

Applications written in Voice eXtensible Markup Language (VoiceXML) provide access through a voice browser to content and services over the telephone, just as Hypertext Markup Language (HTML) provides access through a web browser running on a PC. The universal accessibility of the telephone and its ease of use makes VoiceXML applications a powerful alternative to HTML for accessing the information and services of the World Wide Web.

Guide Contents
<a href="#">Troubleshooting Cisco IOS Voice Overview</a>
<a href="#">Debug Command Output on Cisco IOS Voice Gateways</a>
<a href="#">Filtering Troubleshooting Output</a>
<a href="#">Cisco VoIP Internal Error Codes</a>
<a href="#">Troubleshooting Cisco IOS Voice Telephony</a>
<a href="#">Troubleshooting Cisco IOS Voice Protocols</a>
<a href="#">Troubleshooting Cisco IOS Telephony Applications</a>
<a href="#">Monitoring the Cisco IOS Voice Network</a>
<a href="#">Cause Codes and Debug Values</a>

## Contents

- [1 Cisco IOS VoiceXML Overview](#)
  - ◆ [1.1 Figure: Cisco IOS VoiceXML Application Components](#)
- [2 Debugging Cisco VoiceXML Applications](#)
  - ◆ [2.1 <cisco-debug>](#)
  - ◆ [2.2 SUMMARY STEPS](#)
  - ◆ [2.3 DETAILED STEPS](#)
- [3 Error Events](#)
  - ◆ [3.1 error.badfetch](#)
  - ◆ [3.2 error.semantic](#)
  - ◆ [3.3 error.unsupported.format](#)
- [4 JavaScript or ECMA Script](#)
- [5 Troubleshooting Speech Recognition and Synthesis](#)
  - ◆ [5.1 Table: Speech Recognition or Synthesis Fails](#)
- [6 Troubleshooting ASR and TTS Server Functionality](#)
  - ◆ [6.1 SUMMARY STEPS](#)
  - ◆ [6.2 DETAILED STEPS](#)

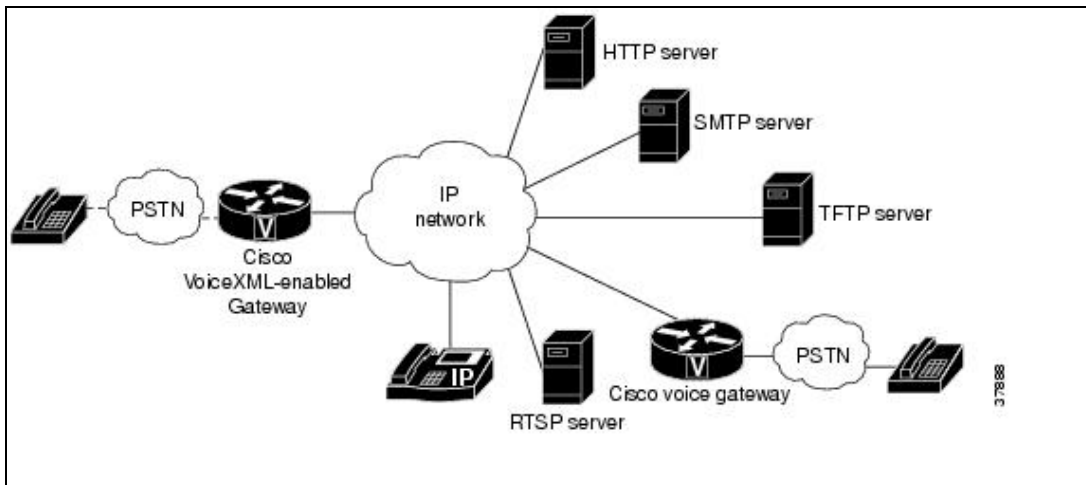
## Cisco IOS VoiceXML Overview

The Cisco IOS VoiceXML feature provides a platform for interpreting VoiceXML documents. When a telephone call is made to the Cisco VoiceXML-enabled gateway, VoiceXML documents are downloaded from web servers, providing content and services to the caller, typically in the form of pre-recorded audio in an IVR application. Customers can access online business applications over the telephone, providing for example, stock quotes, sports scores, or bank balances.

VoiceXML brings the advantages of web-based development and content delivery to voice applications. It is similar to HTML in its simplicity and in its presentation of information. The Cisco IOS VoiceXML feature is based on the *W3C VoiceXML 2.0 Working Draft* and is designed to provide web developers great flexibility and ease in implementing VoiceXML applications.

Figure: Cisco IOS VoiceXML Application Components shows components that can be configured as a part of a VoiceXML application installed on a Cisco voice gateway:

**Figure: Cisco IOS VoiceXML Application Components**



For information on developing a VoiceXML document for implementing an application on the Cisco voice gateway, refer to the *Cisco VoiceXML Programmer's Guide*.

This section describes some of the troubleshooting techniques for the Cisco VoiceXML features.

- [Debugging Cisco VoiceXML Applications](#)
- [Error Events](#)
- [JavaScript or ECMA Script](#)
- [Troubleshooting Speech Recognition and Synthesis](#)
- [Troubleshooting ASR and TTS Server Functionality](#)

For a list of the latest troubleshooting FAQs, go to the developer support website here:

[http://www.cisco.com/cgi-bin/dev\\_support/access\\_level/products.cgi?product=VOICE\\_XML\\_GATEWAY](http://www.cisco.com/cgi-bin/dev_support/access_level/products.cgi?product=VOICE_XML_GATEWAY)

## Debugging Cisco VoiceXML Applications


To debug Cisco VoiceXML applications at the gateway level, refer to the [Cisco IOS TCL and VoiceXML Application Guide](#).

This section describes troubleshooting at the script level. To troubleshoot Cisco VoiceXML scripts, enable the **debug vxml error** and **debug vxml puts** commands on the gateway. The **debug vxml error** command displays all errors on the console, and the **debug vxml puts** command prints debugging statements used with the <log> element in the VoiceXML document.

<cisco-debug>

<cisco-debug> is used to debug only a specific application. To disable debugging messages for all VoiceXML applications except the specific VoiceXML application you wish to debug, use the <cisco-debug> element in the VoiceXML document in conjunction with the **debug condition application voice** command.

Refer to the *Cisco IOS TCL and VoiceXML Application Guide* for information on debug commands.

 **Note:** Before you use Cisco IOS debug commands to debug a specific application, add <cisco-debug> to the VoiceXML document for the application you want to debug.

For example:


## SUMMARY STEPS


1. Turn on global debug.
2. Add the <cisco-debug enabled = "true"/> and <cisco-debug enabled = "false"/> elements around the specific part of the VoiceXML document where you want to see debugging messages.
3. Add conditional debugging to the specific application.

## DETAILED STEPS

1. Turn on global debug for the areas you want to debug. For example:

```
debug vxml application
debug vxml trace
```

 **Note:** If you do not proceed with step 2 and end your task with step 1, you see error messages for all the applications, irrespective of conditional debug being turned on or off.

 **Note:** The **debug condition application voice** command filters debugging output for only the **debug vxml** and **debug http client** commands. However, it does not filter output for the **debug vxml error**, **debug vxml background**, **debug http client error**, or **debug http client background** commands.

2. Add the <cisco-debug enabled = "true"/> and <cisco-debug enabled = "false"/> elements around the specific part of the VoiceXML document where you want to see debugging messages. For example:

```
<?xml version="1.0"?>
  <vxml version="1.0" application="root.vxml">
    <form>
      <block>
        <cisco-debug enabled = "true"/>
        <prompt>
          <audio src="welcome.au" caching="fast"/>
        </prompt>
        <cisco-debug enabled = "false"/>
        <goto next="getExtension.vxml?"/>
      </block>
    </form>
  </vxml>
```

3. Add conditional debugging to the specific application you want to debug. For example:

Three applications named myapp1, myapp2, and myapp3, all of which can be loaded by using the **call application voice** command are shown below:

```
call application voice myapp1 http://server1/vxml/test1.vxml
call application voice myapp2 http://server2/vxml/test2.vxml
call application voice myapp3 http://server3/vxml/test3.vxml
```

To debug only one of the applications, for example myapp1, use the **debug condition application voice** command to disable debug messages for the other applications, myapp2 and myapp3.

```
debug condition application voice myapp1
```

 **Note:**

<cisco-debug>

Debugging for myapp1 is performed for only those debug areas that have been enabled in step 1 above. Debugging for the specific session must be enabled through the <cisco-debug> tag as shown in step 2 above.

## Error Events

Enabling the **debug vxml error** command displays a list of possible error events on the console. For a list of error events, see the [Tel IVR Events and Status Codes](#) section.

Some of the possible errors generated with the **debug vxml error** command enabled are:

### error.badfetch

Possible Causes	Suggested Actions
<ul style="list-style-type: none"> <li>The VoiceXML interpreter throws this event when there is a failure in retrieving external components in the application. These external components can be VoiceXML documents, prerecorded files, or grammar files.</li> <li>A badfetch error usually occurs when there is an error in fetching an external document.</li> </ul>	<ul style="list-style-type: none"> <li>Verify that the external documents, audio prompts, or grammar files are available at the specified location mentioned in the URL.</li> <li>If the external components are stored on a HTTP server, enable the <b>debug http client error</b> command. * If the external components are stored on a RTSP server, search for error.badfetch.rtsp.xxx, where xxx is a RTSP response code. For values of RTSP response codes, refer to <a href="#">RFC 2326</a> available on the IETF website at <a href="http://www.ietf.org/">http://www.ietf.org/</a>.</li> </ul>

### error.semantic

Possible Causes	Suggested Actions
Logical errors such as referencing an undefined variable.	Verify that all variables referenced in the script are valid and defined.
Defining different grammar types in the same scope in the VoiceXML application.	Verify that only one grammar type is used at the time of recognizing user input.
Failure to define mandatory parameters in Cisco objects. For example, failure to define the account parameter in the authorize object results in a semantic error.	Verify that all mandatory parameters are defined in Cisco objects used in the script.

### error.unsupported.format

Possible Causes	Suggested Actions
A resource format is not supported by the platform.	Verify that all formats used in the script are supported by the specific platforms being used.

## JavaScript or ECMA Script

When the <script> element or ECMA expression is used in a VoiceXML document, enable the **debug java** command for debugging.

```
debug java ?
apm2- JavaScript APM2 Utility Debugging
error- JavaScript Error Debugging
interpreter- JavaScript Interpreter Debugging
```

## Troubleshooting Speech Recognition and Synthesis

Cisco IOS Release 12.2(11)T and later support automatic speech recognition (ASR) and text-to-speech (TTS) capabilities for VoiceXML and Tcl applications on Cisco voice gateways.

The Speech Recognition and Synthesis feature provides interfaces to ASR and TTS media servers by using Media Resource Control Protocol (MRCP), an application-level protocol developed by Cisco and its ASR and TTS media server partners, Nuance Communications and SpeechWorks International. Client devices that are processing audio or video streams use MRCP to control media resources on external media servers, such as speech synthesizers for TTS and speech recognizers for ASR. The Cisco gateway, running a voice application, and the media servers providing speech recognition and speech synthesis, maintain a client/server relationship through an RTSP connection; the gateway is the RTSP client and the RTSP server is the streaming media server providing speech recognition and speech synthesis.

While doing speech recognition, the gateway creates a separate G.711 u-law RTP stream to the media server, enabling the gateway to simultaneously perform speech synthesis or play audio files using a different codec.

If speech recognition or synthesis is not working, [Table: Speech Recognition or Synthesis Fails](#) lists some possible causes and the actions that you can take.

**Table: Speech Recognition or Synthesis Fails**

Possible Causes	Suggested Actions
Server is not configured either on the Cisco gateway or in the VoiceXML document.	Verify that the server location is configured by using at least one of these methods: <ul style="list-style-type: none"> <li>• Globally on the gateway by using the <b>ivr asr-server</b> or <b>ivr tts-server</b> command. Refer to the Cisco IOS TCL and VoiceXML Application Guide.</li> <li>• With the <i>com.cisco.asr-server</i> or <i>com.cisco.tts-server</i> property in the VoiceXML document. Refer to the <i>Cisco VoiceXML Programmer's Guide</i>.</li> </ul>
Gateway cannot access external ASR or TTS server or server is not running.	Ping the external server to make sure that the gateway has connectivity.
RTSP or MRCP errors are occurring between the gateway and the media server.	See the <a href="#">Troubleshooting ASR and TTS Server Functionality</a> .

## Troubleshooting ASR and TTS Server Functionality

### SUMMARY STEPS

1. **debug vxml error** and **debug vxml event**
2. **debug mrp error**
3. **debug rtsp error**, **debug rtsp session**, and **debug rtsp socket**

### DETAILED STEPS

1. Use the **debug vxml error** and **debug vxml event** commands to verify that the external media server is reachable and its location is configured on the gateway or in the VoiceXML document. In the following example, the application failed because the media server is not configured on the gateway or in the VoiceXML document.:

## Cisco\_IOS\_Voice\_Troubleshooting\_and\_Monitoring\_--\_Cisco\_VoiceXML\_Troubleshooting

```
Router# debug vxml error
Router# debug vxml event
*Jan 5 18:24:19.507: //62/36CA25A68036/VXML:/vxml_vapp_tts:
tftp://demo/sample/banking.vxml at line 17: vapp_tts() fail with vapp error 1
**Jan 5 18:24:19.507: //62/36CA25A68036/VXML:/vxml_event_proc:
*Jan 5 18:24:19.507: <event>: event=error.badfetch status=0
*Jan 5 18:24:19.507: //62/36CA25A68036/VXML:/vxml_default_event_handler:
use default event handler
*Jan 5 18:24:19.507: //62/36CA25A68036/VAPP:/vapp_session_exit_event_name: Exit Event
error.badfetch
*Jan 5 18:24:19.507: //62/36CA25A68036/VAPP:/vapp_session_exit_event_name: Exit Event
error.badfetch
*Jan 5 18:24:19.507: //62/36CA25A68036/VAPP:/vapp_session_exit_event_name: Exit Event Name
already set to error.badfetch
*Jan 5 18:24:19.507: //62/36CA25A68036/VXML:/vxml_vapp_terminate: vapp_status=0 ref_count 0
*Jan 5 18:24:19.507: //62/36CA25A68036/VXML:/vxml_vapp_terminate: vxml session terminating
with code=ERROR
vapp status=VAPP_SUCCESS vxml async status=VXML_ERROR_BAD_FETCH
```

In the following example, the application failed because the media server is either unreachable or is not running.

```
*Jan 5 18:36:44.451: //83/ECD9B163804B/VXML:/vxml_media_done: : media play failed to setup
with VAPP error=31, protocol_status_code=0
*Jan 5 18:36:44.451: <event>: event=error.com.cisco.media.resource.unavailable status=0
*Jan 5 18:36:44.451: //83/ECD9B163804B/VXML:/vxml_default_event_handler:
use default event handler
*Jan 5 18:36:44.451: //83/ECD9B163804B/VAPP:/vapp_session_exit_event_name: Exit Event
error.com.cisco.media.resource.unavailable
*Jan 5 18:36:44.451: //83/ECD9B163804B/VAPP:/vapp_session_exit_event_name: Exit Event
error.com.cisco.media.resource.unavailable
*Jan 5 18:36:44.451: //83/ECD9B163804B/VAPP:/vapp_session_exit_event_name: Exit Event Name
already set to error.com.cisco.media.resource.unavailable
*Jan 5 18:36:44.451: //83/ECD9B163804B/VXML:/vxml_vapp_terminate: vapp_status=0 ref_count 0
```

2. Use the **debug mrcp error** command to verify the connection between the gateway and the server. The following example shows the error when the RTSP connection to the server fails:

```
Router# 'debug mrcp error
*May 9 20:29:09.936:Connecting to 10.1.2.58:554 failed
```

The following error occurs when the response from the server is incorrect:

```
*May 9 20:29:09.936:Response from 10.1.2.58:554 failed
*May 9 20:29:09.936:MRCP/1.0 71422 COMPLETE
```

The following error occurs when the recognize request comes out of sequence:

```
*May 9 20:29:09.936:act_idle_recognize:ignoring old recognize request
```

3. Use the **debug rtsp error**, **debug rtsp session**, and **debug rtsp socket** commands to verify the RTSP connection with the media server, for example:

The following message displays if the RTSP connection fails:

```
*Sep 25 15:02:32.052: //-1//RTSP:/rtsplib_connect_to_svr: Socket Connect failed:
172.19.140.31:554
```

The following message displays if the RTSP client receives an incorrect response from the server:

## Cisco\_IOS\_Voice\_Troubleshooting\_and\_Monitoring\_--\_Cisco\_VoiceXML\_Troubleshooting

```
*Sep 25 15:03:35.062: //-1//RTSP:/rtsp_process_single_svr_resp: Parse Server Response failed,  
172.19.140.31:554
```

The following message displays if the codec configured on the IP side is not G.711:

```
*Sep 25 15:05:15.765: //-1//RTSP:/rtsplib_rtp_associate_done: Association mismatch
```