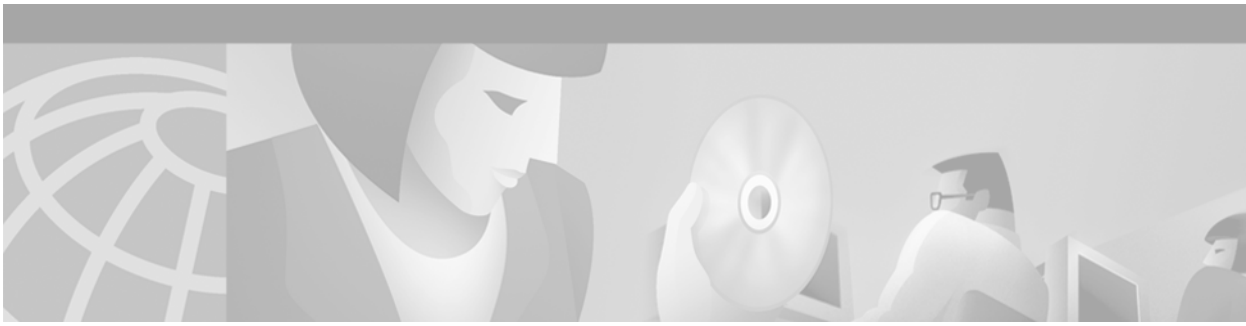

Event Tracer

Feature History

Release	Modification
12.0(18)S	This feature was introduced.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.

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 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

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.

Supported Platforms

- Cisco 12000 Internet router

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, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If successful, account details with a new random password will be e-mailed to you. 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

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

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

See the following sections for configuration tasks for the Event Tracer feature. Each task in the list is identified as either required or optional.

- Configuring Event Tracing (Optional)
- Configuring the Event Trace Size (Optional)
- Configuring the Event Trace Message File (Optional)
- Verifying Event Trace Operation (Optional)

Follow the instructions in the "Prerequisites" 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."

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

Note 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.

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.

```
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.

```
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.

```
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.

```
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

This section provides the following configuration examples:

- Configuring Event Tracing for One Component Example
- Configuring Event Tracing for Multiple Components Example
- Configuring the Event Trace Size Example
- Configuring the Event Trace Message File Example

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
```

Command Reference

This section documents new commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

New Commands

- **monitor event-trace (EXEC)**
- **monitor event-trace (global)**
- **monitor event-trace dump-traces**
- **show monitor event-trace**

monitor event-trace (EXEC)

To control the event trace function for a specified Cisco IOS software subsystem component, use the **monitor event-trace** (EXEC) command in privileged EXEC mode.

monitor event-trace *component* {**clear** | **disable** | **dump** [**pretty**] | **enable** | **one-shot**}

Syntax Description

component	Name of the Cisco IOS software subsystem component that is the subject of the event trace. To get a list of components that support event tracing in this release, use the **monitor event-trace ?** command.
clear	Clears existing trace messages for the specified component from memory on the networking device.
disable	Turns off event tracing for the specified component.
dump	Writes the event trace results to the file configured using the **monitor event-trace** (global) command. The trace messages are saved in binary format.
pretty	(Optional) Saves the event trace message in ASCII format.
enable	Turns on event tracing for the specified component.
one-shot	Clears any existing trace information from memory, starts event tracing again, and disables the trace when the trace reaches the size specified using the **monitor event-trace** (global) command.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(18)S	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use the **monitor event-trace** (EXEC) command to control what, when, and how event trace data is collected. Use this command after you have configured the event trace functionality on the networking device using the **monitor event-trace** (global) command.

Note The amount of data collected from the trace depends on the trace message size configured using the **monitor event-trace** (global) command for each instance of a trace.

The Cisco IOS software allows for the subsystem components to define whether support for event tracing is enabled or disabled at boot time. You can 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, you would enter the **enable** form of either of these commands.

To determine whether a subsystem has enabled or disabled event tracing, use the **monitor event-trace ?** command to get a list of software components that support event tracing. To determine whether event tracing is enabled by default for the subsystem, use the **show monitor event-trace** command to view trace messages.

Use the **show monitor event-trace** command to display trace messages.

Use the **monitor event-trace** *component* **dump** command to save trace message information for a single event. By default, trace information is saved in binary format. If you want to save trace messages in ASCII format, possibly for additional application processing, use the **monitor event-trace** *component* **dump pretty** command.

To write the trace messages for all events currently enabled on a networking device to a file, enter the **monitor event-trace dump-traces** command.

To configure the file where you want to save trace information, use the **monitor event-trace** (global) command.

Examples

The following example shows the privileged EXEC commands to stop event tracing, clear the current contents of memory, and re-enable the trace function for the IPC component. This example assumes that the tracing function is configured and enabled on the networking device.

```
Router# monitor event-trace ipc disable
Router# monitor event-trace ipc clear
Router# monitor event-trace ipc enable
```

The following example shows how the **monitor event-trace one-shot** command accomplishes the same function as the previous example except in one command. In this example, once the size of the trace message file has been exceeded, the trace is terminated.

```
Router# monitor event-trace ipc one-shot
```

The following example shows the command for writing trace messages for an event in binary format. In this example, the trace messages for the IPC component are written to a file.

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

The following example shows the command for writing trace messages for an event in ASCII format. In this example, the trace messages for the MBUS component are written to a file.

```
Router# monitor event-trace mbus dump pretty
```

Related Commands

Command	Description
monitor event-trace (global)	Configures event tracing for a specified Cisco IOS software subsystem component.
monitor event-trace dump-traces	Saves trace messages for all event traces currently enabled on the networking device.
show monitor event-trace	Displays event trace messages for Cisco IOS software subsystem components.

monitor event-trace (global)

To configure event tracing for a specified Cisco IOS software subsystem component, use the **monitor event-trace** (global) command in global configuration mode. To change the default setting to enable or disable event tracing, refer to the "Usage Guidelines" section for this command.

monitor event-trace *component* {**disable** | **dump-file** *filename* | **enable** | **size** *number*}

Syntax Description

component	Name of the Cisco IOS software subsystem component that is the object of the event trace. To get a list of components that support event tracing in this release, use the **monitor event-trace ?** command.
disable	Turns off event tracing for the specified component.
dump-file	Specifies the file where event trace messages are written from memory on the networking device.
filename	Name of the file for writing trace messages. The maximum length of the filename (path and filename) is 100 characters and path can point to flash memory on the networking device or to a TFTP or FTP server.
enable	Turns on event tracing for the specified component provided that the component has been configured using the **monitor event-trace** (global) command.
size	Sets the number of messages that can be written to memory for a single instance of a trace. **Note** Some Cisco IOS software subsystem components set the size by default. To view the size parameter, use the **show monitor event-trace** *component* **parameters** command. When the number of event trace messages in memory exceeds the size, new messages will begin to overwrite the older messages in the file.
number	Number of messages for each instance of a trace. Specify a number in the range 1 to 65536.

Defaults

Enabled or disabled depending on the software component.

Command Modes

Global configuration

Command History

Release	Modification
12.0(18)S	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use the **monitor event-trace** (global) command to enable or disable event tracing and to configure event trace parameters for Cisco IOS software subsystem components.

Note Event tracing is intended for use as a software diagnostic tool and should be configured only under the direction of a TAC representative. In Cisco IOS software images that do not provide subsystem support for the event trace function, the **monitor event-trace** (global) command is not available.

The Cisco IOS software allows the subsystem components to define whether support for event tracing is enabled or disabled by default. The command interface for event tracing allows users to change the default 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.

Additionally, default settings do not show up in the configuration file. If the subsystem software enables event tracing by default, the **monitor event-trace** *component* **enable** command will not show up in the configuration file of the networking device; however, disabling event tracing that has been enabled by default by the subsystem will create a line in the configuration file.

Note The amount of data collected from the trace depends on the trace message size configured using the **monitor event-trace** (global) command for each instance of a trace.

To determine whether a subsystem has enabled or disabled event tracing, use the **monitor event-trace ?** command to get a list of software components that support event tracing.

To determine whether event tracing is enabled by default for the subsystem, use the **show monitor event-trace** command to view trace messages.

Examples

The following example shows how to enable event tracing for the IPC subsystem component in Cisco IOS software and configure the size to 4096 messages. The trace messages file is set to *ipc-dump* in slot0 (flash memory).

```
monitor event-trace ipc enable
monitor event-trace ipc dump-file slot0:ipc-dump
monitor event-trace ipc size 4096
```

Related Commands

Command	Description
monitor event-trace (EXEC)	Controls event trace function for a specified Cisco IOS software subsystem component.
monitor event-trace dump-traces	Saves trace messages for all event traces currently enabled on the networking device.
show monitor event-trace	Displays event trace messages for Cisco IOS software subsystem components.

monitor event-trace dump-traces

To save trace messages for all event traces currently enabled on the networking device, use the **monitor event-trace dump-traces** command in privileged EXEC mode.

monitor event-trace dump-traces [**pretty**]

Syntax Description

pretty	(Optional) Saves the event trace message in ASCII format.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(18)S	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use the **monitor event-trace dump-traces** command to save trace message information for all event traces currently enabled on a networking device. By default, trace information is saved in binary format. If you want to save trace messages in ASCII format, possibly for additional application processing, use the **monitor event-trace dump-traces pretty** command.

To write the trace messages for an individual trace event to a file, enter the **monitor event-trace** (EXEC) command.

To configure the file where you want to save messages, use the **monitor event-trace** (global) command.

Examples

The following example shows how to save the trace messages in binary format for all event traces enabled on the networking device.

```
monitor event-trace dump-traces
```

The following example shows how to save the trace messages in ASCII format for all event traces enabled on the networking device.

```
monitor event-trace dump-traces pretty
```

Related Commands

Command	Description
monitor event-trace (EXEC)	Controls event trace function for a specified Cisco IOS software subsystem component.
monitor event-trace (global)	Configures event tracing for a specified Cisco IOS software subsystem component.
show monitor event-trace	Displays event trace messages for Cisco IOS software subsystem components.

show monitor event-trace

To display event trace messages for Cisco IOS software subsystem components, use the **show monitor event-trace** command in privileged EXEC mode.

show monitor event-trace [**all-traces**] [*component* {**all** | **back** *time* | **clock** *time* | **from-boot** *seconds* | **latest** | **parameters**}]

Syntax Description

all-traces	(Optional) Displays all event trace message in memory to the console.
component	(Optional) Name of the Cisco IOS software subsystem component that is the object of the event trace. To get a list of components that support event tracing in this release, use the **monitor event-trace ?** command.
all	Displays all event trace messages currently in memory for the specified component.
back	Specifies how far back from the current time you want to view messages. For example, you can gather messages from the last 30 minutes.
time	Length of time in hours and minutes format (hh:mm).
clock	Displays event trace messages starting from a specific clock time.
time	Time from which to display messages in hours and minutes format (hh:mm).
from-boot	Displays event trace messages starting from a specified number of seconds after booting.
seconds	Number of seconds since the networking device was last booted (uptime). To view the uptime, in seconds, enter the **show monitor event-trace** *component* **from-boot ?** command.
latest	Displays only the event trace messages since the last **show monitor event-trace** command was entered.
parameters	Displays the trace parameters. Currently, the only parameter displayed is the size (number of trace messages) of the trace file.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(18)S	This command was introduced.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.

Usage Guidelines

Use the **show monitor event-trace** command to display trace message information.

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 this happens, the **show monitor event-trace** command will generate a

message indicating that some messages might be lost; however, messages will continue to display on the console. If the number of lost messages is excessive, the **show monitor event-trace** command will stop displaying messages.

Examples

The following sample output illustrates the **show monitor event-trace** *component* command output for the IPC component. Notice that each trace message is numbered and is followed by a the timestamp (derived from the device uptime). Following the timestamp is the component-specific message data.

```
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 configured for event tracing on the networking device, 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.

```
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
```

Related Commands

Command	Description
monitor event-trace (EXEC)	Controls event trace functions for a specified Cisco IOS software subsystem component.
monitor event-trace (global)	Configures event tracing for a specified Cisco IOS software subsystem component.
monitor event-trace dump-traces	Saves trace messages for all event traces currently enabled on the networking device.