



Event Router

Version 2.4.

8 August 2022

User Guide



Copyright © 2022 Axway

All rights reserved.

This documentation describes the following Axway software:

Axway Event Router 2.4.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, Axway.

This document, provided for informational purposes only, may be subject to significant modification. The descriptions and information in this document may not necessarily accurately represent or reflect the current or planned functions of this product. Axway may change this publication, the product described herein, or both. These changes will be incorporated in new versions of this document. Axway does not warrant that this document is error free.

Axway recognizes the rights of the holders of all trademarks used in its publications.

The documentation may provide hyperlinks to third-party web sites or access to third-party content. Links and access to these sites are provided for your convenience only. Axway does not control, endorse or guarantee content found in such sites. Axway is not responsible for any content, associated links, resources or services associated with a third-party site.

Axway shall not be liable for any loss or damage of any sort associated with your use of third-party content.

Contents

| | |
|---|-----------|
| About this guide | 6 |
| Who should read this guide | 6 |
| Related documentation | 6 |
| Support services | 6 |
| Training services | 6 |
| 1 About Event Router | 7 |
| Event Router in DMZ | 8 |
| About Processes and entities | 9 |
| About overflow and batch files | 11 |
| On z/OS platforms | 11 |
| 2 Global steps in Event Router configuration | 12 |
| About setting the Target Parameters | 12 |
| 3 General Parameters | 14 |
| After modifying General Parameter File parameters | 14 |
| Special display issues for z/OS | 14 |
| General Parameter File elements | 14 |
| [Agent] section | 15 |
| [System] section | 16 |
| [TcpSource] section | 17 |
| IPv6 configuration parameters | 18 |
| [DefaultTarget] section | 19 |
| [Sentinel] section | 21 |
| [MQSeries] section | 22 |
| General Parameters example | 23 |
| 4 Target Parameters | 25 |
| Target Parameters File elements | 25 |
| <TrkXml> section | 26 |
| <EventRouter> section | 26 |
| <DefaultTarget> | 26 |
| <Target> section | 30 |
| <Route> section | 37 |
| <Authorization> section | 40 |
| Target Parameters example | 41 |

| | |
|--|-----------|
| 5 Security profile | 44 |
| Concepts | 44 |
| Profile parameters | 45 |
| Parameters for SSL session in server mode | 45 |
| Parameters for SSL session in client mode | 47 |
| Use cases | 48 |
| 6 Using the command syntax | 50 |
| UNIX, OS/400 and Windows commands | 50 |
| Setting the environment variables | 50 |
| Entering a single command | 50 |
| Entering a series of commands | 50 |
| z/OS command | 51 |
| Command HELP | 51 |
| 7 Starting and stopping the Event Router | 52 |
| UNIX, Windows and OS/400 | 52 |
| z/OS | 52 |
| Return codes: AGTRUN JCL | 53 |
| 8 Managing entities | 55 |
| 9 Managing targets | 56 |
| Target parameter | 58 |
| 10 Managing traces | 59 |
| 11 Log messages | 61 |
| Log messages - basics | 61 |
| Where to find log messages | 61 |
| How to interpret log messages | 61 |
| How to set log messages | 62 |
| Log messages - details | 63 |
| Connecting to networks | 64 |
| Initializing, enabling, and disabling Event Router targets | 70 |
| Initializing and stopping Event Router entities | 77 |
| Connecting to Event Router targets | 80 |
| Sending messages | 85 |
| Receiving messages | 90 |
| WebSphere MQ (formerly MQSeries) Interface messages | 95 |
| Executing commands | 99 |
| Detailing errors | 102 |

| | |
|---|------------|
| 12 Error traces | 106 |
| 13 Cipher suites | 107 |
| Appendix A: AGTCRYPT utility | 109 |
| Generate the encryption key | 109 |
| Encrypt the certificate password | 110 |
| Decrypt an encrypted password | 110 |
| Renew the key (future use) | 111 |
| Modify the Event Router configuration files | 111 |
| z/OS Installation | 112 |
| Example | 112 |

About this guide

This guide describes configuring and using Axway Event Router.

Who should read this guide

This guide is intended for people who administrate and use Event Router. This guide presumes you have knowledge of:

- Your company's business processes and practices
- Your company's hardware, software and IT policies
- Axway Sentinel and the Event Router component

Related documentation

Axway Sentinel Event Router is accompanied by a complete set of documentation, covering all aspects of using the component. These documents include the following:

- *Sentinel Event Router Installation Guide*
- *Sentinel Event Router Release Notes*
- *Sentinel Event Router User Guide*

Support services

The Axway Global Support team provides worldwide 24 x 7 support for customers with active support agreements.

Email support@axway.com or visit <https://support.axway.com>.

Training services

Axway offers training across the globe, including on-site instructor-led classes and self-paced online learning.

For details, go to: <http://www.axway.com/support-services/training>

About Event Router

1

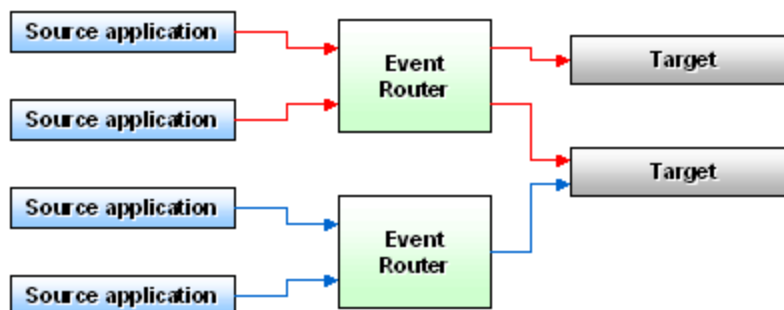
The Event Router collects messages from one or more tracked applications, and then routes them to one or more target servers or to Websphere MQ applications. In this book:

- Applications that create and send event messages using the Sentinel Tracking API are referred to as **source applications**
- Target servers and target Websphere MQ applications are referred to as **targets**

The routing services provided by the Event Router benefit your Sentinel architecture. Via these services, you can:

| | |
|---|--|
| Send the same message to more than one target | Without the Event Router, tracked applications can only send messages to a single Sentinel Server. |
| Reduce the number of input connections that targets must maintain | Rather than maintaining one input connection for each source application, Event Router targets maintain only one input connection for each Event Router. |
| Place your Event Router in a secured architecture | The current version of the Event Router includes features that enable you to configure it to perform securely from the interior of a DMZ. |
| Integrate the queuing services of WebSphere MQ in the Sentinel architecture | The Event Router can pass messages through a WebSphere MQ queue manager. |

The following is an example of one possible Event Router scenario. As this example illustrates, each target maintains only one input connection for each Event Router:



Event Router in DMZ

When you place one or more Sentinel monitored applications in a DMZ, and set DMZ rules to block the initialization of connections to elements situated outside the DMZ, you require a particular Event Router configuration to insure data flow to Event Router targets. This DMZ configuration requires at least:

- One **serving Event Router** (situated in the DMZ). A serving Event Router is an Event Router that you configure to operate in QLTSRV mode.

To configure a *serving* Event Router:

- In the General Parameters File, set the *sap* attribute in the Sentinel section to the listen port for connections with a *requesting* Event Router.

Note When you define the *requesting* Event Router, you enter this same listen port number in the Target/Access element of the Target Parameters file for that Event Router.

- In the Target Parameters File, set the *mode* attribute in the Target/Access element to QLTTREQ or QLTSRV.

The following is an example of Target Parameters File values that define a *serving* Event Router:

```
<Target name="SRV" defaultXntf="yes" defaultXml="yes">
<Access mode="QLTSRV" />
```

Where: the target name is the same name you enter in the "ident" parameter of the requesting Event Router (see following example).

- One **requesting Event Router** (situated outside of the DMZ and physically connected to the serving Event Router). A requesting Event Router is an Event Router that you configure to operate in QLTTREQ mode and for which you enter the identity of the related serving Event Router and sending criteria.

To configure a *requesting* Event Router, in the Target Parameters File:

- In the Target/Access element, set the *mode* attribute element to QLTTREQ. In the port attribute, enter the listen port that you defined in the *sap* attribute of the General Parameters file for the *serving* Event Router.
- In the same Target/Access element, enter the name of the targeted *serving* Event Router in the *ident* attribute.
- In corresponding the Target/Sendif element, set the parameters to define the frequency at which the Event Router connects to the serving Event Router.

The following is an example of Target Parameters File values that define a *requesting* Event Router:

```
<Target name="REQ" defaultXntf="yes" defaultXml="yes">
<Access mode="QLTTREQ" port="1505" addr="hpx3.pa.axway" ident="SRV"/>
<Sendif period ="20"/>
```

Where:

`port` = the listen port number. This is the listen port that you defined in the `sap` attribute of the General Parameters file for the *serving* Event Router.

`ident` = the name of the target *serving* Event Router you defined in the serving Event Router Target Parameters File (see previous example).

Additionally, if your Event Router configuration includes one or more proxy servers, or if you want to define an originating port other than the default port, in the General Parameters File of the serving Event Router, set the TCPSource parameters:

- `tcp_org_port_file`
- `tcp_proxi_file`

About Processes and entities

Functionally, the Event Router comprises a set of physical Event Router processes. These are the units of binary executable code that provide Event Router services.

Each physical Process supports one or more logical processes. In this documentation, Event Router logical processes are referred to as **Entities**. Entities provide a logical structure that facilitates the management of Event Router functions. Via Entities, the Event Router provides you with information about the operational events that occur on your system.

The following table introduces the role of each Event Router entity.

| Entity | Function |
|--------|--|
| DISP | Dispatches incoming messages to the appropriate Event Router entity. To perform this task, DISP: <ol style="list-style-type: none">1. Loads the Target Parameters File.2. Receives messages from one or more source applications.3. Identifies the relevant targets.4. Dispatches the messages to the Processes (SNTL, MQOUT) that then route them to the relevant targets. |
| LOG | Records traces in the log file |

| Entity | Function |
|-----------|--|
| ZLGR | <p>z/OS ONLY. Transfers messages from Logger file. To perform this task, this entity does the following:</p> <ol style="list-style-type: none"> 1. Opens the Logger file. 2. Periodically scans the file for new messages. 3. Sends the messages to DISP. 4. If DISP: <ul style="list-style-type: none"> • Can dispatch the message, ZLGR marks the message for deletion and increment its process messages counter. • Indicates the message is invalid, ZLGR marks the message for deletion and does not increment its process messages counter. • Cannot dispatch the message, ZLGR wait for the next scan to retry sending the message. |
| MQIN | <p>Transfers messages from WebSphere MQ to an Event Router. To perform this task, this entity does the following:</p> <ol style="list-style-type: none"> 1. MQIN retrieves messages from a WebSphere queue. 2. MQIN sends the retrieved messages to DISP. 3. If DISP: <ul style="list-style-type: none"> • Can dispatch the messages, MQIN deletes the messages from the queue in WebSphere MQ • Cannot dispatch the messages, MQIN routes the messages to a retry queue in WebSphere MQ |
| MQOUT | <p>Routes messages to WebSphere MQ. To perform this task, MQOUT:</p> <ol style="list-style-type: none"> 1. Receives the relevant target configuration from DISP. 2. Receives messages from DISP. 3. Stores received messages in an overflow file or a batch file when appropriate. 4. Inserts messages in the relevant WebSphere queues. |
| NET, NETS | <p>Manages the TCP/IP connections to source applications and targets. On some platforms, two entities NET and NETS are defined, one for the source application and one for the targets.</p> |

| Entity | Function |
|--------|---|
| SNTL | <p>Routes messages to one or more:</p> <ul style="list-style-type: none">• Sentinel Servers• Scope Servers <p>To perform this task, the SNTL entity:</p> <ol style="list-style-type: none">1. Receives the relevant target configuration from DISP.2. Receives messages from DISP.3. Stores received messages in an overflow file or a batch file when appropriate.4. Sends messages to the relevant targets. |

About overflow and batch files

When you configure the Event Router, you define the following files for each target:

- Overflow - This file stores only the messages that the Event Router cannot send to the relevant target.
- Batch - This file stores messages that the Event Router will send in batches to the relevant target. You specify the conditions under which the Event Router sends batches of messages from this file.

On z/OS platforms

The Event Router can run on z/OS platforms. On these platforms, MQOUT and SNTL entities are each associated with a separate file that combines the features of both the overflow file and the batch file. Each file is referred to as an overflow/batch file. When you configure targets on a z/OS platform, you can only identify the overflow/batch files that you created during installation. For more information, refer to the Sentinel installation documentation.

Global steps in Event Router configuration

2

To configure the Event Router:

1. During the installation process, assign values to the General Parameters of the Event Router. The Event Router stores your settings in the General Parameters file. This file also includes default values for the General Parameters that you do not set during the installation process. You can modify the contents of this file at any time.
2. After the installation, assign values to the Target Parameters of the Event Router. The Target Parameters identify where and how the Event Router sends messages. The Event Router stores these parameters in the Target Parameters file. You can modify the contents of this file at any time.
3. After the installation and if SSL sessions are required, assign values to the Security Profiles of the Event Router. The Security Profiles identify the SSL information, such as the certificate that the Event Router needs to establish a SSL session. The profiles are saved in the Security Profile file. You can modify the contents of this file at any time.

About setting the Target Parameters

When you configure the Event Router, you set the:

- Target Parameters for specific Event Router targets via the <Target> element of the Target Parameters file.
- Default Target Parameters for all Event Router targets via the:
 - <DefaultTarget> element of the Target Parameters file
 - [DefaultTarget] section of the General Parameters file

The default values in the Target Parameters file take priority over the default values that are defined in the General Parameters file, as described in the following example:

Example: In the Target Parameters file, you use the <Target> element to a target. You assign the name TargetOne to this target. Later, the Event Router receives a message for TargetOne and performs the following tasks:

1. In the Target Parameters file, the Event Router searches for parameter values in the <Target> element that you defined for TargetOne. However, in the <Target> element, you did not specify the address of TargetOne.
2. In the Target Parameters file, the Event Router searches the <DefaultTarget> element for the default address for targets. However, in the <DefaultTarget> element, you did not specify a default address.

3. In the General Parameters file, the Event Router searches the [DefaultTarget] section for the default address for targets. Because the [DefaultTarget] section contains a default address, the Event Router routes the message to this address.

When you install the Event Router, you assign values to most General Parameters of the Event Router. Sentinel stores your settings in the General Parameters File, `Trkagent.ini`. Use this file to view and to modify the values of the General Parameters that you:

- Set during the installation process
- Did not set during the installation process

Note The `trkagent.ini` file is found in the `conf` sub-directory of the product. On z/OS, the equivalents are:

- Conf sub-directory is the PDS CNTL from the Target Environment.
 - The member USEPARIN is the `Trkagent.ini` file
- The samples sub-directory is the PDS SAMPLE from the Installation Environment.

After modifying General Parameter File parameters

If you modify values in the General Parameters File, remember to execute the setup program for the Event Router. For more information, refer to the Sentinel Event Router installation documentation specific to the platform.

Special display issues for z/OS

Depending on the code page associated to the terminal, the visualization of the bracket characters may pose a problem. To avoid this problem, the Event Router on z/OS accepts the use of parentheses in place of bracket.

General Parameter File elements

In the General Parameters File, parameters are grouped in the following sections:

| Section | Contains the parameters that |
|-----------------|--|
| [Agent] | Identify the following: <ul style="list-style-type: none"> • Event Router • Target Parameters File • Logging Level of the DISP entity (the dispatcher for an Event Router entity) • Maximum size of incoming messages |
| [System] | Manage SVC communications (currently for z/OS only) |
| [TcpSource] | Manage the transfer of messages from source applications to the Event Router |
| [DefaultTarget] | Set default values for Target Parameters |
| [Sentinel] | Activate the SNTL entity and identify the Logging Level of this entity (the router for Sentinel and Scope Servers) |
| [MQSeries] | Implement the exchange of messages with WebSphere MQ, as follows: <ul style="list-style-type: none"> • Activate the MQIN and MQOUT entities (the message carriers for WebSphere MQ) • Identify WebSphere queues • Identify the Logging Level of the MQIN and MQOUT entities |

[Agent] section

In the [Agent] section, assign values to the parameters that identify the:

- Event Router
- Target Parameters File
- Logging Level of the DISP entity (the dispatcher for Event Router entities)
- Maximum size of incoming messages

The following table introduces the parameters that are in the [Agent] section and describes their values.

| Parameter | Value |
|-----------|---|
| name | The name of the Event Router. The maximum length of this name is 50 characters. The name, address, and listen port are information added by the Event Router in the processed messages (except if relay mode is set). Use a unique name to distinguish the Event Router information on the Sentinel Server. |

| Parameter | Value |
|------------------------|---|
| target_parameters_file | The path to the Target Parameters File (target.xml). During installation, the sample from the install directory is copied in the conf directory where it can be customized. The sample accepts all messages and sends them to the server address set in the [DefaultTarget] section. |
| security_profile_file | The path to the security profile file. A sample (sslconf.ini) is delivered in the install directory. During a new installation, the sample is copied in the conf directory and if SSL is activated, it MUST be customized with the Certificates definitions. |
| encrypt_key_file | Path to the file containing the key used to encrypt data (see appendix A for the description how to create the key file). This parameter is only required if PKCS12 certificate password is encrypted. |
| log | The Logging Level of the DISP entity. By default, the value of this parameter is 0 (zero). |
| message_size | The maximum number of bytes that the Event Router can receive in one message. By default, the value of this parameter is 4000 . |
| api_file | <p>z/OS only.</p> <p>When the Event Router runs on an z/OS platform, Tracked Applications can share an overflow file with the Event Router. In this case, one of the following is possible:</p> <ul style="list-style-type: none"> • A single tracked application transfers messages from the overflow file to the Event Router • The Event Router retrieves messages from the overflow file <p>If you want the Event Router to retrieve messages from a shared overflow file, specify the complete path to this file.</p> |
| api_timer | <p>z/OS only.</p> <p>How often the Event Router checks the shared overflow/batch file for messages, expressed in seconds.</p> |
| queue | <p>z/OS only.</p> <p>On z/OS, in addition to TCPIP, applications can exchange messages using XCF. The name of the XCF member that the Event Router uses to communicate with tracked applications. If you are NOT using XCF to communicate with the Sentinel application, set the value to NOQ. If at least one application is using XCF, put the value corresponding to the TRKQUEUE parameter in the Sentinel application configuration file. When the interface for your tracked applications is not XCF, do not assign a value to this parameter.</p> |

[System] section

Currently for z/OS only. In the [System] section, assign a value to the parameter that identifies the Switched Virtual Circuit handled in the z/OS environment.

The following table introduces the unique parameter of the [System] section and describes its value.

| Parameter | Value |
|-----------|--|
| svc_nb | SVC (Switched Virtual Circuit) communication number. This number corresponds to the TRKSVC parameter in the Sentinel application configuration file. For additional details, see the Sentinel Event Router installation manual for the z/OS platform. |

[TcpSource] section

In the [TcpSource] section, assign values to the parameters that manage the transfer of messages from source applications to the Event Router.

The following table introduces the parameters that are in the [TcpSource] section and describes their values.

| Parameter | Value |
|-------------------|---|
| sap | The TCP/IP port that the Event Router uses to receive messages from source applications. By default, the value of this parameter is 2305. |
| sapssl | The TCP/IP port that the Event Router uses to receive messages on a SSL connection from source applications. A value of zero, which is the default, means that the Event Router does not support SSL. |
| Profile | The name of the section in the security profile file which describes the parameters for the SSL session. |
| local_address | The name or TCP/IP address of the Event Router. |
| incoming_max | The maximum number of source applications that can simultaneously connect to the Event Router. By default, the value of this parameter is 10 (ten). |
| tcp_org_port_file | The path for the file containing the rules concerning the local outgoing port to be used for TCP/IP connections. Default: none. |
| tcp_proxi_file | The path for the file containing the rules concerning the use of a proxy. Default none. |

| Parameter | Value |
|------------------------|---|
| tcp_listen_max_retry | Parameters used if the connection to the server is interrupted. (Port busy, Sentinel server stopped, etc.): |
| tcp_watch_delay | <ul style="list-style-type: none"> We can use tcp_listen_max_retry to specify the number of attempts to connect to server: <ul style="list-style-type: none"> If tcp_listen_max_retry= 0 the system tries to connect until successful. If tcp_listen_max_retry = nn the system tries to connect nn times. The parameter tcp_watch_delay specifies the Time in seconds for the first connection. The parameter tcp_listen_delay_retry specifies the Time in seconds between each connection. |
| tcp_listen_delay_retry | |

IPv6 configuration parameters

| Parameter | Description |
|--------------------------|---|
| ipv6_disable_connect = 1 | Indicates that an address or a name used by Transfer EVENT ROUTER to connect to a host may be either an IPV4 or an IPV6 address. When using a name, this parameter can refer to a list of addresses, of any type. |
| ipv6_disable_listen = 1 | Indicates that an address or name used by Transfer EVENT ROUTER to listen for incoming connections may be either an IPV4 or an IPV6 address. When using a name, this parameter can refer to a list of addresses, of any type. |

It is recommended not to set `ipv6_disable_listen=1`, and `ipv6_disable_connect=0`.

It is also possible to configure IPV6 addresses instead of names in the Event Router configuration files. As the configuration options only apply to hostname resolution, Event Router will use these IPV6 addresses, regardless of the IPV6 configuration options setting.

When `ipv6_disable_connect = 0`, a hostname used by Event Router to connect to a host is assumed to refer to either an IPV4 address or to an IPV6 address, or possibly to a list of addresses of any type in the case of a name.

When a name resolution request returns a list of several entries, EVENT ROUTER tries all these entries successively until either the connection succeeds or the list is exhausted. This behavior allows, in situations where the name refers to both IPV4 and IPV6 addresses, to successfully connect to a remote service that is listening to either IPV6 TCP connections or IPV4 TCP connections but not to both address types.

When `ipv6_disable_listen = 0`, a hostname used by Event Router to listen to incoming connections is assumed to refer to either an IPV4 address or to an IPV6 address, or possibly to a list of addresses of any type in the case of a name.

When a name resolution request returns a list of several results, Event Router only listens to the first entry of the list. When the resolution contains both IPV4 and IPV6 entries, the first entry refers generally to an IPV6 address although it is not guaranteed that all OSs actually behave this way.

This limitation to listening only to the first entry (assumed to belong to the IPV6 family) results in different behaviors depending on the platform capability of listening to both IPV4 and IPV6 calls incoming on a IPV6 listening socket or not. In the latter case, only calls coming from IPV6 clients will be notified to the corresponding Event Router server listening to IPV6 connection.

Although `ipv6_disable_connect` and `ipv6_disable_listen` configuration parameters can be independently configured to either "0" or "1", It is not recommended to set `ipv6_disable_listen = 0` and `ipv6_disable_connect = 1`.

[DefaultTarget] section

In the [DefaultTarget] section, assign default values to Target Parameters. When you do not define a Target Parameter for a specific target, the Event Router uses the default value of the relevant Target Parameter.

The following table introduces the parameters that are in the [DefaultTarget] section and describes their values.

| Parameter | Value |
|--------------------------------------|--|
| directory | <p>The path to the overflow and batch files. The default value of this parameter is the sub directory data.</p> <p>On z/OS platforms, do not enter a value for this parameter.</p> <p>In the Target Parameters file, define overflow and batch files for specific Agent Targets.</p> |
| max_messages | <p>The maximum number of messages that the overflow and batch files can store. By default, the value of this parameter is 10 000.</p> |
| timeout | <p>The number of seconds that the Event Router waits for targets to acknowledge receipt of a message that the Event Router sends.</p> <p>When targets do not acknowledge receipt of messages during the timeout period, the Event Router generates an error message in the log file and disconnects from the target. After the waiting period, the Event Router tries to send the messages again.</p> <p>By default, the value of this parameter is 5 (five).</p> |
| short_wait jump_wait long_wait | <p>The number of seconds in the:</p> <ul style="list-style-type: none"> • short wait. The default value is 10. • jump wait. The default value is 20. • long wait. The default value is 300. |

| Parameter | Value |
|---------------------|---|
| keep_ connection | The number of seconds that the Event Router maintains the connections to targets after successfully sending messages to the targets. By default, the value of this parameter is 30 . |
| heartbeat | The number of minutes between successive emissions of HeartBeat Event messages from the Event Router to the target. By default the value of this parameter is 0 (no emissions). |
| address | TCP/IP address of the default Sentinel Server. |
| port | TCP/IP port of the default Sentinel Server. |
| backup_ address | Optional. TCP/IP address of the default Backup Sentinel Server. If defined, when connection to primary address server fails, an attempt to connect to the backup address is done. |
| backup_port | Optional. TCP/IP port of the default Backup Sentinel Server. |
| ssl | Flag indicating whether the connection should be SSL. Allowed values: Yes, Y, No, N. Any other value will be treated as NO , the default. |
| Profile | The name of the section in the security profile file which describes the parameters for the SSL session. |

About short waits, jump waits, and long waits

Both the General Parameters File and the Target Parameters File include a set of short wait, jump wait, and long wait parameters. When the Event Router cannot send one or more messages to a target, these parameters specify when the Event Router tries to send the messages again.

Each time the Event Router cannot send one or more messages to an target, the Event Router tries to send them again after a period of time that is referred to as the waiting period. This waiting period is expressed via the following formula:

$$\text{waiting period} = \text{short wait} + ((n-1) * \text{jump wait})$$

Where:

- *n* is the number of times that the Event Router tries to send messages
- *short wait* is a period of time that is expressed in seconds
- *jump wait* is a period of time that is expressed in seconds

Notice that each unsuccessful attempt to send messages increases the waiting period.

The Event Router uses the formula shown above until this formula returns the long wait. When the long wait is reached, the Event Router stops calculating the waiting period and uses the long wait as the waiting period.

The following is an example of how the Event Router uses the *short wait*, *jump wait*, and *long wait* parameters.

- If short wait is 30 seconds, then the maximum number of times the Event Router increases the waiting period is four ($n=4$)
- If jump wait is 60 seconds
- If long wait is 210 seconds

| Unsuccessful attempt to send messages | Waiting Period, expressed in seconds |
|---------------------------------------|--------------------------------------|
| First | $30 + ((1-1) * 60) = 30$ |
| Second | $30 + ((2-1) * 60) = 90$ |
| Third | $30 + ((3-1) * 60) = 150$ |
| Fourth | $30 + ((4-1) * 60) = 210$ |
| All subsequent attempts | 210 |

[Sentinel] section

In the [Sentinel] section, assign values to the parameters that activate the SNTL entity and identify the Logging Level of this entity.

The following table introduces the parameters that are in the [Sentinel] section and describes their values.

| Parameter | Value |
|-------------|---|
| active | One of the following <ul style="list-style-type: none"> • 0: The SNTL entity is not activated. The Event Router cannot send messages to any Sentinel Servers. • 1: The SNTL entity is activated. The Event Router can send messages to one or more Sentinel Servers. |
| log | The Logging Level of the SNTL entity. By default, the value of this parameter is 0 (zero). |
| buffer_file | z/OS only. The name of the overflow/batch file for the SNTL entity. By default, the DDNAME BUFFER identifies the value of this parameter. Respect the z/OS conventions for entering file names. Enter either: <ul style="list-style-type: none"> • "DD:BUFFER", or • "'file name'" |

| Parameter | Value |
|----------------------------------|---|
| sap (serving Event Routers only) | Listen port for a serving Event Router (target mode type QLTSRV) for connections to a requesting Event Router. This port must be different from the port defined in the TcpSource sap parameter, which is used by the application connecting to the Event Router. |
| sapssl | Listening port for an Event Router server (target mode type QLTSRV) for connections to a requesting Event Router, except for an SSL session. The same remarks for sap applied. A value of zero, which is the default, means that the Event Router does not support SSL. |
| Profile | The name of the section in the security profile file that describes the parameters for the SSL session. |

[MQSeries] section

In the [MQSeries] section, assign values to the parameters that implement the exchange of messages with WebSphere MQ.

| Parameter | Value |
|------------|---|
| in_active | One of the following: <ul style="list-style-type: none"> 0: The MQIN entity is not activated. The Event Router cannot retrieve messages from a queue. 1: The MQIN entity is activated. The Event Router can retrieve messages from a queue. |
| out_active | One of the following: <ul style="list-style-type: none"> 0: The MQOUT entity is not activated. The Event Router cannot send messages to a queue. 1: The MQOUT entity is activated. The Event Router can send messages to a queue. |
| log | The Logging Level of the MQOUT and MQIN entities. By default, the value of this parameter is 0 (zero). |
| qmgr | The name of the queue manager with which the Event Router exchanges messages. If you do not assign a value to this parameter, the Event Router exchanges messages with the default queue manager. WebSphere MQ identifies this default queue manager. |
| in_queue | The name of the queue from which the MQIN entity retrieves messages. |
| in_qretry | The name of the retry queue in which the MQIN entity inserts messages that the DISP entity cannot dispatch. |

| Parameter | Value |
|-------------|---|
| buffer_file | <p>z/OS only.</p> <p>The name of the overflow/batch file for the MQIN entity. By default, the DDNAME. BUFMQM identifies the value of this parameter.</p> <p>Respect the z/OS conventions for entering file names. Enter either:</p> <ul style="list-style-type: none"> • "DD:BUFFER", or • "file name" |

Note MQSeries is not supported on Sentinel Event Router IBM i platform.

General Parameters example

Caution: If you modify any element in this file except the values of attributes, then Sentinel will not parse the contents of this file correctly.

The following is an example of the General Parameters File. To see details about a section in this file, click the relevant hyperlink.

```
[Agent]
name="test"
target_parameters_file="$trk_home_dir/conf/target.xml"
security_profile_file="$trk_home_dir/conf/sslconf.ini"
log=0
message_size=4000
[TcpSource]
sap=1504
sapssl=0
profile=
local_address="hpx3.pa.sopra"
incoming_max=10
[DefaultTarget]
directory="$trk_home_dir/data"
max_messages=1000
timeout=5
short_wait=10
jump_wait=20
long_wait=300
keep_connection=30
address="sentinelserver.pc.pa.sopra"
port=1305
ssl=no
profile=
[Sentinel]
active=1
log=0
```

```
sapssl=0
profile=
[MQSeries]
out_active=0
in_active=0
log=0
qmgr=""
in_queue=""
inquiry=""
```


Target Parameters

4

In the Target Parameters File, set the Target Parameters for specific Event Router targets. You can also use this file to reset the default Target Parameters for all Event Router targets. The default values in the Target Parameters File take priority over the default values that are defined in the General Parameters File.

The Target Parameters File is an XML file. In this file, each XML attribute corresponds to a single Target Parameter. The path to the Target Parameters File is defined in the [Agent] section of the General Parameters File. By default, the path to this file is `$trk_home_dir/conf/target.xml`.

When the Target Parameters File is delivered with the software, some of the Target Parameters are already set for you. You can usually test the Event Router with these settings. However, because these settings only identify one target and do not specify any batch processing, they are not adequate for a real Event Router environment.

Caution: Do not modify the XML grammar of the Target Parameters File, otherwise Sentinel will incorrectly parse the content of this file. Also keep in mind that the XML parser is case sensitive to the names of XML elements and attributes.

Target Parameters File elements

The following table introduces the XML elements of the Target Parameters File.

| The element... | Contains the parameters that... |
|-----------------|--|
| <TrkXml> | Identify the current version of the Target Parameters File (target.xml) |
| <EventRouter> | Identify the Event Router. This element may contain default values for the Target parameters. These values take precedence over the ones defined in the General parameter file (DefaultTarget section). |
| <Target> | Identify an Event Router target and manage the transfer of messages to the target. For each target, the Target Parameters File must include one <Target> element. |
| <Route> | Identify the conditions for routing messages to Event Router targets |
| <Authorization> | Identify the Correlation Servers that can send commands to the Event Router |

<TrkXml> section

The Target Parameters File (target.xml) contains one and only one <TrkXml> element. This mandatory element contains a single attribute: version. The value of this attribute identifies the current version of the Target Parameters File. Sentinel uses this value to parse the Target Parameters File. **Do not modify the value of the version attribute.**

The current version of the Target Parameters File is 1.0.

<EventRouter> section

The Target Parameters File contains one and only one <EventRouter> element. This mandatory element contains a single attribute: name. The value of this attribute is the name of the Event Router.

The value of the name attribute must match the Event Router name that is defined in the General Parameters File. When the value of this attribute is empty, the Event Router retrieves the Event Router name from the General Parameters File. The maximum length of this name is 50 characters.

The <EventRouter> element nests the following elements:

- <DefaultTarget>
- <Exceptions>

<DefaultTarget>

Each <EventRouter> element nests one and only one <DefaultTarget> element. In turn, this optional element nests other elements. These nested elements can contain default values for some of the Target Parameters.

Some of the Target Parameters in the <DefaultTarget> element are also contained in the General Parameters File. In this topic, an asterisk (*) identifies these Target Parameters. The default values that you specify for these Target Parameters overwrite the corresponding values in the General Parameters File.

The following table introduces the elements that the <DefaultTarget> element nests.

| The element... | Contains the parameters that specify defaults for... |
|----------------|--|
| <Access> | Where and how the Event Router sends messages. |

| The element... | Contains the parameters that specify defaults for... |
|----------------|--|
| <Connection> | <p>How long the Event Router waits:</p> <ul style="list-style-type: none"> • To disconnect from targets after successfully sending messages to the targets • To reconnect to targets after each failed attempt to send messages to the targets • For a response from targets after successfully sending messages to the targets |
| <File> | Where the Event Router stores messages that the Event Router cannot send. |

<Access>

The <DefaultTarget> element nests one and only one <Access> element. Use this optional element to assign default values to the Target Parameters that specify where and how the Event Router sends messages.

The following table introduces the <Access> attributes and describes their values.

| Attribute | Value |
|-----------|---|
| mode | <p>The default method that the Event Router uses to send messages. The possible values of this parameter are:</p> <ul style="list-style-type: none"> • QLT: The Event Router uses TCP/IP protocol to send messages to one or more targets • HTTP: The Event Router uses HTTP protocol to send messages to one or more targets • MQSeries: The Event Router uses the WebSphere API to insert messages in a WebSphere queue • QLTREQ: The Event Router establishes a connection to an serving Event Router (target-type QLTSRV) to retrieve the message saved in the associated buffer file. Use this attribute when configuring your Event Router to connect to an Event Router located in a DMZ. • QLTSRV: The serving Event Router waits for a target-type QLTREQ Event Router to establish a connection. Once the connection is established , it sends the messages saved in the buffer file. Use this attribute when configuring your Event Router to function in a DMZ. <p>By default, the value of this parameter is QLT.</p> |
| *addr | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is a TCP/IP address that replaces the default address of Sentinel Server.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |

| Attribute | Value |
|--------------|---|
| *port | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP port of the default Sentinel Server.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| *backup_addr | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is a TCP/IP address that replaces the default backup address of Sentinel Server. The Event Router ignores this attribute for all other mode attribute values.</p> |
| *backup_port | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP port of the default Backup Sentinel Server. The Event Router ignores this attribute for all other mode attribute values.</p> |
| *ssl | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute indicates whether the connection is SSL.</p> <ul style="list-style-type: none"> • yes: The connection is in SSL. The profile parameter must be declared either in the profile attribute or in the default_target section of the parameter file (profile parameter). • no: The connection is not in SSL. |
| *profile | <p>The name of the section in the security profile file that describes the parameters for the SSL session.</p> |
| *qmgr | <p>When the value of the mode attribute is MQSeries, specify the name of the queue manager with which the Event Router exchanges messages. If you do not assign a value to this parameter, the Event Router exchanges messages with the default queue manager that is identified in WebSphere MQ.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| queue | <p>When the value of the mode attribute is MQSeries, specify the name of the MQSeries queue in which the Event Router puts the messages.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| ident | <p>When the value of the mode attribute is QLTREQ, specify the protocol identification sent by the requesting Event Router. This identification must be the name of the target-type QLTSRV (serving) Event Router.</p> <p>The default value for this attribute is the target name, as defined in the Target Element.</p> |

<Connection>

The <DefaultTarget> element nests one and only one <Connection> element. Use this optional element to assign default values to the Target Parameters that specify the amount of time that the Event Router waits:

- To disconnect from targets after successfully sending messages to the targets
- To reconnect to targets after each failed attempt to send messages to the targets
- For a response from targets after successfully sending messages to the targets

The following table introduces the <Connection> attributes and describes their values.

| Attribute | Value |
|------------------|---|
| *short_wait | The number of seconds in the: |
| *jump_wait | <ul style="list-style-type: none"> • short wait |
| *long_wait | <ul style="list-style-type: none"> • jump wait • long wait |
| *keep_connection | The number of seconds that the Event Router maintains the connections with targets after successfully sending messages to those targets. |
| *timeout | <p>By default, the number of seconds that the Event Router waits for targets to acknowledge receipt of one or more messages that the Event Router sends. The value of the timeout period is expressed in seconds.</p> <p>When targets do not acknowledge receipt of messages during the timeout period, the Event Router generates an error message in the log file and disconnects from the target. After the waiting period, the Event Router tries to send the messages again.</p> |

<File>

The <DefaultTarget> element nests one and only one <File> element. Use this optional element to declare default settings for the overflow files of Event Router Targets.

The following table introduces the <File> attributes and describes their values.

| Attribute | Value |
|---------------|--|
| name | Because the name of each overflow file must be unique, the Event Router assigns the name of an target to the overflow file for the target. The Event Router will ignore any value that you assign to this attribute. Leave this attribute empty. |
| *directory | <p>The path to the default overflow files.</p> <p>On z/OS platforms, do not enter a value for this parameter.</p> |
| *max_messages | The maximum number of messages that the default overflow files can store. When the number of messages in an overflow file reaches the value of max_messages, the Event Router no longer accepts new messages for the relevant target. |

<Exceptions>

The <EventRouter> element nests one and only one <Exceptions> element. Use this optional element to specify whether or not the Event Router routes messages that:

- Do not contain correct XML formatting
- Do not match the relevant DTD

The following table introduces the <Exceptions> attributes and describes their values.

| Attribute | Value |
|-----------|--|
| keep | <p>One of the following:</p> <ul style="list-style-type: none"> • YES: The Event Router routes Exceptions. • NO: The Event Router deletes Exceptions. <p>By default, the value of this parameter is NO.</p> |
| target | <p>When the value of the keep attribute is YES, the value of this parameter is the name of an Event Router target. The Event Router will route Exceptions to this Event Router target. Remember to identify the Event Router target for Exceptions via a <Target> element.</p> |

<Target> section

For each Event Router target, the Target Parameters File (target.xml) must contain one and only one <Target> element. In turn, this mandatory element nests other elements. Taken together, these elements describe the configuration of a single Event Router target.

The following table introduces the <Target> element and the elements that are nested in the <Target> element.

| The element... | Contains the parameters that specify... |
|----------------|--|
| <Target> | The logical properties of an Event Router target. |
| <Access> | The physical properties of an Event Router target. |
| <Connection> | <p>The period of time that the Event Router waits:</p> <ul style="list-style-type: none"> • To disconnect from an Event Router target after successfully sending messages to the target • To reconnect to an Event Router Target after a failed attempt to send messages to the target • For a response from an Event Router Target after successfully sending messages to the target |
| <File> | The overflow file for an Event Router target. |
| <SendIf> | <p>The conditions in which the target-type QLTREQ (requesting) Event Router establishes a connection to a QLTSRV type (serving) Event Router.</p> <p>Use this element when configuring your Event Routers to function with a DMZ.</p> |
| <Batch> | The batch file for an Event Router target. |

<Target>

Use the attributes of the <Target> element to define the logical properties of an Event Router target. This element is mandatory.

The following table introduces the <Target> attributes and describes their values.

| Attribute | Value |
|-------------|--|
| name | <p>The name of the Event Router target</p> <p>If you use the Tracking API to send messages to an Event Router target and explicitly identify the target in a <TrkRoute> element, the name in the <TrkRoute> element must match the name that you specify here.</p> |
| defaultXntf | <p>Note: Only the oldest version of XFB monitors (CFT and Inter.Pel) send messages in XNTF format. One of the following:</p> <ul style="list-style-type: none"> • yes: The relevant Event Router target is the default target for all XNTF messages. • no: The relevant Event Router target is not the default target for all XNTF messages. The XNTF messages that the Event Router sends to the target must explicitly identify the target. • norules: The relevant Event Router target is the default target for all XNTF messages except if one of the routing conditions is true for the message. |
| defaultXml | <p>One of the following:</p> <p>yes: The Event Router target is the default target for all XML messages.</p> <p>no: The Event Router Target is not the default target for all XML messages. The XML messages that the Event Router sends to the target must explicitly identify the target.</p> <p>norules: The relevant Event Router target is the default target for all XNTF messages except if one of the routing conditions is true for the message.</p> |
| heartbeat | <p>The number of minutes between successive emissions of Heartbeat Event messages from the Event Router to the target. The value "0" (zero) indicates no emissions.</p> <p>The default value for this parameter is the value of the heartbeat parameter of the General Parameters file, DefaultTarget section.</p> |

<Access>

Each <Target> element nests one and only one <Access> element. Use the attributes of this element to identify the physical properties of an Event Router target. When the value of any of the <Access> attributes is empty, the Event Router uses the relevant default value.

The following table introduces the <Access> attributes and describes their values.

| Attribute | Value |
|-------------|---|
| mode | <p>The method that the Event Router uses to send messages to the Event Router target. The possible values of this parameter are:</p> <ul style="list-style-type: none"> • QLT: The Event Router Target is a Sentinel Server or an XFB Scope Server. The Event Router uses TCP/IP protocol to send messages to the target. • HTTP: The Event Router Target is a Sentinel Server. The Event Router uses HTTP protocol to send messages to the target. • MQSERIES: The Event Router uses the API for WebSphere MQ to insert messages in a WebSphere queue. • QLTREQ: The Event Router establishes a connection to an serving Event Router. (target-type QLTSRV) to retrieve the message saved in the associated buffer file. Use this attribute when configuring your Event Router to connect to an Event Router in a DMZ. • QLTSRV: The serving Event Router waits for a target-type QLTREQ Event Router to establish a connection. Once the connection is established, it sends the messages saved in the buffer file. Use this attribute when configuring your Event Router to function in a DMZ. |
| addr | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP address of the relevant Sentinel Server</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> <p>For addresses of proxy servers, enter the Hostname followed by the "@" character, followed by the proxynome as defined in the proxy configuration file.</p> <p>Example for a server: addr = Hostname</p> <p>Example for a proxy server: addr = Hostname@proxynome</p> |
| port | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP port of the relevant Sentinel Server.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| backup_addr | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP backup address for the Sentinel Server, if defined. When connection to primary address server fails, an attempt to connect to the backup address is done. The Event Router ignores this attribute for all other mode attribute values.</p> |
| backup_port | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute is the TCP/IP port of the relevant Sentinel Server. The Event Router ignores this attribute for all other mode attribute values.</p> |

| Attribute | Value |
|-----------|--|
| ssl | <p>When the value of the mode attribute is QLT, QLTREQ or HTTP, the value of this attribute indicates whether the connection should be SSL.</p> <ul style="list-style-type: none"> • yes: The connection is in SSL. The profile parameter must be declared either in as attribute or in the default_target section of the parameter file. • no: The connection is not in ssl. |
| profile | <p>The name of the section in the security profile file that describes the parameters for the SSL session.</p> |
| qmgr | <p>When the value of the mode attribute is MQSERIES, the value of this attribute is the name of the target queue manager. By default, the value of this attribute is the name of the local queue manager.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| queue | <p>When the value of the mode attribute is MQSERIES, the value of this attribute is the name of the MQSERIES queue. You can use a queue alias</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |
| ident | <p>When the value of the mode attribute is QLTREQ, specify the protocol identification sent by the requesting Event Router. This identification must be the name of the target-type QLTSRV (serving) Event Router.</p> <p>The default value for this attribute is the target name, as defined in the Target Element.</p> <p>Use this attribute when configuring your Event Router to connect to an Event Router in a DMZ.</p> <p>The Event Router ignores this attribute for all other mode attribute values.</p> |

<Connection>

Each <Target> element nests one and only one <Connection> element. This optional element contains the attributes that specify the amount of time that the Event Router waits:

- To disconnect from the relevant Event Router Target after successfully sending messages to the target
- To reconnect to the relevant Event Router Target after each failed attempt to send messages to the target
- For a response from the relevant Event Router Target after successfully sending messages to the target

When the value of any of the <Connection> attributes is empty, the Event Router uses the relevant default value.

The following table introduces the <Connection> attributes and describes their values.

| Attribute | Details |
|--------------------------------------|---|
| short_wait jump_wait long_wait | The number of seconds in the: <ul style="list-style-type: none"> • short wait • jump wait • long wait |
| keep_ connection | The number of seconds that the Event Router maintains the connection with the relevant Event Router target after successfully sending messages to the target. |
| timeout | The number of seconds that the Event Router waits for the relevant Event Router target to acknowledge receipt of one or more messages that the Event Router sends. When the Event Router target does not acknowledge receipt of messages during the timeout period, the Event Router generates an error message in the Log File and disconnects from the Event Router target. After the waiting period, the Event Router tries to send the messages again. |

<File>

Each <Target> element nests one and only one <File> element. Use the attributes of this optional element to set the parameters that define an overflow file for the relevant Event Router target.

When the value of any of the <File> attributes is empty, the Event Router uses the relevant default value.

The following table introduces the <File> attributes and describes their values.

| Attribute | Details |
|------------------|---|
| name | The name of the overflow file. The maximum length of this name is 50 characters. |
| directory | The path to the overflow file. On z/OS platforms, do not enter a value for this parameter. |
| max_ messages | The maximum number of messages that the overflow file can store. When the number of messages in the overflow file reaches the value of max_messages, the Event Router no longer accepts new messages for the Event Router target. |

<SendIf> (QLTREQ mode only)

Each <Target> element nests one and only one <SendIf> element. When you configure your Event Router to send requests to a DMZ (QLTSRV mode) Event Router, use this element to specify the conditions under which the Event Router establishes a connection. Use this element for QLTREQ mode processing only.

The number of attributes that the <SendIf> element contains depends on the number of conditions that you want to define for the Event Router/QLTSRV connection. Each attribute specifies one condition. To logically join multiple conditions, the Event Router uses the OR operator, as shown in the following example.

```
If<SendIf time="10:05" period="30"/>
```

Then, the Event Router establishes the connection with an QLTSRV target when one of the following conditions is true:

- The current time is 10:05
- 30 minutes have passed since the Event Router last retrieved messages from the destination

The following table introduces the possible attributes of the <SendIf> element and the corresponding values.

| Attribute | Details |
|-----------|---|
| period | A time period triggers processing. This time period is expressed in minutes. For example, if period=30, the Event Router establishes the connection every thirty minutes. |
| time | A time triggers processing. This time is expressed in hh:mm format. For example, if time = 1:00, the Event Router establish connection at one o'clock in the morning. |

<Batch>

Each <Target> element nests one and only one <Batch> element. In turn, this mandatory element nests other elements. Taken together, these elements define batch processing for the relevant Event Router target.

The following table introduces the <Batch> element and the elements that are nested in the <Batch> element.

| The element... | Contains the parameters that... |
|----------------|---|
| <Batch> | Activate batch processing for an Event Router target. |
| <File> | Identify the batch file for an Event Router target. |
| <SendIf> | Specify the conditions for batch processing for an Event Router target. |

<Batch>

The <Batch> element contains the attributes that activate batch processing for the relevant Event Router target.

The following table introduces the <Batch> attributes and describes their values.

| Attribute | Value |
|-----------|---|
| active | One of the following: <ul style="list-style-type: none"> • YES: batch processing is activated for the Event Router target. • NO: batch processing is not activated for the Event Router target. |

| Attribute | Value |
|-----------|--|
| sendAlert | <p>One of the following:</p> <ul style="list-style-type: none"> • YES: batch processing does not apply to Alerts. When the Event Router receives an Alert from a source application, the Event Router immediately sends the Alert to the relevant Event Router target. If the Event Router cannot send the Alert, the Event Router stores the Alert in the overflow file that is defined for the relevant Event Router target. • NO: batch processing applies to Alerts. When the Event Router receives an Alert, the Event Router stores the Alert in the batch file. |

<File>

Each <Batch> element nests one and only one <File> element. When you activate batch processing via the <Batch> element, the <File> element is mandatory. Use this element to define a batch file for the relevant Event Router target.

The following table introduces the <File> attributes and describes their values.

| Attribute | Value |
|--------------|---|
| name | The name of the batch file. The maximum length of this name is 50 characters. Keep in mind that no default value exists for this attribute value. |
| directory | <p>The path to the batch file.</p> <p>On z/OS platforms, do not enter a value for this parameter.</p> |
| max_messages | The maximum number of messages that the batch file can store. When the number of messages in the batch file reaches the value of max_messages, the Event Router no longer accepts new messages for the Event Router target. |

<SendIf>

Each <Batch> element nests one and only one <SendIf> element. Use this element to specify the conditions under which the Event Router sends batches of messages from the batch file to the relevant Event Router target.

The number of attributes that the <SendIf> element contains depends on the number of conditions that you want to define for batch processing. Each attribute specifies one condition. To logically join multiple conditions, the Event Router uses the operator **OR**, as shown in the following example.

```
If<SendIf nb_messages="100" period="30"/>
```

Then, the Event Router sends messages from the relevant batch file when one of the following conditions is true:

- The batch file contains 100 messages
- 30 minutes have passed since the Event Router last sent messages from the batch file

When you activate batch processing via the <Batch> element but leave the <SendIf> element empty, no conditions for batch processing exist. In this case, use the **force** command to send messages from the batch file.

The following table introduces the possible attributes of the <SendIf> element and the corresponding values.

| Attribute | Details |
|-------------|--|
| nb_messages | The number of messages in the batch file triggers batch processing. For example, if nb_messages = 100, the Event Router sends messages from the batch file when this file contains 100 messages. |
| period | A time period triggers batch processing. This time period is expressed in minutes. For example, if period=30, the Event Router sends messages from the batch file every thirty minutes. |
| time | A time triggers batch processing. This time is expressed in hh:mm format. For example, if time = 1:00, the Event Router sends messages from the batch file at one o'clock in the morning. |

<Route> section

The Target Parameters File (target.xml) can contain one or more <Route> elements, which in turn nest other elements. Taken together, these optional elements can specify routing instructions for:

- Only the messages that are described by a single Tracked Object
- All messages that are described by a Tracked Object

The following table introduces the <Route> element and the elements nested in the <Route> element.

| The element... | Contains the parameters that... |
|----------------|--|
| <Route> | Identify a single Tracked Object or, alternatively all Tracked Objects and specify a default routing instruction for the associated messages |
| <Condition> | Specify conditions for: <ul style="list-style-type: none"> • Sending the messages that you identify in <Route> • Not sending the messages that you identify in <Route> |

About test conditions and default instructions

Each time the Event Router receives a message, the Event Router searches for any test conditions that you defined for the relevant Tracked Object. If you defined these conditions, they appear in the `<Route>` element for the relevant Tracked Object.

As the following example shows, the actions that entity DISP performs after this initial search depend on the content of the Target Parameters file:

Test the conditions defined in `<Route object="RelevantTrackedObject">`

If a condition is true for a specific Event Router target

Notify the target

If a condition is false for a specific Event Router target

Do not notify the target

If conditions conflict for a specific Event Router target (*true and false*)

Execute the default instruction defined in `<Route object="RelevantTracked">`

If no condition are set for the RelevantTrackedObject, test the conditions defined in `<Route object="All">`

If a condition is true for a specific Event Router target

Notify the target

If a condition is false for a specific Event Router target

Do not notify the target

If conditions conflict for a specific Event Router target (*true and false*)

Execute the default instructions defined in `<Route object="All">`

`<Route>`

Use the attributes of the `<Route>` element to identify messages that you want the Event Router to route and default routing instructions for these messages. You can identify:

- All messages that are described by a Tracked Object
- Only the messages that are described by a single Tracked Object

The following table introduces the `<Route>` attributes and describes their values.

| Attribute | Value |
|----------------|---|
| object | <p>One of the following:</p> <ul style="list-style-type: none"> • CycleLink: The Event Router applies the relevant routing conditions to the cycle link messages. • All: The Event Router applies the relevant routing conditions to all messages that are associated with a Tracked Object not defined with a specific route definition. • <i>Name of a Tracked Object:</i> The Event Router applies the relevant routing conditions only to messages that are associated with the specified Tracked Object. <p>The wildcard characters ? and * are accepted for Name of a Tracked Object:</p> <ul style="list-style-type: none"> • Use the question mark (?) to represent exactly one character. All other characters specified are required in matching file names. • Use the asterisk character (*) to represent zero or more characters. <p>If wildcard characters are used, the match for routing condition is:</p> <ul style="list-style-type: none"> • Same name defined • Name correspond to first generic name defined • (if defined) object ALL <pre> <Target name="DEFAULT" defaultXntf="no" defaultXml="yes"> </Target> <Target name="BACKUP" defaultXntf="no" defaultXml="no"> </Target> <Route object="XFB*" default_Notify="NotifyIf"> <Condition notify="NotifyIf" target="BACKUP" if=" [Application]=COPY" /> <Condition notify="NotNotifyIf" target="DEFAULT" if=" [Application]=COPY" /> </Route> </pre> |
| default_notify | <p>One of the following:</p> <ul style="list-style-type: none"> • NotifyIf: If the results of the test conditions that are specified in the associated set of <Condition> elements conflict, the Event Router sends the relevant message. • NotNotifyIf: If the results of the test conditions that are specified in the associated set of <Condition> elements conflict, the Event Router does not send the relevant message. |

<Condition>

Each <Route> element must nest at least one <Condition> elements. Use the attributes of this element to define the conditions that instruct the Event Router to send or, alternatively not send messages that you identified in the associated <Route> element.

The following table introduces the <Condition> attributes and describes their values.

| Attribute | Value |
|-----------|---|
| notify | <p>One of the following:</p> <ul style="list-style-type: none"> • NotifyIf: The <Condition> element specifies conditions for sending messages. • NotNotifyIf: The <Condition> element specifies conditions for <i>not</i> sending messages. |
| target | Name of the Event Router Target for the relevant messages |
| if | <p>Condition for the content of the relevant messages. You can use the following logical operators:</p> <ul style="list-style-type: none"> • AND • OR <p>The calling function *HASHMOD (P1, P2, P3): This function returns false or true and executes the following operations on its parameters:</p> <ul style="list-style-type: none"> • P1: Message Attribute, operation is HASH (if attribute name is not between [], then the name itself is used) • P2: Modulo to apply to the hash result • P3: Value to be compare with the modulo result <p>The following comparison operators:</p> <ul style="list-style-type: none"> • = • INFEQ * • NOT • > • < • SUPEQ * <p>*INFEQ represents the operator <=.</p> <p>*SUPEQ represents the operator >=.</p> <p>Enclose attribute <i>names</i> in square brackets. Example: [attribute]</p> <p>Note: If the attribute value of the condition contains special characters, enter the value between single quotes (').</p> |

<Authorization> section

The Target Parameters File (target.xml) contains one and only one <Authorization> element. In turn, this optional element nests one or more <exeCmd> elements. Each <exeCmd> element authorizes one or more Sentinel modules to send commands to the Event Router.

<exeCmd>

Use the attributes of the `<exeCmd>` element to identify the name and the password that one or more Sentinel modules use to send commands to the Event Router. The following Sentinel modules can send commands to the Event Router:

- Correlation Server using External Functions
- Monitoring Interface
- Sentinel Monitoring

When the Event Router receives a command from a Sentinel module, the Event Router compares the login name and password of the Sentinel module to the contents of each `<exeCmd>` element. If the login name and password of the Sentinel module:

- Matches the contents of an `<exeCmd>` element, the Event Router executes the relevant command
- Does not match the contents of an `<exeCmd>` element, the Event Router does not execute the relevant command

The following table introduces the `<Authorization>` attributes and describes their values.

| Attribute | Value |
|-----------|---|
| name | The login name for one or more Sentinel modules that sends commands to the Event Router |
| password | The login password for one or more Sentinel modules that sends commands to the Event Router |

Target Parameters example

Caution: If you modify any element in this file except the values of attributes, then Sentinel will not parse the contents of this file correctly. Also keep in mind that the XML parser is case-sensitive to the names of elements and attributes.

The following is an example of the Target Parameters file.

```
<TrkEventRouterCfg>
<TrkXml Version="1" />
<EventRouter name="EventRouterOne">
<DefaultTarget>
<Access addr="sentinelserver.pc.pa.sopra" mode="QLT" port="1302"/>
<Connection short_wait="60" jump_wait="60" long_wait="3600" keep_connection="30"
timeout="10"/>
<File name="" directory="/EventRouter/data/" max_messages="1000"/>
</DefaultTarget>
<Exceptions keep="yes" target="Errmsg"/>
</EventRouter>
<Target name="XNTF" defaultXntf="yes" defaultXml="no">
```

```

<Access mode="QLT" addr="SentinelServer" port="1302"/>
<Connection short_wait="60" jump_wait="120" long_wait="3600" keep_connection="30"
timeout="10"/>
<File name="OverflowOne" directory="/EventRouter/data/" max_messages="1000"/>
<Batch active="yes" sendAlert="yes">
<File name="BatchOne" directory="/EventRouter/data/" max_messages="1000"/>
<SendIf nb_messages="200" period="60"/>
</Batch>
</Target>
<Target name="QLTXML" defaultXntf="no" defaultXml="yes">
<Access port="1303" mode="QLT"/>
</Target>
<Target name="SCOPEV1" defaultXntf="yes" defaultXml="no">
<Access addr="RS16.pa.sopra" port="44444" mode="QLT"/>
</Target>
<Target name="SENTINEL" defaultXntf="yes" defaultXml="yes">
<Access addr="hpx11.pa.sopra" port="1305" mode="QLT"/>
</Target>
<Target name="Errmsg">
<Batch active="yes" sendAlert="yes">
<File name="Errors" directory="/EventRouter/data/data" max_
messages="1000"/>
</Batch>
</Target>
<Route object="All" default_Notify="NotifyIf">
<Condition notify="NotifyIf" target="SENTINEL" if="[PRODUCTNAME]=API-SCRIPT AND
[APPLICATION]=AgentTest">
</Condition>
</Route>
<Route object="XFBLog" default_Notify="NotifyIf">
<Condition notify="NotifyIf" target="SENTINEL" if="[APPLICATION]=AgentTest AND
[SEVERITY]=1">
</Condition>
</Route>
<Authorization>
<exeCmd name="SCOPEV1" password="passwdSCOPE1"/>
<exeCmd name="SENTINEL" password="passwdSENT"/>
</Authorization>
</TrkEventRouterCfg>

```

The following example dispatches the message to different targets using the function HASHMOD (if the application is not CONTROL). If the application is CONTROL, then the message is dispatched to the target set with Defaultxml=norules.

```

=> File

<?xml version="1.0" encoding="ISO-8859-1"?>

```

```
<!-- ***** -->
<!-- *** Send message to destination depending on cycleid *** -->
<!-- *** Each Target used in routing condition is declared with its *** -->
<!-- *** address and port defined explicitly *** -->
<!-- *** If no routing condition apply, destination is DEFAULT *** -->
<!-- *** target whose address and port are set in the ER definition *** -->
<!-- ***** -->
<TrkEventRouterCfg>
  <TrkXml version="1.0" />
  <EventRouter name="QUAL">
    </EventRouter>

    <Target name="DEFAULT" defaultXntf="yes" defaultXml="norules">
    </Target>
    <Target name="MODULO2_0" defaultXntf="no" defaultXml="no">
      <Access port="1305" addr="slnxcftgov.lab1.lab.ptx.axway.int" />
    </Target>
    <Target name="MODULO2_1" defaultXntf="no" defaultXml="no">
      <Access port="1306" addr="slnxcftgov.lab1.lab.ptx.axway.int" />
    </Target>

    <Route object="ALL" default_Notify="NotifyIf">
      <Condition notify="NotifyIf" target="MODULO2_0" if="[APPLICATION] NOT CONTROL
AND HASHMOD([CYCLEID],2,0)"/>
      <Condition notify="NotifyIf" target="MODULO2_1" if="[APPLICATION] NOT CONTROL
AND HASHMOD([CYCLEID],2,1)"/>
    </Route>

  </TrkEventRouterCfg>
```

Concepts

A security profile lists all elements required to establish a SSL session and is linked to an Event Router session when its name is defined as a profile parameter for one of the following:

- Target definition
 - Either explicit or by default
 - The SSL session will always be in client mode except for the target defined with mode "QLTSRV"
- Event Router definition: Session SSL in Mode SERVER
 - Application Incoming session
 - Target defined in mode QLTSRV

All the profiles are declared in the Security Profile file:

- Each section corresponds to a profile
- The security profile is declared in the General Parameter file (trkagent.ini)
 - AGENT section
 - Parameter security_profile_file

A sample (sslconf.ini) is available in the install directory:

- For a new installation, the sample is copied in the conf directory
- The sample file contains a profile description for each Event Router role. However, the definitions do not contain certificate definitions. The certificate parameters must be set by the user before the Event Router can be started. The profiles declared in the sample are:
 - SENTINEL
 - Profile for a target connecting to a Sentinel Server
 - ERSERVER
 - Profile for incoming session started by Sentinel Universal Agent applications
 - The Event Router is a SSL server and requires at least an User Certificate and Root Certificate
 - QLTSERVER
 - Profile for target declared in mode QLTSRV
 - As for ERSERVER profile, it manages incoming sessions and is in Mode Server

Profile parameters

Parameters for SSL session in server mode

| Parameter | Value |
|------------------------------------|---|
| SSL_USER_CERTIFICATE_FILE | Path to file containing the User Certificate. Mandatory. |
| SSL_USER_CERTIFICATE_FORMAT | Describes the certificate format. Supported formats: <ul style="list-style-type: none">• DER (default)• PEM• PKCS12 |
| SSL_SSL_USER_CERTIFICATE_KEY_FILE | Path to file containing the User Certificate key. The format must be the same as the user certificate file. Mandatory if user certificate is PEM or DER format. |
| SSL_USER_CERTIFICATE_PASSWORD_FILE | Path to the file containing the Passphrase. Mandatory if user certificate format is PKCS12. |
| SSL_CA_CERTIFICATE_FILE | Path to a file containing the list of trusted Authorities certificates. |
| SSL_CA_CERTIFICATE_FORMAT | Format of the CA certificate file. Supported formats: <ul style="list-style-type: none">• DER (default)• PEM• PKCS7 |

| Parameter | Value |
|-------------------|---|
| SSL_VERIFY_POLICY | <p>Authentication mode required.</p> <p>The SSL_VERIFY_POLICY and SSL_USER_CERTIFICATE_FILE parameters set the security profile authentication mode.</p> <p>Note: With the SSL protocol, the server determines whether the client must be authenticated. If the server wants the client to be authenticated, then the server must first be authenticated by the client.</p> <p>Authentication rules in Server Mode:</p> <ul style="list-style-type: none"> • PEER - The server and the client must be authenticated. • OPTIONAL - The server and the client must be authenticated. An invalid certificate is tolerated by the server. • NONE (Default) - Only the server must be authenticated. |
| SSL_CIPHER_SUITE | <p>List of algorithms supported (see annex cipher suite for the full list). The list must be in decreasing order of preference.</p> <ul style="list-style-type: none"> • Select up to eight cipher suites, separated by commas <p>Each value defines three algorithms:</p> <ul style="list-style-type: none"> • Authentication algorithm • Encryption algorithm • Sealing algorithm <p>This list is compared with the list proposed by the client in order of preference, for the purpose of determining the suite to be negotiated.</p> <p>The defaults are : 156,157,61,60,53,47</p> |
| SSL_VERSION_MIN | <p>List of accepted protocol versions. The default is ssl_3.0. Select one of the following SSL/TLS versions:</p> <ul style="list-style-type: none"> • ssl_3.0 • tls_1.0 • tls_1.1 • tls_1.2 |

Parameters for SSL session in client mode

| Parameter | Value |
|------------------------------------|--|
| SSL_USER_CERTIFICATE_FILE | <p>Path to file containing the User Certificate. This parameter is:</p> <ul style="list-style-type: none"> • Mandatory if the SERVER has a policy set to PEER • Optional if the SERVER has a policy set to OPTIONAL • Not Needed if the SERVER has a policy set to NONE |
| SSL_USER_CERTIFICATE_FORMAT | <p>Describes the certificate format. Supported formats:</p> <p>DER (default)</p> <p>PEM</p> <p>PKCS12</p> |
| SSL_SSL_USER_CERTIFICATE_KEY_FILE | <p>Path to file containing the User Certificate key. The format must be the same as the user certificate file. Mandatory if user certificate is PEM or DER format.</p> |
| SSL_USER_CERTIFICATE_PASSWORD_FILE | <p>Path to the file containing the Passphrase. Mandatory if user certificate format is PKCS12 .</p> <p>This password can now be encrypted using the utility agtencrypt. See AGTCRYPT utility on page 109 for the utility description and associated parameters for the ER configuration file.</p> <p>Note Even with key encryption, the file should still be protected by specific read permission.</p> |
| SSL_CA_CERTIFICATE_FILE | <p>Path to file containing a Trusted Authorities certificate.</p> |
| SSL_CA_CERTIFICATE_FORMAT | <p>Format of the CA certificate file. Supported formats :</p> <ul style="list-style-type: none"> • DER (default) • PEM • PKCS7 |
| SSL_VERIFY_POLICY | <p>The parameter if present is ignored. The Event Router always sets the policy to PEER:</p> <ul style="list-style-type: none"> • The server certificate is checked. • If the verification process fails the handshake fails immediately. |

| Parameter | Value |
|------------------|--|
| SSL_CIPHER_SUITE | <p>List of algorithms supported (see annex cipher suites for a full list). The list must be in decreasing order of preference.</p> <ul style="list-style-type: none"> Select up to eight cipher suites, separated by comma. <p>Each value defines three algorithms:</p> <ul style="list-style-type: none"> Authentication algorithm Encryption algorithm Sealing algorithm <p>This list is compared with the list proposed by the client in order of preference, for the purpose of determining the suite to be negotiated. The defaults are : 156,157,61,60,53,47</p> |
| SSL_VERSION_MIN | <p>List of accepted protocol versions. The default is ssl_3.0. Select one of the following SSL/TLS versions:</p> <ul style="list-style-type: none"> ssl_3.0 tls_1.0 tls_1.1 tls_1.2 |

Use cases

How to secure messages between applications (using Sentinel Universal Agent) and Event Router:

- In the [TCPSOURCE] section of the General Parameter file, configure:
 - sapssl (SSL listening port)
 - profile (identifier corresponding to a section in the security profile file i.e ERSERVER)
- In the Security Profile file, add or modify a section (i.e [ERSERVER]) defined for SSL in mode server and configure using your own certificates (not delivered with Event Router installation package):

The user certificate:

- SSL_USER_CERTIFICATE_FILE
SSL_USER_CERTIFICATE_FORMAT
- SSL_USER_CERTIFICATE_KEY_FILE (format DER, PEM) or
SSL_USER_CERTIFICATE_PASSWORD_FILE (format PKCS12)

The root certificate:

- SSL_CA_CERTIFICATE_FILE
SSL_CA_CERTIFICATE_FORMAT

The policy, in this sample only the server must be authenticated:

- `SSL_VERIFY_POLICY=none`

The SSL cipher suite:

- `SSL_CIPHER_SUITE`

3. For all applications that require SSL (UA SSL), provide the listening SSL port, the root certificate and the cipher suite used. In Universal Agent this is configuration in the UA configuration file (`trkapi.cfg`) with the following attributes

- Destination and mode
`TRKIPPORT=sapssl`
`TRKUSESSL=1`
- Root Certificate Information
`SSL_CA_CERTIFICATE_FILE`
`SSL_CA_CERTIFICATE_FORMAT`
- `SSL_CIPHER_SUITE`

How to secure messages between Event Router and Sentinel Server:

1. We assume that the default `target.xml` is used. Meaning, only the default values from the General Parameter File are used. Modify the `[defaulttarget]` section in the file:
 - Port - Port On which the server will the listen for SSL sessions.
 - Profile - Identifier corresponding to a section in the security profile file (i.e sentinel).
 - SSL- Set the value to Yes.
2. In the Security Profile file, add or modify a section (i.e `[sentinel]`) and configure:
 - Root Certificate Information.
The root certificate must correspond to the Sentinel Server trusted certificate authority:
`SSL_CA_CERTIFICATE_FILE`
`SSL_CA_CERTIFICATE_FORMAT`
 - `SSL_CIPHER_SUITE`
The list of cipher suites used must match the one accepted by Sentinel Server.

Using the command syntax

6

This section provides what you need to know about the syntax for Event Router commands on UNIX, OS/400, Windows, and z/OS.

UNIX, OS/400 and Windows commands

Command syntax on UNIX, OS/400 and Windows platforms including:

- Setting the environment variables
- Entering a single command
- Entering a series of commands

Setting the environment variables

Before you execute Event Router commands, you must activate a set of Event router environment variables. These variables are stored in the Event Router profile file (conf/profile).

Refer to the platform installation guide for more details about this profile and its environment variables.

Entering a single command

The syntax for entering a single Event Router command is:

agtcmd command

Where *command* is an Event Router command

When you need to execute a series of Event Router commands, start a command session, described next.

Entering a series of commands

During a command session, Sentinel maintains the connection between you and the Event Router. You do not enter *agtcmd* before each command. However, you cannot start and stop the Event Router (*start*, *stop* and *kill* commands) during a command session.

To:

- start a command session, enter: *agtcmd*
- stop a command session, enter one of the following:
 - *end*
 - *quit*
 - *bye*
 - *exit*

z/OS command

The command syntax for the z/OS platform is:

F job, command

Where:

- *job* is the Event Router job
- *command* is an Event Router command

Command HELP

To display a list of commands available on your platform, enter "HELP" or "?".

To display specific detail about a command parameter, enter:

- *command HELP*
- *command ?*

Examples:

- Unix
 - agtcmd help*
 - agtcmd display help*
- z/OS
 - F job, help*
 - F job, display help*

Starting and stopping the Event Router

7

This section describes the commands that start, stop, and monitor the Event Router on UNIX, OS/400, Windows, and MVS platforms

UNIX, Windows and OS/400

The following table lists and describes the commands that start, stop, and monitor the Event Router on UNIX, OS/400 and Windows platforms. When you enter these commands, remember to respect the syntax for UNIX, OS/400 and Windows platforms.

| Command | Details |
|---------|---|
| about | Displays the following details about the software: Name: The name of the script. Version: The version of the script. Copyright: The copyright of the script. Target: Operating System where you generated the product. Sysname: Type of Operating System. Machine: Operating System used. Version: The version of the Operating System. Release: The release of the Operating System. |
| start | Initializes all Event Router processes and loads the Target Parameter File (<i>target.xml</i>) |
| stop | Stops all Event Router processes |
| kill | Abruptly stops all Event Router processes. Use this command only when the stop command does not stop all Event Router processes. |
| status | Displays a list of the Event Router processes that are active |

z/OS

The following table describes how to start, stop, and monitor the Event Router on z/OS platforms. Sentinel includes return codes for the JCL that starts the Event Router.

Descriptions of these return codes appear below the following table.

| To... | Do this: |
|---|--|
| Start the Event Router and all Event Router processes | Submit the <i>AGTRUN JCL</i> . To see a description of the JCL, refer to the installation manual for the Event Router. |
| Stop all Event Router processes | Terminate the Event Router job via one of the following commands: <ul style="list-style-type: none"> <code>P job</code> <code>F job, stop</code> Where <i>job</i> is the name of the Event Router job. |
| See a list of all active Event Router processes | Consult the SYSPRINT. |

Return codes: AGTRUN JCL

The *AGTRUN* JCL runs all Event Router processes. For details about any ABEND codes that this JCL returns, refer to the appropriate IBM documentation. In addition, consult the log and error files.

The following table lists and describes the Sentinel return codes for the *AGTRUN* JCL.

| Return code | Details |
|-------------|---|
| 0 | Stopped AGTRUN. No error occurred. |
| 1 | Cannot start AGTRUN. Error in the parameters for one or more Event Router processes. Check the configuration in the General and Target Parameter Files. |
| 2 | Error initializing the subtask that the Event Router processes use to communicate with each another |
| 3 | Error starting one or more subtasks |
| 90 | Error dynamically allocating a file |
| 93 | A queue (XCF member) is assigned to more than one Event Router process. Check the configuration in the General and Target Parameter Files. |
| 94 | Error in one or more XCF macros. |
| 96 | A job is currently using a queue (XCF member) that you assigned to an Event Router process. |
| 97 | Error using a memory file. |

| Return code | Details |
|-------------|--|
| 98 | The subtask that the Event Router processes use to communicate with each other is not activated. |
| 99 | Error initializing an Event Router process. |

Managing entities

8

The following table lists and describes the commands that manage Event Router entities. When you enter these commands, respect the syntax for Event Router commands.

| Command | Details |
|-------------------------------------|---|
| <code>details -e entity</code> | Displays information about entities. For each entity, the command displays information specific to the entity function. For example, for MQIN, the details command displays the number of messages that the MQIN entity retrieved during the current session. To display information specific to one process, enter the command with the <code>-e</code> parameter. |
| <code>pause -e entity</code> | Currently, the only valid values for the <code>-e</code> parameter are MQIN and ZLGR. DMQIN - Disconnects the MQIN entity from WebSphere MQ. The Event Router cannot receive WebSphere messages. ZLGR - The Event Router stops checking for new messages in the LOGGER file. |
| <code>restart -e entity</code> | Currently, the only valid values for the <code>-e</code> parameter are MQIN and ZLGR. DMQIN - Reconnects the MQIN entity to WebSphere MQ. The Event Router can receive WebSphere messages. ZLGR - The Event Router restarts the check for new messages in the LOGGER file. |
| <code>loadfile -e entity</code> | Currently, the only valid value for the <code>-e</code> parameter is NET or NETS. Asks the entity to load a new copy of parameter files. For NET and NETS this command loads files that describe proxy and port use. |

Managing targets

9

The following table lists and describes the commands that you use to manage connections to Event Router targets. In this documentation, these commands are referred to as the Target Commands. When you enter the Target Commands, respect the syntax for Event Router commands.

| Command | Details |
|---------------------------------------|--|
| <code>reset -t <i>target</i></code> | Reinitializes an Event Router target and checks any conditions that are defined for Batch Processing. Use this command after you correct an error that prevented you from initializing the Event Router. |
| <code>disable -t <i>target</i></code> | Immediately closes the open connection to an Event Router target, and then disables the target. It also disables and closes the open connections to all targets that are routed the same messages as the target specified in this command. If the Event Router receives messages for these targets, the Event Router rejects them. If you execute this command while the Event Router is sending messages to the relevant target, the Event Router routes the unsent messages to the overflow file or, alternatively the batch file. |
| <code>enable -t <i>target</i></code> | If you disabled an Event Router target via the disable command, this command reopens the connection to the target. The Event Router checks any conditions that are defined for Batch Processing and resets the short, jump, and long waits to zero. |
| <code>froze -t <i>target</i></code> | Prevents the Event Router from sending messages to an Event Router target. If the Event Router receives messages for the relevant target, the Event Router stores them in the overflow file or, alternatively the batch file. To reopen the connection to the Target, use the force command. |
| <code>force -t <i>target</i></code> | Immediately sends messages from an overflow or batch file to the specified Event Router target. The Event Router ignores: <ul style="list-style-type: none">• Any conditions that are defined for Batch Processing• The current values of the short, jump, and long waits |

| Command | Details |
|--|---|
| count [-t target] | <p>Displays the message counters for an Event Router target. If you do not identify a target via the target parameter, this command displays the message counters of every target. In the Log File, the Event Router stores the message counters that this command displays.</p> <p>The following is an example of the message counters that the count command displays:</p> <div style="text-align: center;"> <p>Messages sent 500</p> <p>Sent to the target Each time you initialize the Event Router, it resets this counter to zero</p> <p>Messages stored (20, 300)</p> <p>Stored in the overflow file Stored in the batch file</p> </div> |
| resetcount [-t target] | <p>Reset to zero the SENT message counters for an Event Router target. If you do not identify a target via the target parameter, this command resets the counter of every target. The information returned is the same as the count command.</p> |
| loadconfig | <p>Closes all connections to Event Router targets and loads the Target Parameter File (<i>target.xml</i>). This command does not stop the Event Router.</p> <p>Use this command after you reconfigure one or more Event Router targets via the Target Parameters File (<i>target.xml</i>).</p> |
| purge -t target -n number [-ft file type] | <p>Purges the indicated Event Router target. Use the following parameters:</p> <ul style="list-style-type: none"> • -t = target • -n number (1 to 999) of message to purge in the buffer file. The default is none. If n = *, then all messages are purged. • -ft = indicates batch or immediate file type. The default value is I (immediate). |
| display [-t target] | <p>Displays one of the following statuses of an Event Router target:</p> <p>DISABLE You disabled the relevant target via the disable command. The target cannot receive messages.</p> <p>ENABLE You enabled the relevant target via the enable command. The Agent Target can receive messages.</p> <p>ERROR When you initialized the relevant target, an error occurred. The Log and Error Files contain one or more messages that describe the error.</p> <p>INIT When you initialized the relevant target, the Event Router loaded the appropriate Target Parameters. The Event Router is initializing the interface for the target. If this status persists, verify that the relevant Event Router entity is running. Target parameter</p> |

Target parameter

All Event Router commands for managing targets include the same parameter: `target`. The value of this parameter is the name of a target. In the Target Parameters File, the name of each target is defined in a single `<Target>` element.

Managing traces

10

The following table describes the commands that you can use to manage Event Router traces. Parameter descriptions for these commands appear below the table. When you enter the commands, respect the syntax for Event Router commands.

| Command | Details |
|---|---|
| logarchive | <p>For UNIX and Windows:</p> <p>Directs the Event Router to save Log Messages in a single Log File, <i>log.dat</i>. For Windows, this file is located in the <code>\data</code> subdirectory of the product. For UNIX the path to <i>log.dat</i> is:</p> <pre>trk_home_dir/data/log.dat</pre> <p>The <code>LOGARCHIVE</code> command copies the content of the log file into an archive file and then deletes the current log file. After this operation, the log input is recorded in a new empty log file. The archive file name is the original file name suffixed with “_x”, where x is the archive number (1, 2, ..).</p> <p>For z/OS:</p> <p>Directs the Event Router to archive the contents of the current log file and begins recording Event Router traces in the second log file.</p> <p>Two log files exist for this process on z/OS platforms, referenced with ddnames LOGA and LOGB. On startup, the Event Router saves the more recent of these two files, recording new traces to the oldest file of the two.</p> |
| log -l level [-e <i>entity</i>] | <p>Temporarily overrides the default Logging Levels that are set for the Event Router entity. Each time you reinitialize the Event Router, the Event Router erases any temporary settings and reuses the default Logging Levels.</p> <p>Keep in mind that the Event Router records details about errors in the error file. Before you increase the Logging Level of an entity, consult the error file.</p> |
| trace -l level [-e <i>entity</i>] | <p>Activates Error Traces that you can use to debug the Event Router. Use this command only with the help of Axway support staff</p> |

The following table describes the parameters of the **log** and **trace** commands.

| Parameter | Details |
|----------------|--|
| <i>level</i> | Specifies a Logging Level |
| <i>process</i> | Identifies the name of the relevant Event Router entity. When you do not define a value for this parameter, the Event Router applies the relevant command to all entities. |

The Event Router records trace messages that describe events and errors. These messages are referred to as log messages.

Log messages - basics

Where to find log messages

Log messages describe Event Router events. Where the Event Router records log messages depends on the host platform.

- UNIX, OS/400 and Windows: The Event Router records log messages in a single log file: *log.dat*. This file is stored in the data subdirectory for the product.
- z/OS: The Event Router records log messages in two log files. These files are identified by the DDnames *LogA* and *LogB*. When these log files are not identified in the JCLs, the Event Router dynamically allocates memory for them in SYSOUT. Each time you reinitialize the Event Router on a z/OS platform, the Event Router records log messages in the oldest log file. At any time, you can direct the Event Router to switch from one log file to another.

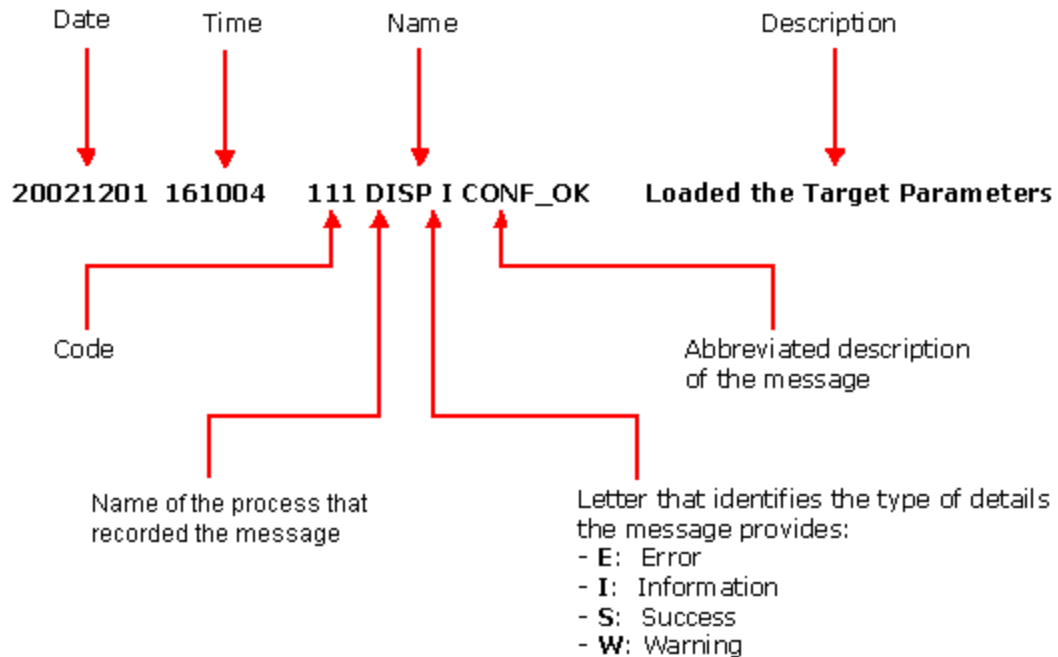
How to interpret log messages

Event Router processes record their own log messages. This means that a log message describes not only an Event Router event, but also a processing event.

In a log file, each log message includes:

- The date and time of the relevant event
- The name that identifies the relevant event
- The description of the relevant event

The following is an example of a log message with details about each part of the message:



How to set log messages

The level of detail in a log message is referred to as the logging level. Because each entity records its own log messages, you can specify one logging level for each Event Router entity. The logging levels that you can specify range from zero to four. These levels are cumulative. For example, if you set the logging level of an entity to two, the process records log messages that are associated with levels zero, one, and two.

By default, the logging level of all entities is zero. In the General Parameters File, you can modify this default value for the entities. Via the **log** command, you can also temporarily override this default value for a specific entity or, alternatively for all entities.

The following table describes the log details that are associated with each logging level.

| Logging level | Events |
|---------------|---|
| 0 | Event Router: <ul style="list-style-type: none"> • Initializes Event Router entities • Encounters an error for a entity • Executes an Event Router command |

| Logging level | Events |
|---------------|--|
| 1 | <p>Event Router entities:</p> <ul style="list-style-type: none"> • Receive requests to open and close connections • Send requests to open and close connections • Send messages from batch files <p>When the logging level is set to one (1), the Event Router records all errors that occur during these events.</p> |
| 2 | <p>Event Router entities open and close connections to targets.</p> <p>When the logging level is set to two (2), the Event Router records all errors that occur during these events.</p> |
| 3 | <p>Event Router entities send, receive, and count messages.</p> <p>When the logging level is set to three (3), the Event Router records all errors that occur during these events.</p> |
| 4 | <p>Event Router entities communicate with external APIs, for example WebSphere MQ</p> <p>When the logging level is set to four (4), the Event Router records all errors that occur during these events.</p> |

Log messages - details

Log messages are records of Event Router processing events. This topic groups the log messages according to these events, as follows:

| Messages | Codes |
|---|---------|
| Connecting to networks on page 64 | 001-109 |
| Initializing, enabling, and disabling Event Router targets on page 70 | 110-124 |
| Initializing and stopping Event Router entities on page 77 | 200-211 |
| Connecting to Event Router targets on page 80 | 300-310 |
| Sending messages on page 85 | 320-333 |
| Receiving messages on page 90 | 400-412 |

| Messages | Codes |
|--|---------|
| WebSphere MQ (formerly MQSeries) Interface messages on page 95 | 500-507 |
| Executing commands on page 99 | 800-860 |
| Detailing errors on page 102 | 901-906 |

Connecting to networks

Each time the Event Router receives a message from a source application or sends a message to an target, the Event Router uses a network connection. The following log messages provide details about these network connections.

- [001 PROCESS E COMMER on page 64](#)
- [002 PROCESS I INITDONE on page 65](#)
- [003 PROCESS I PORT_FIL on page 65](#)
- [004 PROCESS I CONN_INF on page 66](#)
- [005 PROCESS E SHUTDOWN on page 66](#)
- [006 PROCESS E TCPNOTUP on page 66](#)
- [007 PROCESS E TCPAGAIN on page 66](#)
- [101 PROCESS I CONN_REQ on page 67](#)
- [102 PROCESS I CONN_IND on page 67](#)
- [103 PROCESS I CONN_CONF on page 67](#)
- [104 PROCESS I CONN_RESP on page 68](#)
- [105 PROCESS I DISC_REQ on page 68](#)
- [106 PROCESS I DISC_IND on page 68](#)
- [107 PROCSS E CONN_MAX on page 69](#)
- [108 PROCESS E CONN_SSL on page 69](#)
- [109 PROCESS E CONN_SSL on page 69](#)

001 PROCESS E COMMER

| | |
|-------------|---|
| Description | XXXX The description of this message depends on the relevant network interface and operating system. |
|-------------|---|

| | |
|---------------|--|
| Details | When the Event Router tried to initialize a network interface, an error occurred. |
| Action | Refer to the user documentation for the relevant network interface and operating system. |
| Logging level | Zero |

002 PROCESS / INITDONE

| | |
|---------------|--|
| Description | Initialized interface <i>name1</i> SAP <i>nn1</i> MaxConn <i>nn2</i> Device <i>name2</i> LocalAddress <i>addr</i> |
| Details | <p>The Event Router initialized a network interface In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>name1</i>: the network interface • <i>nn1</i>: the TCP/IP port that the Event Router uses to receive messages • <i>nn2</i>: the maximum number of simultaneous connections that the Event Router can maintain for receiving messages • <i>name2</i>: the name of the software or hardware for the network interface. When this information does not apply to your platform, no name is displayed. • <i>addr</i>: the local address of the Event Router |
| Action | None |
| Logging level | Zero |

003 PROCESS / PORT_FIL

| | |
|---------------|--|
| Description | Cannot use file <i>name</i> |
| Details | The file (<i>name</i>) containing information about the Event Router TCP/IP port control is unusable. |
| Action | This message is preceded by one or more messages that provide details about the error. You can also consult the error file for additional details. If the file indicated is invalid, modify the parameter in the General Parameter file and execute the setup operation. |
| Logging level | Zero |

004 PROCESS I CONN_INF

| | |
|---------------|---|
| Description | Active connections <i>nn</i> |
| Details | Indicates the current number of active connections <i>nn</i> managed by the Event Router. |
| Action | None, this message is returned by the DETAILS command. |
| Logging level | Zero |

005 PROCESS E SHUTDOWN

| | |
|---------------|---|
| Description | Stopping interface |
| Details | z/OS only. An error caused the TCPIP interface to be reinitialized. |
| Action | None, the interface will be restarted. |
| Logging level | Zero |

006 PROCESS E TCPNOTUP

| | |
|---------------|---|
| Description | Interface not started |
| Details | z/OS only. The TCPIP interface is temporarily unavailable. All current connection are closed and any new connection is refused. |
| Action | None |
| Logging level | Zero |

007 PROCESS E TCPAGAIN

| | |
|---------------|---|
| Description | Interface restarted |
| Details | z/OS only. The TCPIP interface is again available. Connection to target will be establish if any message is available. New incoming connections are accepted. |
| Action | None |
| Logging level | Zero |

101 PROCESS / CONN_REQ

| | |
|---------------|---|
| Description | (nn) Sent connection request: from <i>addr1</i> to <i>addr2</i> |
| Details | <p>The Event Router requested to connect to target In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>nn</i>: the connection number• <i>addr1</i>: the TCP/IP address of the Event Router• <i>addr2</i>: the address of the target |
| Action | None |
| Logging level | One |

102 PROCESS / CONN_IND

| | |
|---------------|---|
| Description | (nn) Received connection request <i>addr1</i> from <i>addr2</i> |
| Details | <p>An source application requested to connect to the Event Router In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>nn</i>: the connection number• <i>addr1</i>: the TCP/IP address of the Event Router• <i>addr2</i>: the TCP/IP address of the source application |
| Action | None |
| Logging level | One |

103 PROCESS / CONN_CONF

| | |
|-------------|---|
| Description | (nn) Sent acknowledgement for a connection request to <i>addr</i> |
| Details | <p>The Event Router sent a positive acknowledgement for a connection request. In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>nn</i>: the connection number• <i>addr</i>: the TCP/IP address of the source application |
| Action | None |
| Logging | level One |

104 PROCESS / CONN_RESP

| | |
|-------------|---|
| Description | (nn) Received acknowledgement for a connection request from <i>addr</i> |
| Details | <p>The Event Router received a positive acknowledgment for a connection request from a target.</p> <p>In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>nn</i>: the connection number • <i>addr</i>: the address of the target |
| Action | None |
| Logging | level One |

105 PROCESS / DISC_REQ

| | |
|-------------|---|
| Description | (nn) Requesting to disconnectXXXX |
| Details | <p>The Event Router requested to close a network connection. In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>nn</i>: the connection number • <i>XXXX</i>: details about why the Event Router requested to close the network connection |
| Action | None |
| Logging | level One |

106 PROCESS / DISC_IND

| | |
|---------------|--|
| Description | (nn) Received request to disconnect XXXX |
| Details | <p>The Event Router] received a request to close a connection. In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>nn</i>: the connection number • <i>XXXX</i>: details about why the Event Router received a request to close the network connection |
| Action | None |
| Logging level | One |

107 PROCSS E CONN_MAX

| | |
|---------------|---|
| Description | (0) Too many connections active (nn), CONN_REQ Rejected Loading Target Parameters File |
| Details | <p>The Event Router received a request to establish a connection. The connection was rejected because there are already too many active connections.</p> <ul style="list-style-type: none"> nn indicates the current number of active connections. <p>The maximum number of sessions accepted by the Event Router is defined in the configuration (see incoming_max parameter in the General Parameters example on page 23) and is displayed at startup in message 002 INITDONE.</p> |
| Action | Verify if the maximum must be modified. If the current maximum is already important, verify if the applications always keep their connection active and/or if the parameter indicates how long the Event Router can keep active connections (timeout parameter). |
| Logging level | 0 |

108 PROCESS E CONN_SSL

| | |
|---------------|---|
| Description | (nn) Cannot Create SSL Environment Profile (xxxxxxx) for YYYYYY |
| Details | <p>The Event Router received a request to open an SSL connection but the ER is not able to create the environment. In the following order, this message identifies:</p> <ul style="list-style-type: none"> nn: the connection number XXXX: name of the security profile YYYYY: Destination address |
| Action | Check the previous error messages in the LOG and/or in the Event Router process output. Correct the error and retry. |
| Logging level | 0 |

109 PROCESS E CONN_SSL

| | |
|-------------|---|
| Description | (nn) Cannot Create SSL Comm Environment |
| Details | <p>The Event Router received a connection indication request but cannot establish an SSL environment:</p> <ul style="list-style-type: none"> nn: the connection number |

| | |
|---------------|---|
| Action | Check the previous error messages in the LOG and/or in the Event Router process output. Correct the error and retry. |
| Logging level | 0 |

Initializing, enabling, and disabling Event Router targets

When you initialize, enable, and disable an Event Router target, the relevant entity records the following log messages.

- [110 PROCESS I CONF_LOAD on page 70](#)
- [111 PROCESS I CONF_OK on page 71](#)
- [112 PROCESS E CONF_NOK on page 71](#)
- [113 PROCESS I CONF_TYPB on page 71](#)
- [114 PROCESS I CONF_TYPE on page 72](#)
- [115 PROCESS I CONF_ACT on page 72](#)
- [116 PROCESS E CONF_ACTE on page 72](#)
- [117 PROCESS E SRV_FILER on page 73](#)
- [118 PROCESS I SERV_FILE on page 74](#)
- [119 PROCESS I CONF_DEL on page 74](#)
- [120 PROCESS I CONF_DIS on page 74](#)
- [121 PROCESS I CONF_ENA on page 75](#)
- [122 PROCESS I CONF_STA on page 75](#)
- [123 PROCESS I CONF_END on page 1](#)
- [124 PROCESS CONF_BAK on page 1](#)

110 PROCESS I CONF_LOAD

| | |
|---------------|---|
| Description | Loading Target Parameters File |
| Details | The relevant Event Router entity is loading the Target Parameters File. |
| Action | None |
| Logging level | zero |

111 PROCESS I CONF_OK

| | |
|---------------|--|
| Description | Loaded Target Parameters File |
| Details | The relevant Event Router entity finished analyzing the Target Parameter File and is sending details about the file to the other entities. |
| Action | None |
| Logging level | zero |

112 PROCESS E CONF_NOK

| | |
|---------------|--|
| Description | Error loading Target Parameters File (Code <i>code</i>) |
| Details | <p>When the relevant Event Router entity analyzed the Target Parameters File, an error occurred. In this message, code is one of the following:</p> <ul style="list-style-type: none"> • 4: The format of the XML message is not correct. • 7: The XML message does not match the DTD. <p>This message is preceded by one or more messages that provide details about the error. You can also consult the error file for additional details.</p> |
| Action | Correct the format of the Target Parameters File, and then reload the file via the loadconfig command. |
| Logging level | Zero |

113 PROCESS I CONF_TYPB

| | |
|---------------|--|
| Description | Initializing Target: Protocol <i>protocol</i> |
| Details | <p>The relevant Event Router entity is initializing the interface for one or more targets. In this message, <i>protocol</i> identifies one of the following interfaces:</p> <ul style="list-style-type: none"> • HTTP • QLT • MQOUT |
| Action | None |
| Logging level | Zero |

114 PROCESS I CONF_TYPE

| | |
|---------------|--|
| Description | Initialized Target: Protocol <i>protocol</i> |
| Details | <p>The relevant Event Router entity initialized the interface for one or more targets. In this message, <i>protocol</i> identifies one of the following interfaces:</p> <ul style="list-style-type: none">• HTTP• QLT• MQOUT |
| Action | None |
| Logging level | Zero |

115 PROCESS I CONF_ACT

| | |
|---------------|---|
| Description | Initialized Target <i>name</i> |
| Details | <p>The relevant Event Router entity initialized a target. In this message, <i>name</i> identifies the name of the initialized target.</p> |
| Action | None |
| Logging level | Zero |

116 PROCESS E CONF_ACTE

| | |
|-------------|--|
| Description | Error initializing Target <i>name</i> (Code <i>code</i>) |
|-------------|--|

| | |
|---------------|---|
| Details | <p>When the relevant Event Router entity initialized a target, an error occurred. The Event Router cannot send messages to the target. In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>name</i>: the name of the target• <i>code</i>: one of the following codes:<ul style="list-style-type: none">◦ 1: Cannot initialize the target environment◦ 2: Error in the overflow or batch file◦ 4: Internal error◦ 6: Cannot use the specified send mode (HTTP, QLT) <p>In the log file, preceding messages may provide additional details. For more information about the error, consult the error file.</p> |
| Action | Correct the error, and then reinitialize the Event Router target via the reset command. |
| Logging level | Zero |

117 PROCESS E SRV_FILER

| | |
|-------------|---|
| Description | Targetname Cannot use file path (Codecode) |
| Details | <p>When the relevant Event Router entity tried to open the overflow file or batch file, an error occurred. In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>name</i>: the name of the Event Router target• <i>path</i>: the path to the overflow or batch file• <i>code</i>: one of the following codes:<ul style="list-style-type: none">◦ -1: Error in the overflow or batch file◦ -2: An internal error closed the overflow or batch file◦ -3: Cannot send one or more messages from the overflow or batch file◦ -4: Cannot initialize or create the overflow or batch file◦ -5: Cannot enable one or more targets. The Event Router rejects all messages for the target.◦ -6: Cannot initialize the overflow or batch file. The file contains data the Event Router did not save. Alternatively, the Event Router did not create the file.◦ -7: Cannot store one or more messages in the overflow or batch file. The file is full. <p>In the Log File, preceding messages may provide additional details. For more information about the error, consult the error file.</p> |

| | |
|---------------|--|
| Action | When this message displays the code <ul style="list-style-type: none">• four (4), verify that you created the relevant overflow or batch file in Sentinel.• six (6), verify that you created the relevant overflow or batch file in Sentinel. After you correct the error, reinitialize the overflow or batch file via the reset command. |
| Logging level | Zero |

118 PROCESS / SERV_FILE

| | |
|---------------|---|
| Description | Target <i>name</i> File <i>path</i> contains <i>nn</i> messages |
| Details | You initialized an Event Router target. The relevant Event Router entity counted the number of messages that are saved in the overflow and batch files. In the following order, this message identifies: <ul style="list-style-type: none">• <i>name</i>: the name of the target• <i>path</i>: the path to the overflow or batch file• <i>nn</i>: the number of messages in the overflow or batch file |
| Action | None |
| Logging level | Three |

119 PROCESS / CONF_DEL

| | |
|---------------|---|
| Description | Deleted Target <i>name</i> |
| Details | The relevant Event Router entity is deleting the configuration of an Event Router target from the Target Parameters File. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | Zero |

120 PROCESS / CONF_DIS

| | |
|-------------|------------------------------------|
| Description | Disabled Target <i>name</i> |
|-------------|------------------------------------|

| | |
|---------------|--|
| Details | You executed the disable command. The relevant Event Router entity is disabling an Event Router target. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | Zero |

121 PROCESS / CONF_ENA

| | |
|---------------|--|
| Description | Enabling Target <i>name</i> |
| Details | You executed the enable command. The relevant Event Router entity is enabling an Event Router target. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | Zero |

122 PROCESS / CONF_STA

| | |
|-------------|---|
| Description | Target <i>name</i> : Status <i>status</i> Type <i>type</i> |
|-------------|---|

| | |
|---------------|--|
| Details | <p>You executed the display command. This message displays the status of an Event Router target.</p> <p>In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>name</i>: the name of the target• <i>status</i>: one of the following statuses:<ul style="list-style-type: none">◦ DISABLE: You executed the disable command. The target cannot receive messages.◦ ENABLE: You executed the enable command. The target can receive messages.◦ ERROR: When you initialized the target, an error occurred.◦ INIT: You executed the enable or the reset command. The target is being initialized.◦ FULL: The buffer file is full.◦ <i>type</i>: indicates the type of protocol and, in brackets, target connection information:<ul style="list-style-type: none">◦ OLT [TCPIP address]◦ MQOUT [<i>WebSphere MQ manager name, queue name</i>] |
| Action | <p>When the status of an Event Router target is:</p> <ul style="list-style-type: none">• ERROR, use messages in the log and error files to correct the error, and then reinitialize the target via the reset command.• INIT and this status persists, check that the Event Router entity that manages the target is running• FULL, check the preceding message to identify and correct the cause of the connection failure. Then, execute the force command. Most common failure cause: server not started. |
| Logging level | Zero |

123 PROCESS / CONF_END

| | |
|---------------|--|
| Description | All Targets Ready |
| Details | Indicates the load of the target configuration is completed. |
| Action | None, this message is returned at started or following a LOADCONFIG command. |
| Logging level | Zero |

124 PROCESS CONF_BAK

| | |
|---------------|---|
| Description | Target name: Backup (<i>backup address</i>) |
| Details | Complete the message 122 when a backup address is defined for the target. |
| Action | None, this message is returned by the DETAILS command. |
| Logging level | Zero |

Initializing and stopping Event Router entities

To monitor whether or not the Event Router correctly initialized and stopped entities, use the following log messages.

- [200 PROCESS I INIT_OK](#) on page 77
- [201 PROCESS E INIT_NOK](#) on page 78
- [202 PROCESS I TERM_OK](#) on page 78
- [203 PROCESS I INIT](#) on page 78
- [204 PROCESS I PRC_FROZE](#) on page 78
- [205 PROCESS I PRC_RESET](#) on page 79
- [206 PROCESS E INIT_ERR](#) on page 79
- [210 PROCESS I PRC_STOP](#) on page 79
- [211 PROCESS I REQ_STOP](#) on page 80
- [212 PROCESS I PRD_STOP](#) on page 80

200 PROCESS I INIT_OK

| | |
|---------------|---|
| Description | Initialized process |
| Details | The Event Router initialized the relevant entity. |
| Action | None |
| Logging level | Zero |

201 PROCESS E INIT_NOK

| | |
|---------------|---|
| Description | Error initializing process (Codecode) |
| Details | <p>The Event Router cannot initialize the relevant entity. In the current version of Sentinel, the <i>code</i> for this message is always one (1).</p> <p>In the log file, preceding messages may provide additional details. For more information about the error, consult the error file.</p> |
| Action | <p>When the DISP entity cannot load the Target Parameters File, correct the Target Parameters File. Then, execute the loadconfig command.</p> <p>Otherwise, stop all Event Router entities, and then correct the error.</p> |
| Logging level | Zero |

202 PROCESS I TERM_OK

| | |
|---------------|---|
| Description | Stopping process |
| Details | The Event Router is stopping the relevant entity. |
| Action | None |
| Logging level | Zero |

203 PROCESS I INIT

| | |
|---------------|--|
| Description | Beginning to initialize process |
| Details | The Event Router is beginning to initialize the relevant entity. |
| Action | None |
| Logging level | Zero |

204 PROCESS I PRC_FROZE

| | |
|-------------|---------------------------------|
| Description | Temporarily stop process |
|-------------|---------------------------------|

| | |
|---------------|---|
| Details | Following a pause command, the Event Router stops the relevant entity and frees external resources. |
| Action | Use the reset command to resume processing. |
| Logging level | Zero |

205 PROCESS I PRC_RESET

| | |
|---------------|---|
| Description | Restart process |
| Details | Following a restart command, the Event Router is resuming the relevant entity. |
| Action | None |
| Logging level | Zero |

206 PROCESS E INIT_ERR

| | |
|---------------|--|
| Description | <i>error message</i> |
| Details | Information message about the error. The Event Router cannot initialize the relevant entity. The entity will stop with the additional message 201. For more information about the error, consult the error file. |
| Action | Stop all Event Router entities, and then correct the error. |
| Logging level | Zero |

210 PROCESS I PRC_STOP

| | |
|---------------|---|
| Description | Event Router being stopped |
| Details | Information indicating the Event Router is stopping, either during an operator command or due to an internal stop command (initialization error). All commands are rejected . |
| Action | None |
| Logging level | Zero |

211 PROCESS I REQ_STOP

| | |
|---------------|--|
| Description | Request <i>request</i> Rejected, stopping |
| Details | The command is rejected as the Event router is stopping. |
| Action | None |
| Logging level | None |

212 PROCESS I PRD_STOP

| | |
|---------------|---|
| Description | Error, Stopping Event Router |
| Details | Following an error during entity initialization, an internal stop command is generated. For more information about the error, consult the log and error file. |
| Action | Correct the error and restart the Event Router |
| Logging level | Zero |

Connecting to Event Router targets

In the following log messages, the SNTL and TIVO entities record their attempts to open and close connections to Event Router targets.

- [300 PROCESS I SRV_CON on page 81](#)
- [301 PROCESS I SRV_CONT on page 81](#)
- [302 PROCESS E SRV_CONE on page 81](#)
- [303 PROCESS I SRV_CONA on page 82](#)
- [304 PROCESS I SRV_CND on page 82](#)
- [305 PROCESS I SRV_COUNT on page 82](#)
- [306 PROCESS I SRV_CONR on page 83](#)
- [307 PROCESS I SRV_CONR on page 83](#)
- [308 PROCESS I SRV_CONF on page 84](#)
- [309 PROCESS I SRV_CREQ on page 84](#)
- [310 PROCESS I SRV_INFO on page 85](#)

300 PROCESS I SRV_CON

| | |
|---------------|---|
| Description | Connected to target <i>name</i> [(<i>backup</i>)] |
| Details | The relevant Event Router entity successfully connected to a target. In this message, <i>name</i> identifies the name of the target. (<i>backup</i>) indicates that the connection is done using the target backup address. |
| Action | None |
| Logging level | Two |

301 PROCESS I SRV_CONT

| | |
|---------------|---|
| Description | Disconnected from target <i>name</i> [(<i>backup</i>)] |
| Details | The relevant Event Router entity closed the connection to an target. In this message, <i>name</i> identifies the name of the target. (<i>backup</i>) indicates that the connection was establish using the target backup address. |
| Action | None |
| Logging level | Two |

302 PROCESS E SRV_CONE

| | |
|---------------|---|
| Description | Failure connecting to target <i>name</i> [(<i>backup</i>)] |
| Details | The relevant Event Router entity cannot connect to a target. In this message, <i>name</i> identifies the name of the target. (<i>backup</i>) indicates that target backup address is use. |
| Action | When the logging level of the relevant Event Router entity is set to four, preceding messages in the log file may provide additional details. For more information about the error, consult the error file. |
| Logging level | Zero |

303 PROCESS / SRV_CONA

| | |
|---------------|---|
| Description | Reestablished connection to target <i>name</i> [(<i>backup</i>)] |
| Details | After several unsuccessful attempts to connect to an Event Router target, the relevant Event Router entity successfully connected to the target. In this message, <i>name</i> identifies the name of the target. (<i>backup</i>) indicates that the connection is done using the target backup address. |
| Action | None |
| Logging level | Zero |

304 PROCESS / SRV_CND

| | |
|---------------|--|
| Description | Matched send conditions for target <i>name</i> |
| Details | The condition for sending a batch of messages to an Event Router target is true. The relevant Event Router entity is sending messages from the batch file to the target. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | One |

305 PROCESS / SRV_COUNT

| | |
|-------------|---|
| Description | Target <i>name</i>: Messages sent <i>nn</i>, saved (<i>overflow</i>, <i>batch</i>)[, <i>backup mm</i>] |
| Details | <p>Via the count command, you displayed the message counters for one or, alternatively all Event Router targets. In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>name</i>: the name of the target • <i>nn</i>: the number of messages sent to the target (this includes the number of messages sent on backup address, if defined). Each time you initialize the Event Router, the Event Router resets this counter to zero. • <i>overflow</i>: the number of messages in the overflow file • <i>batch</i>: the number of messages in the batch file • <i>,backup mm</i>: Present if backup address is set. <i>mm</i> is the number of messages sent using the backup address. |

| | |
|---------------|--|
| Action | When the overflow file contains messages, consult the preceding messages in the log file. These messages provide details about why the Event Router did not send the messages. Otherwise, no action is required. |
| Logging level | Zero |

306 PROCESS / SRV_CONR

| | |
|---------------|--|
| Description | Connection for session <i>number</i> rejected <i>reason</i> |
| Details | <p>An error occurred during the initial protocol exchange between the target of a request and the server. The message appears on both sides. This message displays the following information:</p> <ul style="list-style-type: none">• <i>number</i>: the number of the network session• <i>reason</i>: identifies the reason for rejection by displaying one of the following codes:<ul style="list-style-type: none">◦ 6 - Invalid or no data◦ 10 - Target identifier is not defined on the server site The server does not retry the connection. |
| Action | The target definition is not equivalent at both sites, or an unknown program has tried to establish a connection. In the first case, modify the target parameter on the requester site and then use the loadconfig command to reload the target parameter file. |
| Logging level | Zero |

307 PROCESS / SRV_CONR

| | |
|---------------|--|
| Description | Connection for target <i>target</i>: rejected <i>reason</i>: code |
| Details | <p>The destination target rejected the connection due to a error in the protocol exchange. This message displays the following information:</p> <ul style="list-style-type: none">• <i>target</i>: target name• <i>reason</i>: identifies the reason for rejection by displaying the following code:<ul style="list-style-type: none">◦ Invalid or no data (=timeout) |
| Action | <p>The target definition is not equivalent at both sites, or an unknown program has tried to establish a connection.</p> <p>In the first case, modify the target timeout parameter on the requester site and then use the loadconfig command to reload the target parameter file.</p> |
| Logging level | Zero |

308 PROCESS / SRV_CONF

| | |
|---------------|---|
| Description | Disconnecting from target <i>name</i>: reason code: |
| Details | <p>The destination target server rejected the connection. This message displays the following information:</p> <ul style="list-style-type: none"> • <i>name</i>: the name of the target server • <i>reason</i>: identifies the disconnection reason by displaying one of the following codes: <ul style="list-style-type: none"> ◦ 11 - no messages ready to be deliver ◦ 12 - the target status temporarily does not allow the exchange of messages The requester site will initiate a connection retry later, according to its parameters. This message appears on both the Event Router and the target server site. |
| Action | If the code is 12, check the trace messages on the server site, or consult the error file for additional details. Correct any errors on the server site. Then, on the requester site, execute the force command, or wait for the automatic retry. |
| Logging level | Two |

309 PROCESS / SRV_CREQ

| | |
|---------------|---|
| Description | Target <i>name</i>: Received messages <i>nn</i>, Rejected messages <i>nnn</i>, [, backup <i>mm</i>) |
| Details | <p>Result of the count command for a QLT Requester target. In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>name</i>: the name of the target server • <i>nn</i>: the number of messages retrieved by the target server (this includes the number of messages sent on backup address, if defined). Each time you initialize the Event Router, the Event Router resets this counter to zero. • <i>nnn</i>: the number of messages that the Event Router was not able to process • <i>,backup mm</i>: Present if backup address is set. <i>mm</i> is the number of messages sent using the backup address. |
| Action | None |
| Logging level | Zero |

310 PROCESS I SRV_INFO

| Description | Current Targets <i>nn</i> Cumulative Msg <i>nnn</i> |
|---------------|---|
| Details | <p>Indicates the number of target servers served by the Event Router, and the number of messages processed since the last time the target configuration was loaded. In the following order, this message identifies:</p> <ul style="list-style-type: none"> • <i>nn</i>: number of targets • <i>nnn</i>: the number of messages processed since the last time the target configuration was loaded |
| Action | None |
| Logging level | Zero |

Sending messages

When the SNTL and TIVO entities send messages to Event Router targets, they can record the following log messages.

- [320 PROCESS I MSG_SND](#) on page 85
- [321 PROCESS I MSG_SOK](#) on page 86
- [322 PROCESS E MSG_SERR](#) on page 86
- [323 PROCESS W MSG_SAVE](#) on page 86
- [324 PROCESS E MSG_NSAV](#) on page 87
- [325 PROCESS E DB_FULL](#) on page 87
- [327 PROCESS I MSG_SERR](#) on page 88
- [328 PROCESS I MSG_MQERR](#) on page 88
- [329 PROCESS I MSG_TIME](#) on page 88
- [330 PROCESS I PRO_QDAT](#) on page 89
- [331 PROCESS I PRO_QMSG](#) on page 89
- [332 PROCESS I PRO_QPRO](#) on page 89
- [333 PROCESS I PRO_RESP](#) on page 90

320 PROCESS I MSG_SND

| Description | Sent message to target <i>name</i> |
|-------------|---|
|-------------|---|

| | |
|---------------|---|
| Details | The relevant Event Router entity sent a message to a target. In this message, <i>name</i> identifies the <i>name</i> of the target. |
| Action | None |
| Logging level | Three |

321 PROCESS I MSG_SOK

| | |
|---------------|--|
| Description | Received acknowledgement from target <i>name</i> |
| Details | The Event Router entity received an acknowledgment for a message from an target. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | Three |

322 PROCESS E MSG_SERR

| | |
|---------------|---|
| Description | Received no acknowledgment or negative acknowledgment <i>code</i> from target <i>name</i> |
| Details | <p>The relevant Event Router entity sent a message to a target. However, the entity received no acknowledgment or a negative acknowledgment.</p> <p>The entity saves the message in the overflow or batch file. After the waiting period, the entity tries to send the message again.</p> <p>In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>code</i>: the code for the negative acknowledgment. This code depends on the type of target.• <i>name</i>: the name of the target |
| Action | Refer to the relevant server documentation. |
| Logging level | Zero |

323 PROCESS W MSG_SAVE

| | |
|-------------|--|
| Description | Cannot send message to target <i>name</i>. Saved message. |
|-------------|--|

| | |
|---------------|--|
| Details | The relevant Event Router entity was not able to send a message to an target. The entity saved the message in the relevant overflow or batch file. In this message, <i>name</i> identifies the name of the target. |
| Action | For details about why the Event Router entity did not send the message, consult the preceding messages in the log file. |
| Logging level | Three |

324 PROCESS E MSG_NSAV

| | |
|---------------|---|
| Description | Cannot save message for target <i>name</i> |
| Details | <p>The relevant Event Router entity was not able to send a message to a target. The entity was also not able to save the message in the overflow or batch file. The Event Router rejects any additional messages for the target.</p> <p>As long as the connection to the target is maintained, the entity saves the message in computer memory. After the waiting period, the entity tries to send the message again. In this message, <i>name</i> identifies the name of the target.</p> |
| Action | <p>For details about why the Event Router did not send the message, consult the preceding messages in the log file. After you correct the error, use the reset command to reinitialize the target.</p> <p>If necessary, stop and restart the Event Router. In this case, the Event Router clears the stored message from computer memory.</p> |
| Logging level | Zero |

325 PROCESS E DB_FULL

| | |
|-------------|--|
| Description | File is full for target <i>name</i> |
| Details | <p>The overflow or batch file for the relevant Event Router target is full. This file cannot store a new message. The Event Router rejects any additional messages for the target.</p> <p>As long as the connection to the target is maintained, the Event Router saves the message in computer memory. After the waiting period, the Event Router tries to send the message again.</p> <p>In this message, <i>name</i> identifies the name of the target.</p> |

| | |
|---------------|---|
| Action | To send messages from the overflow or batch file, use the force command. You can also configure the overflow or batch file to store more messages. If you stop the Event Router, the Event Router will clear the stored message from computer memory. To load the new configuration without stopping the Event Router, execute the loadconfig command. |
| Logging level | Zero |

327 PROCESS I MSG_SERR

| | |
|---------------|--|
| Description | Error formatting message for Target <i>name</i> (Code <i>nn</i>) |
| Details | The relevant Event Router entity cannot format a message for a target. In the following order, this message identifies the name of the target. |
| Action | Refer to the target configuration in the Target Parameters File In the log file, preceding messages may provide additional details. For more information about the error, consult the error file. |
| Logging level | Zero |

328 PROCESS I MSG_MQERR

| | |
|---------------|--|
| Description | Insertion for target <i>name</i> (Code <i>WebSphere code</i>, Reason <i>WebSphere code</i>) |
| Details | The relevant Event Router entity cannot insert a message in WebSphere queue. |
| Action | Check the <i>return</i> and reason codes in the WebSphere MQ manuals. Locate the destination queue manager name and queue name in the target parameter file or in the output trace file message. |
| Logging level | Zero |

329 PROCESS I MSG_TIME

| | |
|-------------|---|
| Description | No data received in time allowed for <i>TargetName</i> |
| Details | After a requester target establishes a connection to the Event Router, the target receives no data before timer expiration. |

| | |
|---------------|---|
| Action | Either the definition of the target does not correspond between requester and server sites, or an unknown program is trying to establish a connection. For the first case, modify the target parameter on the requester site (time out parameter) and execute the loadconfig command to reload the target parameter file. |
| Logging level | Zero |

330 PROCESS / PRO_QDAT

| | |
|---------------|---|
| Description | Invalid data received for target <i>name</i> |
| Details | A protocol error has occurred during the message exchange. |
| Action | Either the definition of the target does not correspond between requester and server sites, or an unknown program is trying to establish a connection. For the first case, modify the target parameter on the requester site (time out parameter) and execute the loadconfig command to reload the target parameter file. |
| Logging level | Zero |

331 PROCESS / PRO_QMSG

| | |
|---------------|--|
| Description | Received message for target <i>name</i> |
| Details | Indicates that a message from the server target has arrived and will be processed. |
| Action | None |
| Logging level | Three |

332 PROCESS / PRO_QPRO

| | |
|---------------|---|
| Description | Sent protocol data for target <i>name</i> |
| Details | Indicates that protocol information has been exchanged between the Event Router and the target. |
| Action | None |
| Logging level | Three |

333 PROCESS I PRO_RESP

| Description | Message processed by dispatcher for target <i>name</i> , <i>code</i> |
|---------------|--|
| Details | <p>The message 331 was received and was processed by the Event Router dispatcher. A response will be sent to the server target.</p> <p>If the code is not 0, the connection is closed and connection retry conditions apply. For the list of codes, meanings and actions, refer to log message 403.</p> |
| Action | <p>If the code is not zero, check the previous message, or consult the error file for additional details. Take the appropriate action based on the returned code (refer to log message 403).</p> <p>After correcting the error, either:</p> <ul style="list-style-type: none"> • execute the force command on the requester site, or alternatively • wait for the automatic retry |
| Logging level | Zero |

Receiving messages

When the DISP entity receives messages from a source application, this entity can record the following log messages.

- [400 PROCESS I MSG_RCV](#) on page 90
- [401 PROCESS I MSG_ROK](#) on page 91
- [402 PROCESS I MSG_RERR](#) on page 91
- [403 PROCESS E LSG_RDAT](#) on page 91
- [404 PROCESS E MSG_RXML](#) on page 92
- [405 PROCESS I MSG_INFO](#) on page 93
- [410 PROCESS I LGR_COUNT](#) on page 94
- [411 PROCESS I LGR_GET](#) on page 94
- [412 I LGR_RESP](#) on page 94

400 PROCESS I MSG_RCV

| Description | Received message |
|-------------|--|
| Details | The relevant Event Router entity received a message from a source application. |

| | |
|---------------|-------|
| Action | None |
| Logging level | Three |

401 PROCESS I MSG_ROK

| | |
|---------------|---|
| Description | Received message. Sending acknowledgement |
| Details | The relevant Event Router entity received and accepted a message from a source application. Keep in mind that the received message can be an exception. |
| Action | None |
| Logging level | Three |

402 PROCESS I MSG_RERR

| | |
|---------------|---|
| Description | Rejected message. Sending negative acknowledgment (Code <i>code</i>) |
| Details | <p>The relevant Event Router entity received a message from a source application. However, the entity rejected the message and sent a negative acknowledgement. In the current version of Sentinel, this message always contains the code three (3).</p> <p>In the log file, preceding messages may provide additional details. For more information about the error, consult the error file.</p> |
| Action | Correct the error at the source application. |
| Logging level | Zero |

403 PROCESS E LSG_RDAT

| | |
|-------------|---|
| Description | Error analyzing message: XXXX (Code <i>code</i>) |
|-------------|---|

| | |
|---------------|--|
| Details | <p>The relevant Event Router entity received a message from a source application. However, the message generated an error.</p> <p>For all codes except 4 and 7 (see below), the entity returns a negative acknowledgement to the source application, which allows it to send the message again after the problem is resolved.</p> <p>For codes 4 and 7, to prevent the source application from resending the message, the entity sends a positive acknowledgment to the source application. If you have configured the Event Router to route exceptions, the entity routes the message to the target for Exceptions.</p> <p>In the following order, this message provides:</p> <ul style="list-style-type: none">• <i>XXXX</i>: details about why the message generated an error• <i>code</i>: one of the following codes:<ul style="list-style-type: none">◦ 1 NOT READY: At least one of the targets is not enabled.◦ 2 NOT FOUND: The entity cannot find at least one of the targets.◦ 3 FORMAT ERR: The name of at least one target is not correctly formatted.◦ 4 XML ITS FORMAT: The XML format of the message does not match the DTD.◦ 5 TOO MANY TARGETS: The message identifies too many targets. The maximum number of targets is ten.◦ 6 NO DEFAULT FOUND: The message does not identify a target, and no default target is defined.◦ 7 XML PARSING ERROR: The XML format of the message is not correct.◦ 8 PROCESS NOT READY: The entity cannot receive the message. The Event Router is initializing the entity or updating the Target Parameters File. |
| Action | <p>Correct the error at the source application.</p> <p>If the code is 7 XML PARSING ERROR and the Event Router is not configured to route Exceptions, you can modify the configuration in the Target Parameters File.</p> |
| Logging level | Zero |

404 PROCESS E MSG_RXML

| | |
|-------------|---|
| Description | Invalid XML format: XXXX (Code code) |
|-------------|---|

| | |
|---------------|---|
| Details | <p>The relevant Event Router entity received a message from an target. However, the format or content of the message is invalid. To prevent the source application from resending the message, the entity sends a positive acknowledgment to the source application.</p> <p>If the Event Router is:</p> <ul style="list-style-type: none">• not configured to route Exceptions, the entity rejects the message• configured to route Exceptions, the entity routes the message to the target for Exceptions <p>In the following order, this message provides:</p> <ul style="list-style-type: none">• <i>XXXX</i>: details about the invalid message• <i>code</i>: one of the following codes:<ul style="list-style-type: none">◦ 4: The format of the XML message is not correct.◦ 7: The XML message does not match the DTD. <p>In the log file, preceding messages may provide additional details. For more information about the error, consult the error file.</p> |
| Action | <p>Correct the error at the source application, and then resend the message.</p> <p>If the Event Router is not configured to route Exceptions, you can modify the configuration in the Target Parameters File.</p> |
| Logging level | Zero |

405 PROCESS / MSG_INFO

| | |
|---------------|--|
| Description | Messages accepted, rejected, processed |
| Details | <p>As a return result of the details command, this message displays statistical information for the Event Router dispatcher entity (DISP), in the following order:</p> <ul style="list-style-type: none">• accepted: number of messages received and analyzed without errors from source applications and targets (MQSeries and servers)• rejected: number of messages received and analyzed with errors from source applications and targets (MQSeries and servers)• processed: number of message created, analyzed and then sent to a target |
| Action | None |
| Logging level | Zero |

410 PROCESS / LGR_COUNT

| | |
|---------------|---|
| Description | State <i>status</i> ; Messages processed <i>processed</i> Ignored <i>ignored</i> |
| Details | <p>Indicates relevant Event Router process state. Where:</p> <ul style="list-style-type: none"> processed = Number of messages the relevant Event Router entity has retrieved from the logger file and sent for processing by the dispatcher. ignored = The number of messages that cannot be processed by the dispatcher. status = One of the following Entity states: <ul style="list-style-type: none"> INIT: Initializing in progress ERROR: Problem with initialization or File Access. See previous log message for cause of problem FROZEN: Result of a PAUSE command READY: Logger file opened, waiting to execute a new scan PROC MSG: Message retrieved and sent to dispatcher. Waiting response STOPPING: Command stop being processed |
| Action | None. |
| Logging level | Zero |

411 PROCESS / LGR_GET

| | |
|---------------|--|
| Description | Retrieve message from logger |
| Details | The relevant Event Router entity read a message from the logger file |
| Action | None. |
| Logging level | Three |

412 / LGR_RESP

| | |
|-------------|--|
| Description | Message processed by dispatcher, code <i>code</i> |
|-------------|--|

| | |
|---------------|---|
| Details | Information message about the response send by the dispatcher. See message 403 for a list of the code value. If the value is from 4 to 7, the message is ignored, otherwise the message is resent at next logger scan. |
| Action | For the values other than 4 to 7, correct the error (see message 403). |
| Logging level | Three |

WebSphere MQ (formerly MQSeries) Interface messages

Entities that use the Event Router WebSphere MQ API can record the following log messages.

- [500 PROCESS I MQI_GET](#) on page 95
- [501 PROCESS I MQI_RESP](#) on page 95
- [502 PROCESS W MQI_RETRY](#) on page 96
- [503 PROCESS E MQI_RETRY](#) on page 96
- [504 PROCESS I MQI_RETBG](#) on page 96
- [505 PROCESS I MQI_RETED](#) on page 97
- [506 PROCESS I MQI_INFO](#) on page 97
- [507 PROCESS I MQI_AVAIL](#) on page 98

500 PROCESS I MQI_GET

| | |
|---------------|---|
| Description | Retrieve message from WebSphere MQ |
| Details | The relevant Event Router entity is reading a message from a WebSphere queue. |
| Action | None |
| Logging level | Three |

501 PROCESS I MQI_RESP

| | |
|-------------|---|
| Description | Message processed by dispatcher (Code <i>code</i>) |
|-------------|---|

| | |
|---------------|--|
| Details | The DISP entity has handled the message retrieved from a WebSphere queue. The code in this message is any of the possible values listed for the message 403. If the code is zero, the Event Router removes the message from WebSphere MQ to the retry queue for later processing. |
| Action | If the code is not zero, see message 403 and correct the error. The Event Router processes the message again later. |
| Logging level | Three |

502 PROCESS W MQI_RETRY

| | |
|---------------|--|
| Description | Message put in retry queue (<i>retry queue name</i>) |
| Details | Following a problem, the DISP entity rejects the message. The Event Router inserts the message in the retry queue. |
| Action | See message 403 for the error reason and correct the error. The Event Router processes the message again later. |
| Logging level | Four |

503 PROCESS E MQI_RETRY

| | |
|---------------|--|
| Description | No more space available in <i>queue name</i> |
| Details | WebSphere MQ has rejected the insert message queue request (MQPUT). No queue space is available. The relevant Event Router entity suspends message management until space is again available. |
| Action | Increase the storage limits at the WebSphere MQ (for example, boost the <i>maximum message in queue</i> parameter). |
| Logging level | Zero |

504 PROCESS I MQI_RETBG

| | |
|-------------|---|
| Description | Beginning routing message <i>from message queue name</i> |
|-------------|---|

| | |
|---------------|--|
| Details | The retry queue timer has expired, the relevant Event Router entity takes the messages from the retry queue and inserts them in the message queue. |
| Action | None |
| Logging level | Three |

505 PROCESS / MQI_RETED

| | |
|---------------|---|
| Description | End routing message from <i>message queue name</i> |
| Details | The relevant Event Router entity routes all the messages from the retry queue to the message queue. |
| Action | None |
| Logging level | Three |

506 PROCESS / MQI_INFO

| | |
|-------------|---|
| Description | State (<i>status</i>) ; Msg: read (<i>processed</i>), retry (<i>retries</i>), error (<i>errors</i>) |
|-------------|---|

| | |
|---------------|---|
| Details | <p>Indicates relevant Event Router process state. Where:</p> <ul style="list-style-type: none"> • <i>processed</i> = number of messages the relevant Event Router entity has retrieved from the queue and sent for processing by the dispatcher. • <i>retries</i> = the number of messages the relevant Event Router entity has taken from the retry queue and inserted in the queue • <i>errors</i> = the number of messages rejected by the dispatcher that the relevant Event Router entity has inserted in the retry queue • <i>status</i> = One of the following Entity states: <ul style="list-style-type: none"> ◦ INIT: Initializing in progress ◦ ERROR: Problem with initialization or Queue manager. See previous log message for cause of problem ◦ FROZEN: result of a PAUSE command ◦ DISCONNECT: Not connected to queue manager. Try again when the retry timer has expired. See previous log message for cause of problem ◦ CONNECT: Connected to Queue manager but working queue not open. Try again when the retry timer has expired. See previous log message for cause of problem ◦ WAIT: Connected to Queue manager and working queue opened. Wait for messages to be processed ◦ PROC MSG Message retrieved and sent to dispatcher. Waiting response |
| Action | If status is not WAIT or PROC MSG, see status explanation. Otherwise, if <i>retries</i> and <i>errors</i> increase significantly, the error causing the message to be rejected is not solved. Execute the action required for message 403. |
| Logging level | Zero |

507 PROCESS / MQL_AVAIL

| | |
|---------------|--|
| Description | Cannot process messages, retry later |
| Details | The relevant Event Router entity has either failed to connect to MGR or to open the working queue. The entity starts the retry timer and sets status to DISCONNECT or CONNECT. |
| Action | See previous log message for cause of problem. |
| Logging level | Zero |

Executing commands

When you execute an Event Router command, the relevant entity can record the following log messages.

- [800 PROCESS I CMDINFO](#) on page 99
- [801 PROCESS W CMDNFND](#) on page 99
- [802 PROCESS W CMDISTA](#) on page 100
- [803 PROCESS E CMDNLOG](#) on page 100
- [804 PROCESS I CMDLIST](#) on page 100
- [805 PROCESS W CMDNLST](#) on page 100
- [806 PROCESS W CMDISTA](#) on page 101
- [807 PROCESS U CMDPROC](#) on page 101
- [808 PROCESS W CMDOSTA](#) on page 101
- [850 PROCESS W CMDFROZ](#) on page 102
- [860 PROCESS W CMDNOLOG](#) on page 102

800 PROCESS I CMDINFO

| | |
|---------------|--|
| Description | <i>command</i> |
| Details | The relevant Event Router entity received a command. |
| Action | None |
| Logging level | Zero |

801 PROCESS W CMDNFND

| | |
|---------------|---|
| Description | Cannot find target <i>name</i> |
| Details | The relevant Event Router entity cannot find the target that is specified in a command. In this message, <i>name</i> identifies the name of the specified target. |
| Action | To see a list of available targets, execute the display command. Then, reenter the relevant command. |
| Logging level | Zero |

802 PROCESS W CMDISTA

| | |
|---------------|--|
| Description | Command invalid for TargetStatus <i>name status</i> |
| Details | <p>Due to the status of an Event Router target, the relevant entity cannot execute a command. In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>name</i>: the name of the target• <i>status</i>: the status of the target |
| Action | None |
| Logging level | Zero |

803 PROCESS E CMDNLOG

| | |
|---------------|---|
| Description | Invalid context for target <i>name</i> |
| Details | An internal error occurred for an Event Router target. In this message, name identifies the name of the target. |
| Action | Contact Axway. |
| Logging level | Zero |

804 PROCESS I CMDLIST

| | |
|---------------|--|
| Description | End of list |
| Details | You executed a command that displayed a series of messages. This message marks the end of this series. |
| Action | None |
| Logging level | Zero |

805 PROCESS W CMDNLST

| | |
|-------------|---|
| Description | Cannot execute command. No target is specified |
|-------------|---|

| | |
|---------------|---|
| Details | Because no Event Router target is specified in a command, the relevant entity cannot execute the command. |
| Action | To see a list of available targets, execute the display command. Then, reenter the relevant command. |
| Logging level | Zero |

806 PROCESS W CMDISTA

| | |
|---------------|--|
| Description | Command <i>name</i> not supported by Entity |
| Details | The relevant Event Router entity does not support the command you entered. |
| Action | Reenter the command including the correct entity name. |
| Logging level | Zero |

807 PROCESS U CMDPROC

| | |
|---------------|--|
| Description | Command <i>name</i> is being processed |
| Details | The Event Router has begun processing the command. |
| Action | None |
| Logging level | Zero |

808 PROCESS W CMDOSTA

| | |
|---------------|---|
| Description | Command Invalid, Target <i>name</i> required status <i>status name</i> |
| Details | The command entered cannot currently be processed. The command destination target does not have the status necessary to handle the command. For example, a purge command can only be handled by a target with the status frozen. |
| Action | Use a command to modify the target status. For the above example, execute the froze command on the target. |
| Logging level | Zero |

850 PROCESS W CMDFROZ

| | |
|---------------|--|
| Description | Saving messages for target <i>name</i> |
| Details | You executed the freeze command for an Event Router target. The relevant entity is saving all messages for the target in the overflow file or, alternatively the batch file. In this message, <i>name</i> identifies the name of the target. |
| Action | None |
| Logging level | Zero |

860 PROCESS W CMDNOLOG

| | |
|---------------|---|
| Description | Target <i>login</i> not authorized. Cannot execute command |
| Details | The relevant Event Router entity cannot execute the command sent by the Sentinel Server. The login information is missing or incorrect. |
| Action | Correct the command on the server. |
| Logging level | Zero |

Detailing errors

When an error occurs, the Event Router can provide details about the error via the following log messages.

- [901 PROCESS I TCPINFO on page 102](#)
- [902 PROCESS I FI1INFO on page 103](#)
- [903 PROCESS I XMLINFO on page 103](#)
- [904 PROCESS E MQSERR on page 103](#)
- [905 PROCESS I MQCONN on page 104](#)
- [906 PROCESS E MQSOK on page 104](#)

901 PROCESS I TCPINFO

| | |
|-------------|---|
| Description | This message depends on the relevant error |
|-------------|---|

| | |
|---------------|--|
| Details | This message details an error that exists in the TCP/IP network and provides recommendations about how to correct the error. In the log file, preceding messages may provide additional details and recommendations. For more information about the error, consult the error file. |
| Action | Consult the other messages in the log and error files. |
| Logging level | Zero |

902 PROCESS I FI1INFO

| | |
|---------------|--|
| Description | This message depends on the relevant error. |
| Details | This message contains details about an error that exists in an overflow or batch file. In the log file, preceding messages may provide additional details and recommendations. For more information about the error, consult the error file. |
| Action | Consult the other messages in the log and error files. |
| Logging level | Zero |

903 PROCESS I XMLINFO

| | |
|---------------|---|
| Description | This message depends on the relevant error |
| Details | This message contains details about an error that exists in an XML message. In the log file, preceding messages may provide additional details and recommendations. For more information about the error, consult the error file. |
| Action | Consult the other messages in the log and error files. |
| Logging level | Zero |

904 PROCESS E MQSERR

| | |
|-------------|--|
| Description | <i>(type)(name)</i> Fails <i>(return code, reason code)</i> |
|-------------|--|

| | |
|---------------|--|
| Details | <p>The WebSphere MQ request has failed.</p> <p>In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>type</i>: the API request type, for example:<ul style="list-style-type: none">◦ MQGET◦ MQOPEN◦ ...• <i>name</i>: the name of the object for which the request was done• <i>return code</i>: WebSphere MQ return code• <i>reason code</i>: WebSphere MQ reason code |
| Action | Check the return and reason code in the WebSphere MQ manuals. if necessary, consult the other messages in the log and error files. |
| Logging level | Zero |

905 PROCESS I MQCONN

| | |
|---------------|---|
| Description | Connection to qmgr <i>name</i> successful |
| Details | The relevant Event Router entity has successfully established a connection with the queue manager named in the message. |
| Action | None |
| Logging level | Zero |

906 PROCESS E MQSOK

| | |
|-------------|--|
| Description | (<i>type</i>)(<i>name</i>) Successful |
| Details | <p>The WebSphere MQ request was successful.</p> <p>In the following order, this message identifies:</p> <ul style="list-style-type: none">• <i>type</i>: the API request type, for example:<ul style="list-style-type: none">◦ MQGET◦ MQOPEN◦ ...• <i>name</i>: the name of the object for which the request was done |

11 Log messages

| | |
|--------|------|
| Action | None |
|--------|------|

| | |
|---------------|------|
| Logging level | Four |
|---------------|------|

In a log file, the Event Router notifies you about Event Router errors. For more technical details about the errors you find in the log file, consult the error traces. The Event Router records error traces in a set of error files that include one file for each Event Router process. The names and locations of these error files depend on the host platform:

- UNIX, OS/400 and Windows: The name of each error file is `Process.out`, where `Process` is the name of an Event Router process. These files are stored in the `tmp` subdirectory for the product. Each time you reinitialize the Event Router, the Event Router erases the contents of the error files.
- z/OS: The DDname of each error file is `TRACEnumber`, where `number` is a number that identifies an Event Router process in `SYSOUT`.

Each error trace includes:

- The date and time of the relevant error
- The label **ERROR from** *function_name*, where *function_name* is the name of the function that generated the relevant error.
- One or more messages that provide details about the relevant error. Each message includes the date and time that are associated with the relevant details.

The following is an example of an error trace:

| Date | Time | Label | One or more messages |
|------------|----------|----------------------|----------------------|
| 2002/12/01 | 16:10:04 | ERROR from NetIoSync | |
| 2002/12/01 | 16:10:04 | processing IOSYNC | |

Cipher suites

13

Sentinel Event Router supports the cipher suites listed below, and prioritizes them as displayed in the Order used column; that is, the Event Router overrides your cipher suite order. The order, between two approximate levels of security, favors the cipher suite that provides a better level of performance.

| Suite | Order used | Authentication Confidentiality / Integrity | Minimum TLS version |
|-------|------------|--|---------------------------|
| 49199 | 6 | ECDHE_RSA_WITH_ AES_128_GCM_ SHA256 | 1.2 |
| 49200 | 7 | ECDHE_RSA_WITH_ AES_256_GCM_ SHA384 | 1.2 |
| 49191 | 8 | ECDHE_RSA_WITH_ AES_128_CBC_SHA256 | 1.2 |
| 49192 | 9 | ECDHE_RSA_WITH_ AES_256_CBC_SHA384 | 1.2 |
| 156 | 10 | TLS_AES_128_GCM_ SHA256 | 1.2 |
| 157 | 11 | TLS_AES_128_GCM_ SHA384 | 1.2 |
| 60* | 12 | TLS_RSA_WITH_AES_ 128_CBC_SHA256 | 1.2 |
| 61* | 13 | TLS_RSA_WITH_AES_ 256_CBC_SHA256 | 1.2 |
| 47 | 14 | TLS_RSA_WITH_AES_ 128_CBC_SHA | 1.2 |

| Suite | Order used | Authentication Confidentiality / Integrity | Minimum TLS version |
|-------|------------|--|---------------------------|
| 53 | 15 | TLS_RSA_WITH_AES_256_CBC_SHA | 1.2 |
| 10 | 16 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | 1.0 |
| 5 | 17 | TLS_RSA_WITH_RC4_128_SHA | 1.0 |
| 4 | 18 | TLS_RSA_WITH_RC4_128_MD5 | 1.0 |
| 59* | 19 | TLS_RSA_WITH_NULL_SHA256 | 1.2 |
| 2 | 20 | TLS_RSA_WITH_NULL_SHA | 1.0 |
| 1 | 21 | TLS_RSA_WITH_NULL_MD5 | 1.0 |
| 9 | 22 | RSA_WITH_DES_CBC_SHA | 1.0 |

Note * To comply with security standards, the use of the cipher suites 59, 60, and 61 is restricted to TLS 1.2 exclusively.

Appendix A: AGTCRYPT utility

This utility allows you to crypt confidential information used by Event Router. It is currently used to encrypt the password of the private key associated to the X.509 user certificate.

The key used for the operations is derived from a password and is built with the following characteristics:

- The key size is 128 bits.
- Cipher algorithm is AES/CBC/PKCS5Padding.
- Secure random number generator is SHA1PRNG.
- The mac algorithm is HmacSHA256.

This utility supports the following operations:

- Generate a key.
- Renew the key (Future use).
- Encrypt data.
- Decrypt previously encrypted data.

To enable the Sentinel Event Router Server to access encrypted data, the key file generated (see the next paragraph) must be added in the configuration of the Event Router.

Generate the encryption key

Execute the following command to generate the key:

```
agtcrypt --genkey --pass PASSWORD --keyfname FILENAME --saltfname  
FILENAME
```

The agtcrypt parameters are:

| Parameter | Description |
|----------------------|---|
| --pass <PASSWORD> | A valid password. The password must be at least 8 characters long, contain upper and lower case characters, at least one numeric character, and contain at least one of the special characters @\$%\$. The password that will be used to generate the encryption key. |

| Parameter | Description |
|---------------------------|---|
| --keyfname <FILENAME> | The key filename. The location where the encryption key will be stored. You must have read and write permissions for this location. |
| --saltfname <FILENAME> | The salt filename. The location where the salt file will be stored. You must have read and write permissions for this location. |

Encrypt the certificate password

Execute the following command to encrypt data:

```
agtcrypt -encrypt --text P12_PASSWORD --keyfname FILENAME -textfname  
FILENAME
```

The agtcrypt parameters are:

| Parameter | Description |
|------------------------------|--|
| --text <P12_ PASSWORD> | The certificate password. |
| --keyfname <FILENAME> | The key filename. The location where the encryption key is stored. You must have read permissions for this location. |
| -- --textfname <FILENAME> | The encrypted password filename. The location where the certificate password will be stored. You must have read and write permissions for this location. |

Decrypt an encrypted password

Execute the following command to decrypt the data previously encrypted and passed in either the data (--text parameter) or the file (--textfname parameter):

```
agtcrypt -decrypt --keyfname FILENAME --text PASSWORDDATA | -textfname  
FILENAME
```

The agtcrypt parameters are:

| Parameter | Description |
|--------------------------|---|
| --keyfname <FILENAME> | The key filename. The location where the encryption key is stored. You need to have read permissions for this location. |

| Parameter | Description |
|--|--|
| <code>--text <DATA></code> | Encrypted Data as written in the file. |
| <code>---textfname <FILENAME></code> | The file containing the encrypted data. You need to have read permissions for this location. |

Renew the key (future use)

Execute the following command to re-generate the key:

```
agtcrypt --renewkey --pass PASSWORD --keyfname FILENAME --saltfname  
FILENAME --oldpass PASSWORD
```

The agtcrypt parameters are:

| Parameter | Description |
|---|--|
| <code>--pass <PASSWORD></code> | A valid password. The password must be at least 8 characters long, contain upper and lower case characters, at least one numeric character, and contain at least one of the special characters @#\$%. The password that will be used to generate the encryption key. |
| <code>--oldpass PASSWORD</code> | The password use to generate the key file the first time. |
| <code>--keyfname <FILENAME></code> | The key filename. The location where the old encryption key was stored and will contain the new key. You must have read and write permissions for this location. |
| <code>--saltfname <FILENAME></code> | The salt filename. The location where the salt file will be stored. You must have read and write permissions for this location. |

Modify the Event Router configuration files

To enable the Event Router to use the encrypted data:

- Give the user starting the Event Router read permission to the key file and encrypted data files.
- Add the key file in the configuration parameter file.
- Regenerate the configuration.

To add the Key file name to the configuration parameter file:

- Edit the General Parameters file
 - `conf/trkagent.ini`
 - or for z/OS member `userparin`
- In the section `[AGENT]`
 - Add the keyword `encrypt_key_file = keyfname`

To regenerate the configuration (see additional information in the platform installation guide):

- `agtinst -setup`
- or for z/OS : `JCL D20SETUP`

Notes

- Do not forget to change the information in the Security Profile.
- Modify the value of the `SSL_USER_CERTIFICATE_PASSWORD_FILE` by the file in which you have encrypted the certificate password.

z/OS Installation

A new JCL (D50CRYPT) is provided that contain the statement necessary to execute the `genkey` and `encrypt` operations. The steps to install this JCL are explained in the Readme file.

Example

The following example shows the operations on a Unix platform:

1. Execute the Event Router profile:

```
. EventRouter/conf/profile
```
2. Generate the encryption key:

```
agtcrypt --genkey --keyfname $trk_home_dir/conf/crypt.key --  
saltfname $trk_home_dir/conf/crypt.salt --pass <PASSWORD>
```
3. Encrypt the certificate password:

```
agtcrypt --encrypt --keyfname $trk_home_dir/conf/crypt.key --  
text <P12_password> --textfname $trk_home_dir/conf/user.pw
```
4. Modify the Event Router General Parameter file `conf/trkagent.ini`:

```
[AGENT]  
  
encrypt_key_file="$trk_home_dir/conf/crypt.key"
```
5. Regenerate the configuration:

```
agtinst -setup
```
6. Modify the Security profile:


```
[ERSERVER]
```

```
SSL_USER_CERTIFICATE_FILE = $trk_home_dir/conf/user.p12
```

```
SSL_USER_CERTIFICATE_FORMAT = PKCS12
```

```
SSL_USER_CERTIFICATE_PASSWORD_FILE = $trk_home_dir/conf/user.pw
```

The sample uses files in the Event Router directory. It is recommended that all files related to the encryption are in a separate directory and protected with strict read/write permission.