displays IVR script debug.
<b>settlement</b>	Displays IVR settlement activities.
<b>states</b>	Displays IVR states.
<b>tclcommands</b>	Displays the TCL commands used in the script.

## Defaults

Debug is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.1(5)XM2	Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB	This command was implemented on the Cisco AS5850 platform.

## Examples

The following examples are from the code for Cisco IOS Release 12.1(3)T. The output is displayed when the **debug voip ivr** *type* command is entered:

The following output is displayed when the **debug voip ivr applib** command is entered:

```
Router1# debug voip ivr applib

ivr:
  ivr app library debugging is on
Router1#
Jan 10 17:42:04.180:AppManagerCCAPI_Interface:
Jan 10 17:42:04.180:AppNewLeg
Jan 10 17:42:04.180:AppPushLegORConnection:Pushing LEG[34      ][NULL
] Onto {HAN[TCL_HAND][NULL      ]      ( )}
Jan 10 17:42:04.180:Event CC_EV_CALL_SETUP_IND[29]:LEG[34
][TCL_HAND]
Jan 10 17:42:04.184:AppPushHandler:Pushing {HAN[DC_HAND ][NULL      ]
( )} Onto {HAN[TCL_HAND][NULL      ]      ( LEG[34      ][TCL_HAND] )}
Jan 10 17:42:04.184:AppPushLegORConnection:Pushing LEG[34
][TCL_HAND] Onto {HAN[DC_HAND ][TCL_HAND]      ( )}
Jan 10 17:42:04.184:$ mediaPlay():CallID 34
Jan 10 17:42:04.184:Event CC_EV_CALL_REPORT_DIGITS_DONE[45]:LEG[34
][DC_HAND ]
Jan 10 17:42:17.261:AppMediaCallback:CallID 34 received
                response 'MSW_RESPONSE_TYPE_PLAY'
                with reason 'MSW_REASON_GENERIC_SUCCESS'
Jan 10 17:42:17.261:Event APP_EV_MEDIA_CALLBACK[47]:LEG[34
][DC_HAND ]
Jan 10 17:42:18.209:%ISDN-6-DISCONNECT:Interface Serial0:0
disconnected from unknown , call lasted 13 seconds
```

The following output is displayed when the **debug voip ivr callsetup** command is entered:

```
Router# debug voip ivr callsetup

Jan 10 17:45:57.528:%SYS-5-CONFIG_I:Configured from console by lab on
console
Jan 10 17:46:37.682:InitiateCallSetup:Incoming[66] AlertTime -1
Destinations(1) [ 3450070 ]
Jan 10 17:46:37.682:DNInitiate:Destination[3450070]
Jan 10 17:46:37.682:DNSetupPeer:
Jan 10 17:46:37.682:Destination SetupPeer cid(66), destPat(3450070),
match(2), prefix(), peer(61CB5CAC)
Jan 10 17:46:37.762:DNHandler:
(DN_SETTING[1])--(CC_EV_CALL_ALERT[11])--IGNORED-->>(DN_SETTING[1])
Jan 10 17:46:37.762:CS_Setting_ALERT:
Jan 10 17:46:37.762:CSPopLegAndWait:
Jan 10 17:46:37.762:CallSetupHandler:
(CS_SETTING[0]) ----(CS_EV_ALERT[0])----->>>(CS_CONFINGALERT[4])
Jan 10 17:46:37.762:CS_ConfingAlert_CREATEDONE:
Jan 10 17:46:37.762:CallSetupHandler:
(CS_CONFINGALERT[4])
----(CS_EV_CREATEDONE[4])----->>>(CS_CONFEDALERT[5])
Jan 10 17:46:37.762:CallSetupHandler:
(CS_CONFEDALERT[5])--(DN_SETTING[APP_EV_NULL])--IGNORED-->>>(CS_CONFEDALERT[5])

Router1#
Jan 10 17:46:47.682:CallSetupHandler:
(CS_CONFEDALERT[5])--(DN_SETTING[APP_EV_NULL])--IGNORED-->>>(CS_CONFEDALERT[5])

Jan 10 17:46:48.642:CS_ConfedAlert_CONNECTED:
Jan 10 17:46:48.642:CSDiscReturnAndEmptyLegALL:
Jan 10 17:46:48.642:DNCleanup:
Jan 10 17:46:48.642:DNSettlementCleanup:cid(66) trans=0, provider=0
Jan 10 17:46:48.642:CSReturnIFDone:CallSetup Returning(Status
CS_ACTIVE)
Jan 10 17:46:48.642:CallSetupHandler:
(CS_CONFEDALERT[5]) ----(CS_EV_CONNECTED[1])----->>>(CS_CONFED[3])
Jan 10 17:46:48.646:CallSetupCleanup:
Router #
```

The following output is displayed when the **debug voip ivr digitcollect** command is entered:

```
Router# debug voip ivr digitcollect

ivr:
  ivr digit collect debugging is on
Router1#
Router1#
Router1#
Jan 10 17:47:55.558:DigitCollect:DialPlan=FALSE AbortKey=* TermKey=#
NumPatts=1
      Enable=FALSE InterruptPrompt=TRUE maxDigits=11
Jan 10 17:47:55.558:act_DCRunning_RDone:callid=68 Enable succeeded.
Router1#
Jan 10 17:48:04.006:DCHandlerFunc:PassingThrough
Jan 10 17:48:04.066:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:04.066:act_DCRunning_RDone:callid=68 Reporting disabled.
Jan 10 17:48:04.066:DigitCollectComplete:Status 5=DC_MATCHED_PATTERN.
Digits=1
Jan 10 17:48:04.070:DigitCollect:DialPlan=FALSE AbortKey=* TermKey=#
NumPatts=0
      Enable=FALSE InterruptPrompt=TRUE maxDigits=11
Jan 10 17:48:04.070:DCHandlerCleanup:
Jan 10 17:48:04.074:act_DCRunning_RDone:callid=68 Enable succeeded.
Router1#
Router1#
Jan 10 17:48:08.038:DCHandlerFunc:PassingThrough
Jan 10 17:48:09.246:DCHandlerFunc:PassingThrough
Jan 10 17:48:09.286:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:09.478:DCHandlerFunc:PassingThrough
Jan 10 17:48:09.506:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:10.739:DCHandlerFunc:PassingThrough
Jan 10 17:48:10.779:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:11.027:DCHandlerFunc:PassingThrough
Jan 10 17:48:11.067:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:11.687:DCHandlerFunc:PassingThrough
Jan 10 17:48:11.747:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:12.219:DCHandlerFunc:PassingThrough
Jan 10 17:48:12.279:act_DCRunning_Digit::pLeg 68 Digit 2
Jan 10 17:48:14.227:DCHandlerFunc:PassingThrough
Jan 10 17:48:14.287:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:14.779:DCHandlerFunc:PassingThrough
Jan 10 17:48:14.859:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:15.307:DCHandlerFunc:PassingThrough
Jan 10 17:48:15.359:act_DCRunning_Digit::pLeg 68 Digit 1
Jan 10 17:48:15.719:DCHandlerFunc:PassingThrough
Jan 10 17:48:15.759:act_DCRunning_Digit::pLeg 68 Digit 2
Jan 10 17:48:16.219:DCHandlerFunc:PassingThrough
Jan 10 17:48:16.299:act_DCRunning_Digit::pLeg 68 Digit T
Jan 10 17:48:16.299:act_DCRunning_RDone:callid=68 Reporting disabled.
Jan 10 17:48:16.299:DigitCollectComplete:Status 5=DC_MATCHED_PATTERN.
Digits=1111121112
Jan 10 17:48:16.303:DCHandlerCleanup:
Jan 10 17:48:16.335:DigitCollect:DialPlan=TRUE AbortKey=* TermKey=#
NumPatts=0
      Enable=FALSE InterruptPrompt=TRUE maxDigits=0
Jan 10 17:48:16.339:act_DCRunning_RDone:callid=68 Enable succeeded.
Router #
```

The following output is displayed when the **debug voip ivr script** command is entered:

```
Router# deb voip ivr script

ivr:
  ivr script debugging is on
Router1#
Jan 10 17:49:10.250:FSM Transtion:([1
]CALL_INIT,[29]ev_setup_indication)---([10]act_Setup)--->([4
]LANGSELECTION)
Jan 10 17:49:10.250:TotalLanguages= 2
Router1#
Router1#
Jan 10 17:49:16.662:FSM Transtion:([4
]LANGSELECTION,[55]ev_digitcollect_done)---([1 ]act_LangSelect)--->([5
]CARDSELECTION)
Router1#
Router1#
Jan 10 17:49:20.630:([5 ]CARDSELECT,[47]ev_media_d) -----> NOTHANDLED
Jan 10 17:49:26.770:FSM Transtion:([5
]CARDSELECTION,[55]ev_digitcollect_done)---([2
]act_GotCardNumber)--->([6 ]AUTHORIZE)
Jan 10 17:49:26.806:FSM Transtion:([6
]AUTHORIZE,[49]ev_authorize_done)---([8 ]act_FirstAuthorized)--->([7
]GETDEST)
Jan 10 17:49:26.806: aaa authorize Status=ao_000
Router1#
Router1#
Router1#
Jan 10 17:49:33.395:([7 ]GETDEST , [47]ev_media_d) -----> NOTHANDLED
Jan 10 17:49:36.411:FSM Transtion:([7
]GETDEST,[55]ev_digitcollect_done)---([3 ]act_GotDest)--->([8
]SECONDAUTHORIZE)
Jan 10 17:49:36.451:FSM Transtion:([8
]SECONDAUTHORIZE,[49]ev_authorize_done)---([5
]act_SecondAuthorized)--->([10]PLACECALL)
Jan 10 17:49:36.451: aaa authorize Status=ao_000
Jan 10 17:49:42.179:FSM Transtion:
([10]PLACECALL,[47]ev_media_done)---([9
]act_CallSetup)--->([10]PLACECALL)
```

The following output is displayed when the **debug voip ivr tclcommands** command is entered:

```
Router# debug voip ivr tclcommands

ivr tcl commands debugging is on
Router1#
Jan 10 17:50:29.106:tcl_infotagCmd:infotag get leg_ani
Jan 10 17:50:29.106:tcl_getInfoCmd:get leg_ani
Jan 10 17:50:29.106:vtr_ci_incani:argc 2 argindex 2
Jan 10 17:50:29.106:tcl_infotagCmd:infotag set med_language 1
Jan 10 17:50:29.106:tcl_setInfoCmd:set med_language 1
Jan 10 17:50:29.106:vtw_ms_language:
Jan 10 17:50:29.106:tcl_legCmd:leg setupack leg_incoming
Jan 10 17:50:29.106:tcl_setupAckCmd:setupack leg_incoming
Jan 10 17:50:29.106:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:29.106:tcl_legCmd:leg proceeding leg_incoming
Jan 10 17:50:29.106:tcl_callProceedingCmd:proceeding leg_incoming
Jan 10 17:50:29.106:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:29.110:tcl_legCmd:leg connect leg_incoming
Jan 10 17:50:29.110:tcl_callConnectCmd:connect leg_incoming
Jan 10 17:50:29.110:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:29.110:tcl_legCmd:leg collectdigits leg_incoming param1
patterns
Jan 10 17:50:29.110:tcl_collectDigitsCmd:collectdigits leg_incoming
param1 patterns
Jan 10 17:50:29.110:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:29.110:tcl_mediaCmd:media play leg_incoming _welcome.au
%s1000 %c1 _lang_sel1.au %s1000 %c2 _lang_sel2.au
Jan 10 17:50:29.110:tcl_mediaPlayCmd:play leg_incoming _welcome.au
%s1000 %c1 _lang_sel1.au %s1000 %c2 _lang_sel2.au
Jan 10 17:50:29.110:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Router1#
Router1#
Jan 10 17:50:35.506:tcl_infotagCmd:infotag get evt_status
Jan 10 17:50:35.506:tcl_getInfoCmd:get evt_status
Jan 10 17:50:35.506:vtr_ev_status:
Jan 10 17:50:35.510:tcl_infotagCmd:infotag get evt_dcdigits
Jan 10 17:50:35.510:tcl_getInfoCmd:get evt_dcdigits
Jan 10 17:50:35.510:vtr_ev_dcdigits:
Jan 10 17:50:35.510:DCDIGITS [1]
Jan 10 17:50:35.510:tcl_infotagCmd:infotag set med_language 1
Jan 10 17:50:35.510:tcl_setInfoCmd:set med_language 1
Jan 10 17:50:35.510:vtw_ms_language:
Jan 10 17:50:35.510:tcl_legCmd:leg collectdigits leg_incoming param1
Jan 10 17:50:35.510:tcl_collectDigitsCmd:collectdigits leg_incoming

param1
Jan 10 17:50:35.510:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:35.510:tcl_mediaCmd:media play leg_incoming
_enter_card_num.au
Jan 10 17:50:35.510:tcl_mediaPlayCmd:play leg_incoming
_enter_card_num.au
Jan 10 17:50:35.514:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Router1#
Jan 10 17:50:43.878:tcl_infotagCmd:infotag get evt_status
Jan 10 17:50:43.878:tcl_getInfoCmd:get evt_status
```

## debug voip ivr

```

Jan 10 17:50:43.878:vtr_ev_status:
Jan 10 17:50:43.882:tcl_infotagCmd:infotag get evt_dcdigits
Jan 10 17:50:43.882:tcl_getInfoCmd:get evt_dcdigits
Jan 10 17:50:43.882:vtr_ev_dcdigits:
Jan 10 17:50:43.882:DCDIGITS [1111121112]
Jan 10 17:50:43.882:tcl_aaaCmd:aaa authorize 111112 1112 50073
leg_incoming
Jan 10 17:50:43.882:tcl_AuthorizeCmd:authorize 111112 1112 50073
leg_incoming
Jan 10 17:50:43.882:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:43.882:Authorize
Jan 10 17:50:43.882:    account=111112
Jan 10 17:50:43.882:    password=1112
Jan 10 17:50:43.882:    ani      =50073
Jan 10 17:50:43.882:    dnis    =
Jan 10 17:50:43.910:tcl_infotagCmd:infotag get evt_status
Jan 10 17:50:43.910:tcl_getInfoCmd:get evt_status
Jan 10 17:50:43.910:vtr_ev_status:
Jan 10 17:50:43.914:tcl_infotagCmd:infotag get aaa_avpair_exists
creditAmount
Jan 10 17:50:43.914:tcl_getInfoCmd:get aaa_avpair_exists creditAmount
Jan 10 17:50:43.914:vtr_ra_avpair_exists:
Jan 10 17:50:43.914:tcl_infotagCmd:infotag get aaa_avpair creditAmount

Jan 10 17:50:43.914:tcl_getInfoCmd:get aaa_avpair creditAmount
Jan 10 17:50:43.914:vtr_ra_avpair:
Jan 10 17:50:43.914:tcl_legCmd:leg collectdigits leg_incoming param2
Jan 10 17:50:43.914:tcl_collectDigitsCmd:collectdigits leg_incoming
param2
Jan 10 17:50:43.914:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1
Jan 10 17:50:43.914:tcl_mediaCmd:media play leg_incoming _you_have.au
%a1000 %s1000 _enter_dest.au
Jan 10 17:50:43.914:tcl_mediaPlayCmd:play leg_incoming _you_have.au
%a1000 %s1000 _enter_dest.au
Jan 10 17:50:43.918:vtd_lg_incoming:Legs [71 ]VARTAG Translation Leg
Count=1

```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>debug voip ivr call setup</b>	Displays the call setup information.
<b>debug voip ivr digit collect</b>	Displays the digits collected during the call.
<b>debug voip ivr script</b>	Displays the scripts being processed.
<b>debug voip ivr telcommands</b>	Displays the TCL commands being called.

# Glossary

**CA**—call agent.

**ISP**—Internet service provider.

**IVR**—interactive voice response.

**media stream**—A single media instance. For example, an audio stream.

**MGC**—Media gateway controller.

**MGCP**—Media Gateway Control Protocol.

**PMH**—Protocol Message Handler.

**PSTN**—Public Switched Telephony Network.

**RTSP**—Real Time Streaming Protocol.

**RTSP session**—”Session” refers to a session between the application and the RTSP client. Each call leg that is configured to use RTSP streaming has a session.

**RTP**—Real Time Protocol. A protocol used to transport real time frames over a UDP network.

**TCL**—Tool Command Language.

**VoFR**—Voice over Frame Relay.

**VoIP**—Voice over IP.

