Monitoring and Troubleshooting Voice Applications

Event logs and statistics introduced in the Voice Application Monitoring and Troubleshooting Enhancements feature enable detailed monitoring of voice application instances and call legs. Records for terminated application instances and call legs are saved in history to assist in fault isolation. This comprehensive management information helps you diagnose problems in the network and identify the causes.

Tip To immediately begin using this feature, proceed to the "Enabling Event Logging and Statistics Globally for Voice Applications" section on page 10.

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

- "Overview of Cisco IOS Tcl IVR and VoiceXML Applications" on page 1
- Entire Cisco IOS Voice Configuration Library—including library preface and glossary, other feature documents, and troubleshooting documentation—at http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary/vcl.htm.

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

Feature History for Voice Application Monitoring and Troubleshooting Enhancements

Release	Modification
12.3(8)T	This feature was introduced.
12.3(14)T	A new command-line interface structure for configuring Tcl and IVR applications was introduced and affected the commands for configuring this feature.

Contents

Prerequisites for Voice Application Monitoring and Troubleshooting

- A Tcl IVR 2.0 or VoiceXML application must be configured on the voice gateway as described in Configuring Basic Functionality for Tcl IVR and VoiceXML Applications, or you must use one of the call applications contained in Cisco IOS software.
- Incoming telephony call legs including voice, fax, or modem.

Restrictions for Voice Application Monitoring and Troubleshooting

- Event logs for IP call legs are not supported.
- Tcl IVR 1.0 is not supported.
- Statistics and event logs for dynamically loaded scripts that are not configured on the gateway are not supported.

Information About Voice Application Monitoring and Troubleshooting Enhancements

To monitor and troubleshoot voice applications on the Cisco voice gateway, you should understand the following concepts:

Description of Voice Application Monitoring and Troubleshooting Features

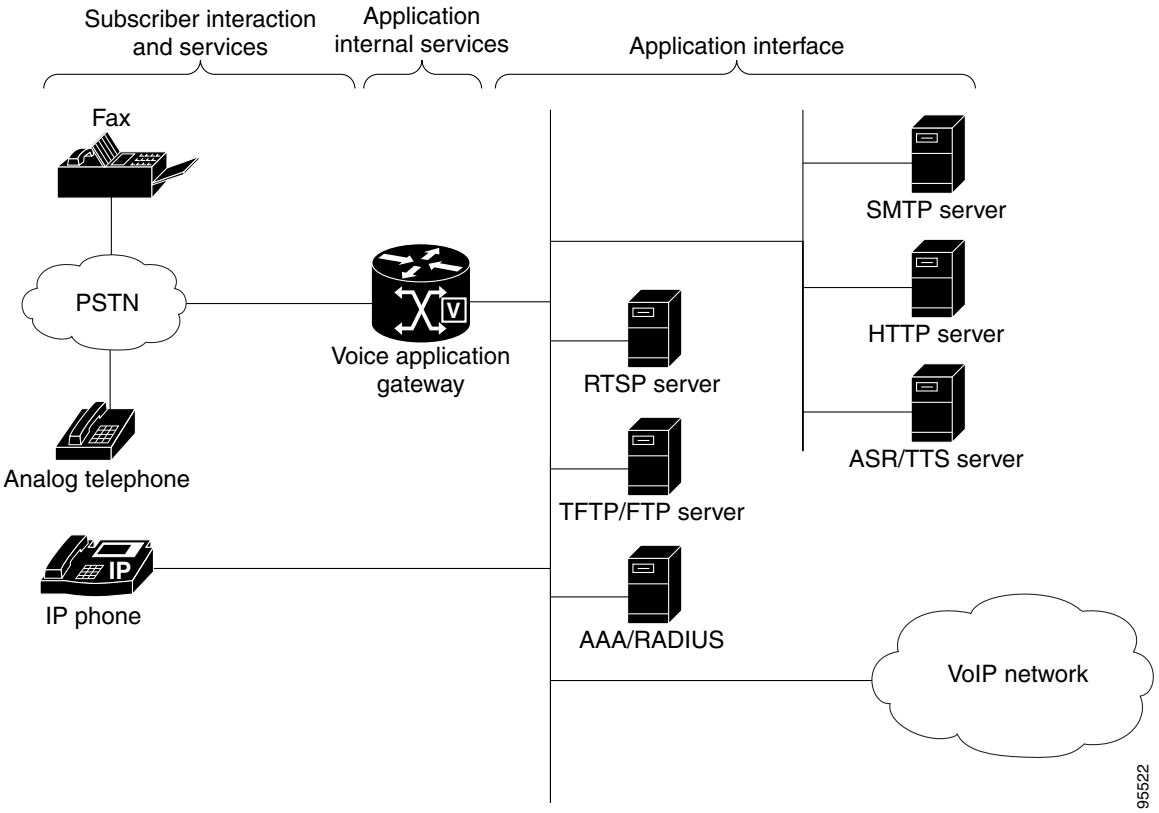
Voice calls are setup, maintained, and terminated by different subsystems within Cisco IOS software. Before Cisco IOS Release 12.3(x)T, managing calls from a voice application perspective could be challenging when diagnosing call progress, delays, exceptions, and failures during post call-analysis. Although debug commands are available for the individual subsystems, the volume of output and the expertise required to understand all the subsystems make it difficult to use **debug** commands for identifying the cause of a failure or gathering specific details about a call.

This feature enables detailed monitoring of call applications by providing command-based event logging and statistics collection for voice application instances, application interfaces, and call legs on the Cisco voice gateway. The event logs and statistics provide a high-level view of application transactions in simple, nontechnical language. Event logs include subscriber dialog interactions and back-end server transactions. This event logging is separate from the generic message logging available with the **logging buffered** command. You can enable monitoring globally for all voice applications and application interfaces, or for individual applications, interfaces, or servers. These comprehensive management features help you diagnose performance problems and isolate faults in a production-level network.

After an application instance terminates, the event log and statistics information is moved from the active table to the history table in the application information system (AIS) database. You can view the event logs and statistics by using **show** commands on the voice gateway or by using SNMP to access the MIB call application tables. You can set the size of the event log buffer to meet your needs and the memory resources of your voice gateway. You can also configure the voice gateway to write the event logs to an external FTP server. The gateway saves the event logs to the selected server when the event log buffer becomes full or when the application instance or call leg terminates.

Figure 12-1 shows the major subsystem components involved in voice application calls.

Figure 12-1 Voice Application Network Components



The voice application monitoring features provide:

- Statistics and event logs for voice applications including Tcl IVR 2.0, VoiceXML, T.37 (on-ramp/off-ramp), and the applications contained in Cisco IOS software such as the default session application.

- Counters maintained per application instance. Gauges maintained for application instance, application, and gateway levels.
- Counters and gauges for various external interfaces used by applications including AAA, ASR, HTTP, RTSP, SMTP, TFTP, and TTS servers, and flash memory and the gateway memory.
- Event logs in plain English for each application instance.
- Counters and gauges accessible through SNMP by call application MIB, or through Cisco IOS commands and posted to FTP server on request.
- Event logs for voice call legs.

Counters and Gauges for Voice Application Statistics

Statistics for voice applications and call legs are presented as counters and gauges. These measurements are based on event counts or rates. They measure the total count of a specific event, or the derived rate of change of events during a period of time. Counters track totals, such as the number of errors, and do not reset to zero unless cleared. Gauges reflect real-time information for active calls, such as the number of currently connected calls. Gauges fluctuate over time and can return to zero. For example, a call is included in an active call gauge while it is active. When the call terminates, the active call gauge is decreased while counters in the history table increase. Gauges are available at the gateway level and application level for active calls and instances.

The accumulated statistics are stored in the gateway. You can configure the maximum duration to store the historical records according to your needs and the availability of system memory. You can display the application statistics on the gateway by using **show** commands.

Monitoring Levels for Voice Applications

Statistics and event logs for voice application instances are organized in a hierarchical model, providing a top-down approach to monitoring. You can detect faults or performance irregularities at a global level, then drill down to more specific details to isolate a fault. Data is available for both active application instances and terminated instances. After an application instance terminates, its data records move from active to history.

You monitor voice applications at three levels. Data is collected for each application instance at the lowest, most detailed level and propagated up to the highest level where the data is consolidated and presented in summary form. The three monitoring levels are:

- Gateway level—High-level summary of all applications running on the Cisco voice gateway. The statistics are cumulative compiled from all instances of all applications on the gateway. Because there is only one record per gateway, the amount of information is brief and does not impact gateway resources so you can request it regularly. In most scenarios, you begin the monitoring process by displaying the gateway-level statistics with the **show call application gateway-level** command. An error condition might be shown or a value might exceed a threshold, indicating a fault in one or more application instances on the gateway. After an application instance terminates, its gateway-level statistics are added to the history counters.
- Application level—Statistics representing all instances of a particular application. There may be multiple records, one record for each configured application. If several records show the same error, you can identify which application is involved and ignore the others without the error. You display application-level statistics with the **show call application app-level** command. Statistics from this

level are propagated up to the gateway level where they are consolidated with statistics for all other applications on the gateway. After an application instance terminates, its application-level statistics are added to the history counters.

- Application instance (session) level—Detailed information for specific application instances. You display the data at this level by using the **show call application session-level** command. You can focus on the application instances that are affected by looking for records with similar errors. Event logs are also available at this level providing further details about an application instance. Statistics from this level are propagated up to the application level where they are consolidated by application. After an application instance terminates, its session records are appended to history.

You can also display application interface statistics for transactions between applications and back-end servers. For information, see the **show call application interface** command.

Table 12-1 lists the type of records available at each monitoring level.

Table 12-1 Statistics and Event Log Output

Level	Statistics		Event-Log	
	Active	**History**	**Active**	**History**
Gateway	Gauges	Counters	No	No
Application	Gauges	Counters	No	No
Application instance	Gauges, counters	Counters	Yes	Yes

Figure 12-2 illustrates the monitoring levels for identifying application instances affected by an error.

Figure 12-2 Application Monitoring Levels

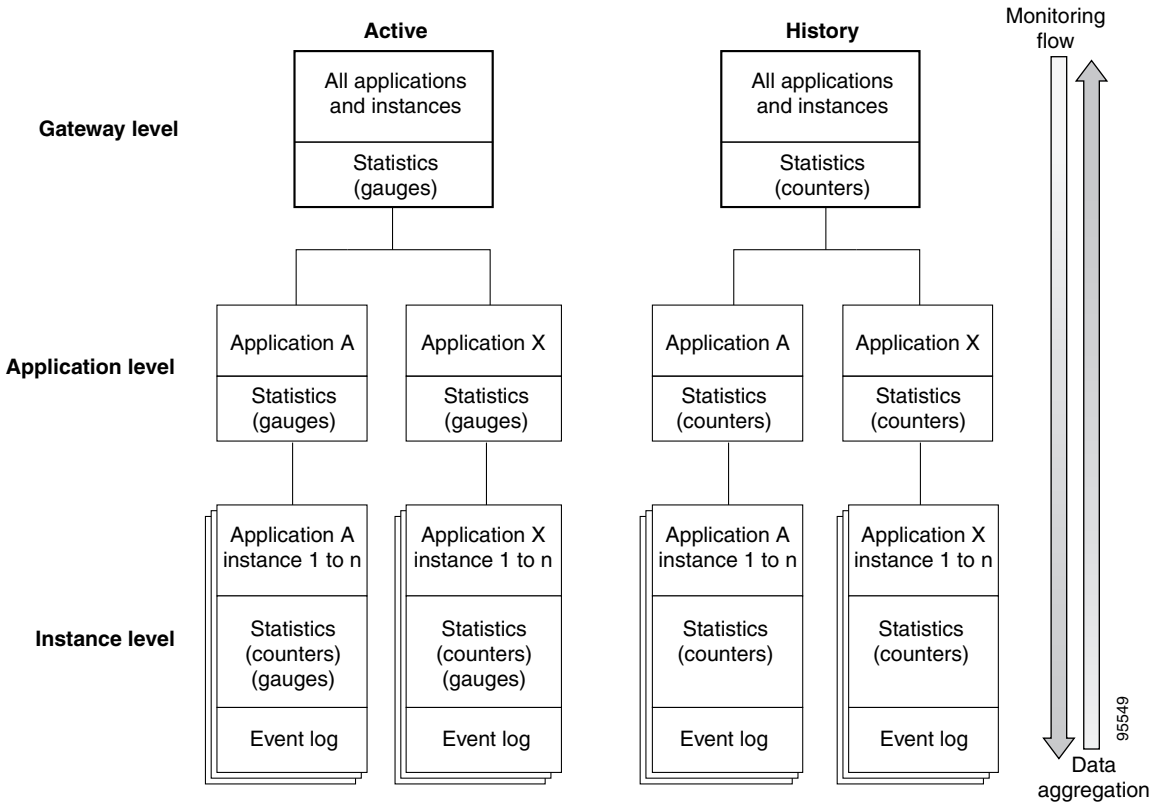


Figure 12-3 shows the type of information available for application interfaces.

Figure 12-3 Application Interface Event Logs and Statistics

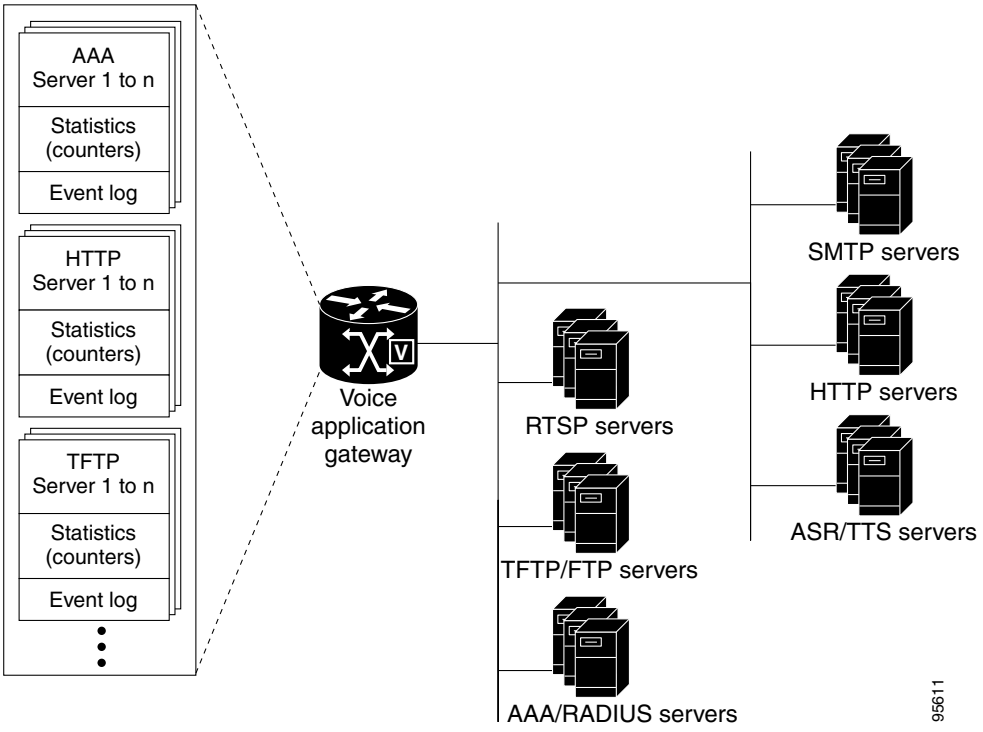
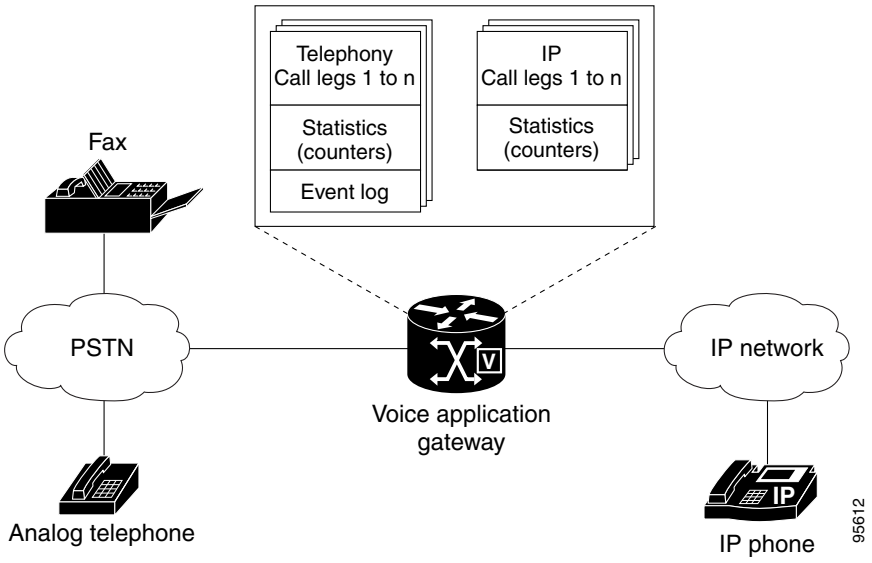


Figure 12-4 shows the type of information available for voice call legs.

Figure 12-4 Call Leg Event Logs and Statistics



Application Instance Statistics

The following statistics are available for application instances.

Subscriber Interaction

- DTMF attempts, matches, no matches, no input, and long pound
- ASR attempts, matches, no matches, and no input
- AAA authentication successes and failures
- AAA authorization successes and failures

Subscriber Services

- For incoming and outgoing PSTN and IP call legs:
  - Call legs setup, currently connected, and total legs connected
  - Call legs handed off, incoming and outgoing
  - Call legs handed off and returned, incoming and outgoing
  - Call legs disconnected for cause of normal, user error, or system error
- Media attempts, successes, failures, and currently active sessions for prompt playout, recording, and TTS on call legs seen by an application instance.

Application Internal Services

- Bridged handoffs, returned bridged handoffs, blind handoffs, and handoff failures for application instances
- Place call requests, successes, and failures
- Document fetch requests, successes, and failures
- Document submit requests, successes, and failures
- ASNL subscription attempts, successes, pending, failures, and notifications received

For a description of each statistic and an example of the output, see the **show call application session-level** command in the *Cisco IOS Voice Command Reference, Release 12.3T* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123tcr/123tvr/index.htm.

Application Interface Statistics

Statistics for application interfaces are available at the gateway level. The statistics are summarized for the gateway and can be viewed by service type or for a specific server. The following statistics are provided for each of the service types: AAA, ASR, flash memory, HTTP, gateway memory, RTSP, SMTP, TFTP, TTS.

- Read requests, successes, and failures
- Write requests, successes, and failures
- Bytes transmitted and received
- Rate (minimum, maximum, and average)

For a description of the specific statistics supported for each interface type, and an example of the output, see the **show call application interface** command in the *Cisco IOS Voice Command Reference, Release 12.3T* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123tcr/123tvr/index.htm.

Benefits of Voice Application Monitoring and Troubleshooting Enhancements

- Event logs are captured at run time unlike debug commands that are enabled after a problem occurs.

- Event log output uses simple wording unlike debug output that includes information specific to the internal system.
- Event logs can be selectively started and saved in the voice gateway startup configuration unlike debug commands that must be executed after each gateway reload.

Guidelines for Enabling Statistics and Event Logging for Voice Applications

The voice application monitoring and troubleshooting enhancements provide separate commands for enabling statistics and event logs. Statistics are useful for both monitoring and troubleshooting and you can enable statistics collection without a noticeable impact on system performance.

Event logging is designed primarily for troubleshooting faults after an error is indicated in the statistics. Depending on the amount of gateway traffic and the type of event logs or event log options that are enabled, the gateway could experience performance issues because of the impact on processor and memory resources. Use the following guidelines to selectively enable event logging:

- Enable event logging only when troubleshooting a problem
- Enable only the specific type of event log that is required to troubleshoot the problem (application, back-end server, call leg)
- Enable event logging only for a specific application if using application event logging

Throttling Mechanism for Event Logging

Event logging consumes processor memory to store the event logs. To prevent event logging from adversely impacting system resources for production traffic, the gateway uses a throttling mechanism. When free processor memory drops below 20%, the gateway automatically disables event logging for all new application sessions, application interfaces, and call legs for which logging is enabled. It resumes event logging when free memory rises above 30%.

While throttling is occurring, the gateway does not capture any new event logs even if event logging is enabled through Cisco IOS commands. You should monitor free memory on the gateway and enable event logging only when necessary for isolating faults.

Memory Requirements for Writing Event Logs to FTP

You can enable the gateway to write event logs to an external FTP server. This could adversely impact gateway memory resources in some scenarios, for example, when:

- The gateway is consuming high processor resources and FTP does not have enough processor resources to flush the logged buffers to the FTP server.
- The designated FTP server is not powerful enough to perform FTP transfers quickly enough
- Bandwidth on the link between the gateway and the FTP server is not large enough
- The gateway is receiving a high volume of short-duration calls or calls that are failing

You should enable FTP dumping only when necessary and not enable it in situations where it might adversely impact your system performance.

How to Configure Monitoring for Voice Applications

This section contains the following procedures:

Enabling Event Logging and Statistics Globally for Voice Applications

Perform this task to enable event logs and statistics globally for all voice application instances, application interface types, and call legs.

Note This procedure enables event logs and statistics globally for all applications, application interface types, and call legs. To enable or disable event logs for specific applications or interface types, see the "Enabling Event Logging for a Specific Voice Application or Interface" section on page 12.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **monitor**
5. **event-log**
6. **stats**
7. **interface event-log**
8. **interface stats**
9. **exit**
10. **exit**
11. **call event-log**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)#application`	Enters application configuration mode.

Step 4	`monitor`
	
	Example:
	`Router(config-app)# monitor`
		Enters application configuration monitor mode.

Step 5	**event-log** **Example:** `Router(config-app-monitor)# event-log`	Enables event logging for voice application instances.
Step 6	**stats** **Example:** `Router(config-app-monitor)# stats`	Enables statistics collection for voice applications.
Step 7	**interface event-log** **Example:** `Router(config-app-monitor)# interface event-log`	Enables event logging for all external interfaces used by voice applications.
Step 8	**interface stats** **Example:** `Router(config-app-monitor)# interface stats`	Enables statistics collection for application instances.
Step 9	**exit** **Example:** `Router(config-app-monitor)# exit`	Exits the application configuration monitor mode.
Step 10	**exit** **Example:** `Router(config-app)# exit`	Exits the application configuration mode.

	Command or Action	Purpose

Step 11	`call leg event-log`
	
	Example:
	`Router(config)# call leg event-log`
		Enables transaction event logging for voice, fax, and modem call legs.

Step 12	**`exit`** **Example:** `Router# exit`	Exits the current mode.

Enabling Event Logging for a Specific Voice Application or Interface

Perform this task to enable event logging and statistics collection for a specific voice application, application interface type, or server.

Note This procedure enables or disables event logs for specific applications, interface types, or servers. If you have already enabled event logs globally by performing the task in the "Enabling Event Logging and Statistics Globally for Voice Applications" section on page 10, you can skip this task unless you want to individually disable selected applications, interface types, or servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service** *application-name*
5. **paramspace appcommon event-log** [**enable** | **disable**]
6. **exit**
7. **monitor**
8. **interface event-log** {**aaa** | **asr** | **flash** | **http** | **ram** | **rtsp** | **smtp** | **tftp** | **tts**} [**server** *server*] [**disable**]
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)#application`	Enters application configuration mode.
Step 4	**service** *application-name* **Example:** `Router(config-app)# service session`	Enter parameter configuration mode for the application.
Step 5	**paramspace appcommon event-log** [**enable** | **disable**] **Example:** `Router# paramspace appcommon event-log enable`	Enables event logging for a specific voice application.
Step 6	**exit** **Example:** `Router(config-app)# exit`	Exits configuration submode.
Step 7	**monitor** **Example:** `Router(config-app)#  monitor`	Enters application monitor configuration mode.
Step 8	**interface event-log** {**aaa** | **asr** | **flash** | **http** | **ram** | **rtsp** | **smtp** | **tftp** | **tts**} [**server** *server*] [**disable**]] **Example:** `Router(config-app-monitor)# interface event-log http disable`	Enables event logging for a specific external interface used by voice applications.
Step 9	**exit** **Example:** `Router# exit`	Exits the current mode.

Monitoring Voice Applications for Active Calls

Perform this task to do basic load monitoring of voice applications when detecting faults or unacceptable ranges while a call is still active.

SUMMARY STEPS

1. **enable**
2. **show call application active gateway-level**
3. **show call application active app-level summary**
4. **show call application active app-level app-tag** *application-name*
5. **show call application active session-level summary**
6. **show call application active session-level session-id** *session-id*
7. **show call application interface summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show call application active gateway-level** **Example:** `Router# show call application active gateway-level`	Displays gateway-level statistics for all currently active voice application instances.
Step 3	**show call application active app-level summary** **Example:** `Router# show call application active app-level summary`	Displays application-level statistics for all active voice applications.
Step 4	**show call application active app-level app-tag** *application-name* **Example:** `Router# show call application active app-level app-tag sample_app`	Displays application-level statistics for a specific voice application.
Step 5	**show call application active session-level summary** **Example:** `Router# show call application active session-level summary`	Displays statistics for all currently active voice application instances.

Step 6	**show call application active session-level session-id** *session-id* **Example:** `Router# show call application active session-level session-id 5`	Displays event logs and statistics for a specific voice application instance.
Step 7	**show call application interface summary** **Example:** `Router# show call application interface summary`	Displays aggregated statistics for all application interfaces.

For examples of the command output for the steps in this task, see the "Examples for Monitoring Voice Applications" section on page 16.

Monitoring Voice Applications for Terminated Calls

Perform this task to monitor voice applications, detect faults, and identify the source of faults after a call terminates.

SUMMARY STEPS

1. **enable**
2. **show call application history gateway-level**
3. **show call application history app-level summary**
4. **show call application history app-level app-tag** *application-name*
5. **show call application history session-level summary**
6. **show call application history session-level session-id** *session-id*
7. **show call application interface summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show call application history gateway-level** **Example:** `Router# show call application history gateway-level`	Displays gateway-level statistics for all voice application instances in the history table.
Step 3	**show call application history app-level summary** **Example:** `Router# show call application history app-level summary`	Displays summarized statistics from the history table for all voice applications.

Step 4	**show call application history app-level app-tag** *application-name* **Example:** `Router# show call application history app-level app-tag sample_app`	Displays statistics from the history table for a specific voice application.
Step 5	**show call application history session-level summary** **Example:** `Router# show call application history session-level summary`	Displays statistics from the history table for all voice application instances.
Step 6	**show call application history session-level session-id** *session-id* **Example:** `Router# show call application history session-level session-id 5`	Displays event logs and statistics for the specific voice application instance.
Step 7	**show call application interface summary** **Example:** `Router# show call application interface summary`	Displays aggregated statistics for all application interfaces.

Examples for Monitoring Voice Applications

This section includes output examples demonstrating the task for isolating faults in your voice application network. In this sample scenario, one of the audio prompts used by an application is not available from the specified TFTP server, either because the gateway does not have connectivity to the server or the audio file is not located on the specified server.

Sample Output for show call application history gateway-level Command

In the following example, the **show call application history gateway-level** command shows that a media failure occurred during a prompt playout. The counters in the output are an aggregation of all application instances that were run and terminated on the gateway. You must next determine which application had the error.

```
Router# show call application history gateway-level

Gateway level statistics for history application sessions:
Sessions w/ stats:             7
Last reset time:               *Aug 26 17:56:34 PST
 Subscriber Service - Call
                                                PSTN                VOIP
                                          Incoming Outgoing  Incoming Outgoing
Legs setup:                               7        0         0        0
Total legs connected:                     7        0         0        0
Legs handed in:                           0        0         0        0
Legs handed in returned back:             0        0         0        0
Legs handed out:                          0        0         0        0
Legs handed out came back:                0        0         0        0
Legs disconnected normally:               7        0         0        0
Legs disconnected for user error:         0        0         0        0
Legs disconnected for system error:       0        0         0        0

 Subscriber Service - Media
```

```
                                        Play        Record      TTS
Media attempts:                         9           0           1
Media successes:                        8           0           1
Media aborts:                           0           0           0
Media failures:                         1           0           0
Total media duration (in seconds):      57          0           17

 Application Internal Service - Document Read-Write
                                        Read        Write
Doc requests:                           1           0
Doc successes:                          1           0
Doc failures:                           0           0

 Subscriber Interaction - DTMF
DTMFs not matched:                      0
DTMFs matched:                          3
DTMFs no input:                         0
DTMFs long pound:                       0
```

Sample Output for show call application history app-level summary Command

In the following example, the **show call application history app-level summary** command lists all applications that are loaded on the gateway and shows that an error occurred with the application named menu. You can then view statistics for this specific application.

```
Router# show call application history app-level summary

Application level history Info:
                                  Sessions                      Last Reset
App Name                 Stats w/ Stats   Total      Errors     Time
session                  N     0          0          0
fax_hop_on               N     0          0          0
clid_authen              N     0          0          0
clid_authen_collect      N     0          0          0
clid_authen_npw          N     0          0          0
clid_authen_col_npw      N     0          0          0
clid_col_npw_3           N     0          0          0
clid_col_npw_npw         N     0          0          0
Default                  N     0          0          0
lib_off_app              N     0          0          0
fax_on_vfc_onramp_app    N     0          0          0
debitcard_20             N     0          0          0
aaa_tcl                  N     0          0          0
coapp                    N     0          0          0
doc_write                Y     1          1          0          *Aug 26 18:21:07
generic                  Y     3          3          0          *Aug 26 18:05:26
menu                     Y     3          3          1          *Aug 26 18:42:48
```

Sample Output for show call application history app-level app-tag Command

In the following example, the **show call application history app-level app-tag** command for the application named menu shows that there was a media failure when the application tried to play an audio prompt. You must next determine which instance of the application had the error.

```
Router# show call application history app-level app-tag menu

Application level history Info:

Application name:              menu
URL:                           tftp://sample/menu.vxml
Total sessions:                3
Sessions w/ stats:             3
Last reset time:               *Aug 26 18:42:48 PST
```

```
 Subscriber Service - Call
                                                PSTN              VOIP
                                        Incoming Outgoing  Incoming Outgoing
Legs setup:                             3        0         0        0
Total legs connected:                   3        0         0        0
Legs handed in:                         0        0         0        0
Legs handed in returned back:           0        0         0        0
Legs handed out:                        0        0         0        0
Legs handed out came back:              0        0         0        0
Legs disconnected normally:             3        0         0        0
Legs disconnected for user error:       0        0         0        0
Legs disconnected for system error:     0        0         0        0

 Subscriber Service - Media
                                        Play       Record     TTS
Media attempts:                         6          0          0
Media successes:                        5          0          0
Media aborts:                           0          0          0
Media failures:                         1          0          0
Total media duration (in seconds):      29         0          0

 Subscriber Interaction - DTMF
DTMFs not matched:                      0
DTMFs matched:                          3
DTMFs no input:                         0
DTMFs long pound:                       0
```

Sample Output for show call application history session-level summary Command

In the following example, the **show call application history session-level summary** command shows all the instances in history for the menu application. You can next review the statistics and event log for the specific instance that had the error, which has a session ID of 14.

```
Router# show call application history session-level summary

SID  Application Name       Stat Err Cnt    Log Stop Time       Duration
7    generic                Y    0          Y   *Aug 26 18:05:2 00:00:12
8    generic                Y    0          Y   *Aug 26 18:05:3 00:00:10
9    generic                Y    0          Y   *Aug 26 18:05:4 00:00:09
D    doc_write              Y    0          Y   *Aug 26 18:21:0 00:00:17
12   menu                   Y    0          Y   *Aug 26 18:42:4 00:00:11
13   menu                   Y    0          Y   *Aug 26 18:43:0 00:00:13
14   menu                   Y    1          Y   *Aug 26 18:43:1 00:00:09
```

Sample Output for show call application history session-level session-id Command

In the following example, the **show call application history session-level session-id** command for session 14 shows the event log for this instance. The event log shows the name and location of the audio prompt for which the error occurred. You can next review statistics for your application interfaces.

```
Router# show call application history session-level session-id 14

Session Info:
Session id:          14
Session name:
Application name:    menu
Application URL:     tftp://sample/menu.vxml
Start time:          *Aug 26 18:43:09 PST
Stop time:           *Aug 26 18:43:19 PST

Statistics:
 Subscriber Service - Call
                                                PSTN              VOIP
                                        Incoming Outgoing  Incoming Outgoing
```

```
Legs setup:                              1        0          0         0
Total legs connected:                    1        0          0         0
Legs handed in:                          0        0          0         0
Legs handed in returned back:            0        0          0         0
Legs handed out:                         0        0          0         0
Legs handed out came back:               0        0          0         0
Legs disconnected normally:              1        0          0         0
Legs disconnected for user error:        0        0          0         0
Legs disconnected for system error:      0        0          0         0

 Subscriber Service - Media
                                         Play       Record     TTS
Media attempts:                          2          0          0
Media successes:                         1          0          0
Media aborts:                            0          0          0
Media failures:                          1          0          0
Total media duration (in seconds):       9          0          0

 Subscriber Interaction - DTMF
DTMFs not matched:                       0
DTMFs matched:                           1
DTMFs no input:                          0
DTMFs long pound:                        0

Event log:
buf_size=4K, log_lvl=INFO
<ctx_id>:<timestamp>:<seq_no>:<severity>:<msg_body>
14:1061952189:450:INFO: Session started for App-type = menu, URL = tftp://sample/menu.vxml
14:1061952189:451:INFO: Incoming Telephony call received, LegID = 35
14:1061952189:452:INFO: LegID = 35: Calling = 4089023198, called = 52927, dial peer = 1
14:1061952189:453:INFO: LegID = 35: Leg State = LEG_INCCONNECTED
14:1061952189:454:INFO: Playing prompt #1: tftp://sample/audio/menu.au
14:1061952198:458:INFO: Prompt playing finished successfully.
14:1061952199:459:INFO: DTMF digit matched pattern v0, user input = 2
14:1061952199:460:INFO: Playing prompt #1: tftp://demo/audio/spanish_menu.au
14:1061952199:463:ERR : Prompt play setup failure.
14:1061952199:464:INFO: Script received event = "error.badfetch"
14:1061952199:465:INFO: LegID = 35: Call disconnected, cause = normal call clearing (16)
14:1061952199:468:INFO: Session done, terminating cause =
```

Sample Output for show call application interface summary Command

In the following example, the s**how call application interface summary** command shows statistics for each of your interface types including TFTP where the error occurred. You can next display the event log for the specific TFTP server.

```
Router# show call application interface summary

Aggregated statistics for tts service:
Stats last reset time *Aug 26 18:41:01 PST
Read requests:                   0
Read successes:                  0
Read failures:                   0
Read aborts:                     0

Aggregated statistics for asr service:
Stats last reset time *Aug 26 18:41:01 PST
Read requests:                   0
Read successes:                  0
Read failures:                   0
Read aborts:                     0

Aggregated statistics for tftp service:
Stats last reset time *Aug 26 18:41:01 PST
```

```
Read requests:                   2
Read successes:                  5
Read failures:                   1
Read aborts:                     0
Total bytes read:                255047
```

Sample Output for show call application interface Command

In the following example, the **show call application interface** command with the **tftp** keyword shows the event log for the TFTP server named demo. The event log shows that the gateway could not play the audio prompt because it could not find the TFTP server.

```
Router# show call application interface tftp

Server name:         sample

Statistics:
Last reset time *Aug 26 18:41:01 PST
Read requests:                   2
Read successes:                  2
Read failures:                   0
Read aborts:                     0
Total bytes read:                255047

Event log:
Last reset time *Aug 26 18:05:13 PST
buf_size=4K, log_lvl=INFO
<ctx_id>:<timestamp>:<seq_no>:<severity>:<msg_body>
sample:1061949913:173:INFO: ID = 653088C8: Read requested for URL =
tftp://sample/audio/menu.au
sample:1061949919:176:INFO: ID = 653088C8: Streamed read transaction Successful URL =
tftp://sample/audio/menu.au
sample:1061952156:416:INFO: ID = 651E5D6C: Read requested for URL =
tftp://sample/audio/welcome_test.au
sample:1061952166:423:INFO: ID = 651E5D6C: Streamed read transaction Successful URL =
tftp://sample/audio/welcome_test.au
-------------------------------------------------------------------------------
Server name:         demo

Statistics:
Last reset time *Aug 26 18:41:01 PST
Read requests:                   1
Read successes:                  0
Read failures:                   1
Read aborts:                     0
Total bytes read:                0

Event log:
Last reset time *Aug 26 18:38:05 PST
buf_size=4K, log_lvl=INFO
<ctx_id>:<timestamp>:<seq_no>:<severity>:<msg_body>
demo:1061952199:461:INFO: ID = 6542D968: Read requested for URL =
tftp://demo/audio/spanish_menu.au
demo:1061952199:462:ERR : ID = 6542D968: Read transaction failed URL =
tftp://demo/audio/spanish_menu.au, reason = IFS error 65540=Invalid IP address or hostname
-------------------------------------------------------------------------------
```

Monitoring Voice Call Legs

Perform this task to do basic monitoring of voice call legs. This can help you diagnose problems that occur during call setup before the call reaches the application.

Note Statistics for call legs are enabled by default. If you have previously enabled call leg event logs by following the steps in the "Enabling Event Logging and Statistics Globally for Voice Applications" section on page 10, you can skip directly to Step 5 of this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call leg event-log**
4. **exit**
5. **show call leg** {**active** | **history**} [**summary** | [**last** *number* | **leg-id** *leg-id*] [**event-log** | **info**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call leg event-log** **Example:** `Router(config)# call leg event-log`	Enables transaction event logging for voice, fax, and modem call legs.
Step 4	**exit** **Example:** `Router# exit`	Exits the current mode.
Step 5	**show call leg** {**active** | **history**} [**summary** | [**last** *number* | **leg-id** *leg-id*] [**event-log** | **info**]] **Example:** `Router# show call leg history last 2`	Displays call leg event logs or statistics.

What To Do Next

- To clear accumulated statistics and event logs for applications and application interfaces, see the "Clearing Event Logs and Statistics for Application Instances and Interfaces" section on page 22.
- To monitor events for application instances or call legs as they are occur, see the "Displaying Event Logs for Applications or Call Legs in Real-Time" section on page 22.

- To modify the default settings for event logs, see the following sections:
  - Modifying Event Log Settings for Application Instances, page 23
  - Modifying Event Log Settings for Application Interfaces, page 25
  - Modifying Event Log Settings for Call Legs, page 26
  - Modifying Event Log History Limits, page 27

Clearing Event Logs and Statistics for Application Instances and Interfaces

Perform this task to reset statistic counters to zero.

SUMMARY STEPS

1. **enable**
2. **clear call application** [**app-tag** *application-name*] **stats**
3. **clear call application interface** [[{**aaa** | **asr** | **flash** | **http** | **ram** | **rtsp** | **smtp** | **tftp** | **tts**} [**server** *server*]] [**event-log** | **stats**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear call application** [**app-tag** *application-name*] **stats** **Example:** `Router# clear call application app-tag sample_app stats`	Clears application-level statistics in the history table and subtracts the statistics from the gateway level, for all applications or a specific application.
Step 3	**clear call application interface** [[{**aaa** | **asr** | **flash** | **http** | **ram** | **rtsp** | **smtp** | **tftp** | **tts**} [**server** *server*]] [**event-log** | **stats**]] **Example:** `Router# clear call application interface http event-log`	Clears application interface statistics or event logs for all interfaces or a specific interface type or server.

Displaying Event Logs for Applications or Call Legs in Real-Time

Perform this task to display event logs for active calls as the events occur. If you are debugging or testing a script in a production-level network, the high call volume can make it difficult to select and view a specific event log while a call is still active. This task enables dynamic logging so that you can view events as they happen for active application instances or call legs. The output continues until the call terminates or you stop the display by using the **stop** keyword.

SUMMARY STEPS

1. **enable**
2. **monitor call application event-log** {**app-tag** *application-name* {**last** | **next**} | **session-id** *session-id* [**stop**] | **stop**}
3. **monitor call leg event-log** {**leg-id** *leg-id* [**stop**] | **next** | **stop**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor call application event-log** {**app-tag** *application-name* {**last** | **next**} | **session-id** session-*id* [**stop**] | **stop**} **Example:** `Router# monitor call application event-log session-id 5`	Displays events for the most recently active call or the next new call for a specific application, or for a specific instance.
Step 3	**monitor call leg event-log** {**leg-id** *leg-id* [**stop**] | **next** | **stop**} **Example:** `Router# monitor call leg event-log leg-id 5`	Displays events for next active call leg, or specific call leg.

Modifying Event Log Settings for Application Instances

Perform this task to modify the default settings for the event logs that are generated for voice application instances.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **monitor**
5. **event-log dump ftp** *server*[**:***port*]*/file* **username** *username* **password** [*encryption-type*] *password*
6. **event-log max-buffer-size** *kbytes*
7. **event-log error-only**
8. **exit**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)#application`	Enters application configuration mode.

Step 4	`monitor`
	
	Example:
	`Router(config-app)# monitor`
		Enters application configuration monitor mode.

Step 5	**event-log dump ftp** *server*[**:***port*]**/***file* **username** *username* **password** [*encryption-type*] *password* **Example:** `Router(config-app-monitor)# event-log dump ftp ftp-server/elogs/app_elogs.log username myname password 0 mypass`	(Optional) Specifies the location of the external file to which the gateway writes the contents of the event log buffer.
Step 6	**event-log max-buffer-size** *kbytes* **Example:** `Router(config-app-monitor)# event-log max-buffer-size 8`	(Optional) Sets the maximum size of the event log buffer for each application instance.
Step 7	**event-log error-only** **Example:** `Router(config-app-monitor)# event-log error-only`	(Optional) Logs only error events for application instances.
Step 8	**exit** **Example:** `Router(config-app-monitor)# exit`	Exits the application configuration monitor mode.
Step 9	**exit** **Example:** `Router(config-app)# exit`	Exits the application configuration mode.

Modifying Event Log Settings for Application Interfaces

Perform this task to modify the default settings for event logs generated for types of server interfaces that communicate with voice applications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **monitor**
5. **interface event-log dump ftp** *server*[**:***port*]*/file* **username** *username* **password** [*encryption-type*] *password*
6. **interface event-log max-buffer-size** *kbytes*
7. **interface max-server-records** *number*
8. **interface event-log error-only**
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)#application`	Enters application configuration mode.

Step 4	`monitor`
	
	Example:
	`Router(config-app)# monitor`
		Enters application configuration monitor mode.

	Command or Action	Purpose
Step 5	**event-log dump ftp** *server*[**:***port*]**/***file* **username** *username* **password** [*encryption-type*] *password* **Example:** `Router(config-app-monitor)# interface event-log dump ftp ftp-server/elogs/int_elogs.log username myname password 0 mypass`	(Optional) Specifies the location of the external file to which the gateway writes the contents of the interface event log buffer.
Step 6	**interface event-log max-buffer-size** *kbytes* **Example:** `Router(config-app-monitor)# interface event-log max-buffer-size 50`	(Optional) Sets the maximum size of the event log buffer for each application interface.
Step 7	**interface max-server-records** *number* **Example:** `Router(config-app-monitor)# interface max-server-records 50`	(Optional) Sets the maximum number of interface event-log records that are saved.
Step 8	**interface event-log error-only** **Example:** `Router(config-app-monitor)# interface event-log error-only`	(Optional) Limits interface event logging to error events only.
Step 9	**exit** **Example:** `Router(config-app-monitor)# exit`	Exits the application configuration monitor mode.

Modifying Event Log Settings for Call Legs

Perform this task to modify the default settings for event logs generated for voice call legs.

Restrictions

Event logs are available for telephony call legs only. Event logs for IP call legs are not supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call leg event-log dump ftp** *server*[**:***port*]**/***file* **username** *username* **password** [*encryption-type*] *password*
4. **call leg event-log max-buffer-size** *kbytes*
5. **call leg event-log error-only**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call leg event-log dump ftp** *server*[**:***port*]**/***file* **username** *username* **password** [*encryption-type*] *password* **Example:** `Router(config)# call leg event-log dump ftp ftp-server/elogs/leg_elogs.log username myname password 0 mypass`	(Optional) Specifies the location of the external file to which the voice gateway writes the contents of the event log buffer.
Step 4	**call leg event-log max-buffer-size** *kbytes* **Example:** `Router(config)# call leg event-log max-buffer-size 50`	(Optional) Sets the maximum size of the event log buffer for each call leg.
Step 5	**call leg event-log error-only** **Example:** `Router(config)# call leg event-log error-only`	(Optional) Enables transaction event logging for error events only.
Step 6	**exit** **Example:** `Router# exit`	Exits the current mode.

Modifying Event Log History Limits

Perform this task to modify the default settings for saving event logs to history.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **monitor**
5. **history session event-log save-exception-only**
6. **history session max-records** *number*

7. **history session retain-timer** *minutes*
8. **exit**
9. **exit**
10. **call leg history event-log save-exception-only**
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)#application`	Enters application configuration mode.

Step 4	`monitor`
	
	Example:
	`Router(config-app)# monitor`
		Enters application configuration monitor mode.

Step 5	**history session event-log save-exception-only** **Example:** `Router(config-app-monitor)# history session event-log save-exception-only`	(Optional) Saves event logs to history only for application sessions with exceptions or errors.
Step 6	**history session max-records** *number* **Example:** `Router(config-app-monitor)# history session max-records 50`	(Optional) Sets the maximum number of session records that are saved in the event-log history table.
Step 7	**history session retain-timer** *minutes* **Example:** `Router(config-app-monitor)# history session retain-timer 30`	(Optional) Sets the maximum number of minutes for which session history records are saved.
Step 8	**exit** **Example:** `Router(config-app-monitor)# exit`	Exits the application configuration monitor mode.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-app)# exit`	Exits the application configuration mode.
Step 10	**call leg history event-log save-exception-only** **Example:** `Router# call leg history event-log save-exception-only`	(Optional) Saves event logs only for call legs that have exceptions or errors.
Step 11	**exit** **Example:** `Router# exit`	Exits the current mode.

Configuration Examples for Monitoring Voice Applications

This section includes the following examples:

- Enabling Event Logs and Statistics Globally: Example, page 29
- Customizing Event Logs and Statistics Example, page 31

Enabling Event Logs and Statistics Globally: Example

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
hostname Router
!
!
resource-pool disable
tdm clock priority 1 1/0
spe default-firmware spe-firmware-1
aaa new-model
!
!
ip domain name domain.com
ip host speech-asr 10.10.10.111
ip host ftp-server 10.10.10.119
!
isdn switch-type primary-5ess
!
!
voice service voip
 fax protocol t38 ls-redundancy 7 hs-redundancy 3 fallback none
!
ivr asr-server rtsp://speech-asr/recognizer
ivr tts-server rtsp://speech-asr/synthesizer
!
fax receive called-subscriber $d$
```

```
fax send transmitting-subscriber 5550122
fax send left-header Date: $a$
fax send center-header $d$
fax send right-header $s$
fax send coverpage email-controllable
fax send coverpage comment Cover Page comment
fax interface-type fax-mail
mta send server 10.10.10.112 port 25
mta send subject sample subject
mta send postmaster postmaster@domain.com
mta send mail-from hostname Router.domain.com
mta send mail-from username user1
mta receive aliases [10.10.10.100]
mta receive maximum-recipients 1000
dial-control-mib retain-timer 10
dial-control-mib max-size 2
!
!
controller T1 1/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface FastEthernet0/0
 ip address 10.10.10.100 255.255.0.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 no cdp enable
!
interface FastEthernet0/1
 ip address 11.11.11.100 255.255.0.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
!
interface Serial1/0:23
 no ip address
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
!
interface Group-Async0
 no ip address
 no ip route-cache
 no ip mroute-cache
 no peer default ip address
 group-range 3/00 3/107
!
!
interface Dialer1
 no ip address
 no ip route-cache
 no ip mroute-cache
!
ip default-gateway 10.10.10.1
ip classless
ip route 10.0.0.0 255.0.0.0 10.10.10.1
ip route 11.0.0.0 255.0.0.0 11.11.11.1
no ip http server
!
```

```
snmp-server community password RW
snmp-server enable traps tty
!
call leg event-log
!
application
    service onramp tftp://demo/router/TCLware.2.0.1/app_libretto_onramp9.2.0.0.tcl
    !
    service offramp tftp://demo/router/TCLware.2.0.1/app_faxmail_offramp.2.0.1.1.tcl
    !
    service generic tftp://demo/scripts/master/generic.vxml
    !
    monitor
    interface stats
    interface event-log
    stats
    event-log
!
voice-port 1/0:D
!
dial-peer cor custom
!
dial-peer voice 1 pots
 service generic
 incoming called-number .
 direct-inward-dial
!
dial-peer voice 2 mmoip
 service fax_on_vfc_onramp_app out-bound
 destination-pattern .
 information-type fax
 session target mailto:$e$@[10.10.10.112]
!
line con 0
 exec-timeout 0 0
 logging synchronous
line aux 0
 logging synchronous
line vty 5 105
line 2/00 3/107
 no flush-at-activation
 modem InOut
!
scheduler allocate 10000 400
!
end
```

Customizing Event Logs and Statistics Example

```
.
.
.
call leg event-log errors-only
call leg event-log max-buffer-size 10
call leg event-log dump ftp ftp-server/leg_elogs.log username myname password mypass
call leg event-log
!
application
    service onramp tftp://demo/router/TCLware.2.0.1/app_libretto_onramp9.2.0.0.tcl
    !
    service offramp tftp://demo/router/TCLware.2.0.1/app_faxmail_offramp.2.0.1.1.tcl
    !
```

```
    service generic tftp://demo/scripts/master/generic.vxml
    !
    monitor
    interface stats
    interface event-log
    interface event-log error-only
    interface event-log max-buffer-size 50
    interface event-log dump ftp ftp-server/int_elogs.log username myname password mypass
    interface event-log ram disable
    interface max-server-records 20
    stats
    event-log
    event-log error-only
    event-log max-buffer-size 8
    event-log dump ftp ftp-server/app_elogs.log username myname password mypass
!
```

Additional References

- "" on page 1—Describes how to access Cisco Feature Navigator; also lists and describes, by Cisco IOS release, Tcl IVR and VoiceXML features for that release
- "Overview of Cisco IOS Tcl IVR and VoiceXML Applications" on page 1—Describes underlying Cisco IOS Tcl IVR and VoiceXML technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance