Configuring the Event Tracer

Last Updated: October 10, 2011

This document describes the Event Tracer feature. It includes the following sections:

Feature Overview

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, route processor switchover.

Note This feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been preprogrammed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

By default, trace messages saved to a file are saved in binary format without applying additional processing or formatting. Saving messages in binary format allows event tracing to collect informational messages faster and for a longer time prior to a system malfunction or processor switchover. Optionally, event trace messages can be saved in ASCII format for additional file processing.

The Event Tracer feature can support multiple traces simultaneously. To do this, the feature assigns a unique ID number to each instance of a trace. This way, all messages associated with a single instance of a

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

trace get the same ID number. Event tracing also applies a timestamp to each trace message, which aids in identifying the message sequence.

The number of trace messages stored in memory for each instance of a trace is configurable up to 65536 entries. As the number of trace messages stored in memory approaches the configured limit, the oldest entries are overwritten with new messages, which continues until the event trace is terminated.

Event tracing can be configured in “one-shot” mode. This is where the current contents of memory for a specified component are discarded and a new trace begins. New trace messages are collected until the message limit is reached, at which point the trace is automatically terminated.

- Benefits, page 2
- Restrictions, page 2

Benefits

Event tracing has a number of benefits to aid in system diagnosis:

Binary Data Format

Event information is saved in binary format without applying any formatting or processing of the information. This results in capturing event information more quickly and for a longer period of time in the moments leading up to a system malfunction or processor switchover. The ability to gather information quickly is also helpful in tracing events that generate a lot of data quickly.

File Storage

Information gathered by the event tracing can be written to a file where it can be saved for further analysis.

Optional ASCII Data Format

Event tracing provides an optional command to save the information in ASCII format.

Multiple Trace Capability

Event tracing can be configured to trace one or more components of the Cisco IOS software simultaneously, depending on the software version running on the networking device.

Restrictions

Event tracing provides a mechanism to help TAC representatives assist Cisco customers in diagnosing certain Cisco IOS software functions. Configuration of this feature on a networking device is recommended only under the direction of a TAC representative. This feature does not produce customer readable data; therefore, it requires the assistance of a TAC representative for proper configuration and analysis.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Supported Standards MIBs and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

The list of software components that support event tracing can vary from one Cisco IOS software image to another. And in many cases, depending on the software component, the event tracing functionality is enabled or disabled by default. Knowing what software components support event tracing and knowing the existing state of the component configuration is important in deciding whether to configure event tracing.

To determine whether event tracing has been enabled or disabled by default for a specific component, follow these steps:

SUMMARY STEPS

1. Use the **monitor event-trace ?**command in global configuration mode to get a list of software components that support event tracing.
2. Use the **show monitor event-trace** *component* **all** command to determine whether event tracing is enabled or disabled by default for the component.
3. Use the **show monitor event-trace** *component* **parameters** command to find out the default size of the trace message file for the component.

DETAILED STEPS

Step 1 Use the **monitor event-trace ?**command in global configuration mode to get a list of software components that support event tracing.

Example:

```
Router(config)# monitor event-trace ?
```

Step 2 Use the **show monitor event-trace** *component* **all** command to determine whether event tracing is enabled or disabled by default for the component.

Example:

```
Router# show monitor event-trace
```

```
component
 all
```

Step 3 Use the **show monitor event-trace** *component* **parameters** command to find out the default size of the trace message file for the component.

Example:

```
Router# show monitor event-trace

component
 parameters
```

This information can help you in determining your configuration options.

Prerequisites

The list of software components that support event tracing can vary from one Cisco IOS software image to another. And in many cases, depending on the software component, the event tracing functionality is enabled or disabled by default. Knowing what software components support event tracing and knowing the existing state of the component configuration is important in deciding whether to configure event tracing.

To determine whether event tracing has been enabled or disabled by default for a specific component, follow these steps:

- **Step 1 -** Use the **monitor event-trace ?** command in global configuration mode to get a list of software components that support event tracing.

  ```
  Router(config)# monitor event-trace ?
  ```

- **Step 2 -** Use the **show monitor event-trace***component***all** command to determine whether event tracing is enabled or disabled by default for the component.

  ```
  Router# show
  monitor event-trace component all
  ```

- **Step 3 -** Use the **show monitor event-trace***component***parameters** command to find out the default size of the trace message file for the component.

  ```
  Router#
  show
  monitor event-trace
  component
  parameters
  ```

This information can help you in determining your configuration options.

Configuration Tasks

Follow the instructions in the "Configuration Tasks, page 4" section prior to configuring this feature. If the default configuration information meets your site requirements, no further configuration may be necessary, and you may proceed to the section "Verifying Event Trace Operation, page 6."

Configuring Event Tracing

In most cases where Cisco IOS software components support event tracing, the feature is configured by default. For some software components, event tracing is enabled, while for other components event tracing might be disabled. In some cases, a TAC representative may want to change the default settings.

To enable or disable event tracing, use the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **monitor event-trace** *component* **enable** or `Router(config)#` **monitor event-trace** *component* **disable**	Enables or disables event tracing for the specified Cisco IOS software component on the networking device. **Note** Component names are set in the system software and are not configurable. To obtain a list of software components supporting event tracing for this release, use the **monitor event-trace**command.

Configuring the Event Trace Size

In most cases where Cisco IOS software components support event tracing, the feature is configured by default. In some cases, such as directed by a TAC representative, you might need to change the size parameter to allow for writing more or fewer trace messages to memory.

To configure the message size parameter, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **monitor event-trace** *component* **size** *number*	Configures the size of the trace for the specified component. The number of messages that can be stored in memory for each instance of a trace is configurable up to 65536 entries.

Configuring the Event Trace Message File

To configure the file location where you want to save trace messages, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **monitor event-trace** *component* **dump-file** *filename*	Configures the file where the trace messages will be saved. The maximum length of the filename (path:filename) is 100 characters. The path can point to flash memory on the networking device or to a TFTP or FTP server.

Verifying Event Trace Operation

Depending on the software component, event tracing is enabled or disabled by default. In either case, the default condition will not be reflected in the output of the **show running-config** command; however, changing any of the settings for a command that has been enable or disabled by default will cause those changes to show up in the output of the **show running-config** command.

SUMMARY STEPS

1. If you made changes to the event tracing configuration, enter the **show running-config** command in privileged EXEC mode to verify the changes.
2. Enter the **show monitor event-trace** *component* command to verify that event tracing has been enabled or disabled for a component.
3. Verify that you have properly configured the filename for writing trace messages.

DETAILED STEPS

Step 1 If you made changes to the event tracing configuration, enter the **show running-config** command in privileged EXEC mode to verify the changes.

Example:

```
Router# show running-config
```

Step 2 Enter the **show monitor event-trace** *component* command to verify that event tracing has been enabled or disabled for a component.
In the following example, event tracing has been enabled for the IPC component. Notice that each trace message is numbered sequentially (for example, 3667) and is followed by a the timestamp (derived from the device uptime). Following the timestamp is the component specific message data.

Example:

```
Router# show monitor event-trace ipc

3667:  6840.016:Message type:3 Data=0123456789
3668:  6840.016:Message type:4 Data=0123456789
3669:  6841.016:Message type:5 Data=0123456789
3670:  6841.016:Message type:6 Data=0123456
```

To view trace information for all components enabled for event tracing, enter the **show monitor event-trace all-traces** command. In this example, separate output is provided for each event and message numbers are interleaved between the events.

Example:

```
Router# show monitor event-trace all-traces
Test1 event trace:
3667: 6840.016:Message type:3 Data=0123456789
3669: 6841.016:Message type:4 Data=0123456789
3671: 6842.016:Message type:5 Data=0123456789
3673: 6843.016:Message type:6 Data=0123456789
Test2 event trace:
3668: 6840.016:Message type:3 Data=0123456789
```

```
3670: 6841.016:Message type:4 Data=0123456789
3672: 6842.016:Message type:5 Data=0123456789
3674: 6843.016:Message type:6 Data=0123456789
```

Step 3 Verify that you have properly configured the filename for writing trace messages.

Example:

```
Router# monitor event-trace ipc dump
```

Troubleshooting Tips

Event Tracing Does Not Appear to Be Configured in the Running Configuration

Depending on the software component, event tracing is enabled or disabled by default. In either case, the default condition will not be reflected in output of the **show running-config**command; however, changing any of the settings for a command that has been enabled or disabled by default will cause those changes to show up in the output of the **show running-config** command. Changing the condition of the component back to its default state (enabled or disabled), will cause the entry not to appear in the configuration file.

Show Command Output Is Reporting "One or More Entries Lost "

The trace function is not locked while information is being displayed to the console, which means that new trace messages can accumulate in memory. If entries accumulate faster than they can be displayed, some messages can be lost; however, messages will continue to display on the console. If the number of lost messages is excessive, the **show** command will stop displaying messages.

Show Command Output Terminates Unexpectedly

The trace function is not locked while information is being displayed to the console, which means that new trace messages can accumulate in memory. If entries accumulate faster than they can be displayed, some messages can be lost. If the number of lost messages is excessive, the **show** command will stop displaying messages.

Show Command Output Is Reporting That "Tracing Currently Disabled, from EXEC Command"

The Cisco IOS software allows for the subsystem components to define whether support for event tracing is enabled or disabled at boot time. Event tracing allows users to enable or disable event tracing in two ways: using the **monitor event-trace**(EXEC) command in privileged EXEC mode or using the **monitor event-trace**(global) command in global configuration mode. To enable event tracing again in this case, you would enter the **enable** form of either of these commands.

Show Command Output Is Reporting That "Tracing Currently Disabled, from Config Mode"

The Cisco IOS software allows for the subsystem components to define whether support for event tracing is enabled or disabled at boot time. Event tracing allows users to disable event tracing in two ways: using the **monitor event-trace disable** (EXEC) command in privileged EXEC mode or using the **monitor event-trace disable** (global) command in global configuration mode. To enable event tracing again in this case, you would enter the **enable** form of either of these commands.

Event Trace Messages Are Not Being Saved in ASCII Format

By default, the **monitor event-trace** *component* **dump** and **monitor event-trace dump-traces** commands save trace messages in binary format. If you want to save trace messages in ASCII format, use either the **monitor event-trace** *component* **dump pretty**command to write the trace messages for a single event, or the **monitor event-trace dump-traces pretty** command to write trace messages for all event traces currently enabled on the networking device.

Configuration Examples

Configuring Event Tracing for One Component Example

In the following example, the networking device has been configured to trace IPC component events:

```
monitor event-trace ipc enable
```

Configuring Event Tracing for Multiple Components Example

In the following example, the networking device has been configured to trace IPC and MBUS component events:

```
monitor event-trace ipc enable
monitor event-trace mbus enable
```

Configuring the Event Trace Size Example

In the following example, the size of the IPC trace is set to 4096 entries while the size of the MBUS trace is set to 8192 entries:

```
monitor event-trace ipc size 4096
monitor event-trace mbus size 8192
```

Configuring the Event Trace Message File Example

The following example identifies the files in which to write trace messages. In this example, event tracing has been enabled for both the IPC and MBUS components, the IPC trace messages are written to the ipcdump file in flash memory, while the MBUS trace message files are written to the mbusdump file on the TFTP server.

```
monitor event-trace ipc dump-file slot0:ipcdump
monitor event-trace mbus dump-file TFTP:mbusdump
```

Feature Information for Event Tracer

Table 1 Feature Information for Event Tracer

Feature Name	Releases	Feature Information
Event Tracer	12.0(18)S 12.2(8)T	The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software.. The following commands were introduced or modified: **monitor event-trace (EXEC), monitor event-trace (global), monitor event-trace dump-traces, show monitor event-trace**.