Monitoring Parasoft Virtualize Server Events

This topic explains how to configure monitoring for events that occur on Parasoft Virtualize Servers.

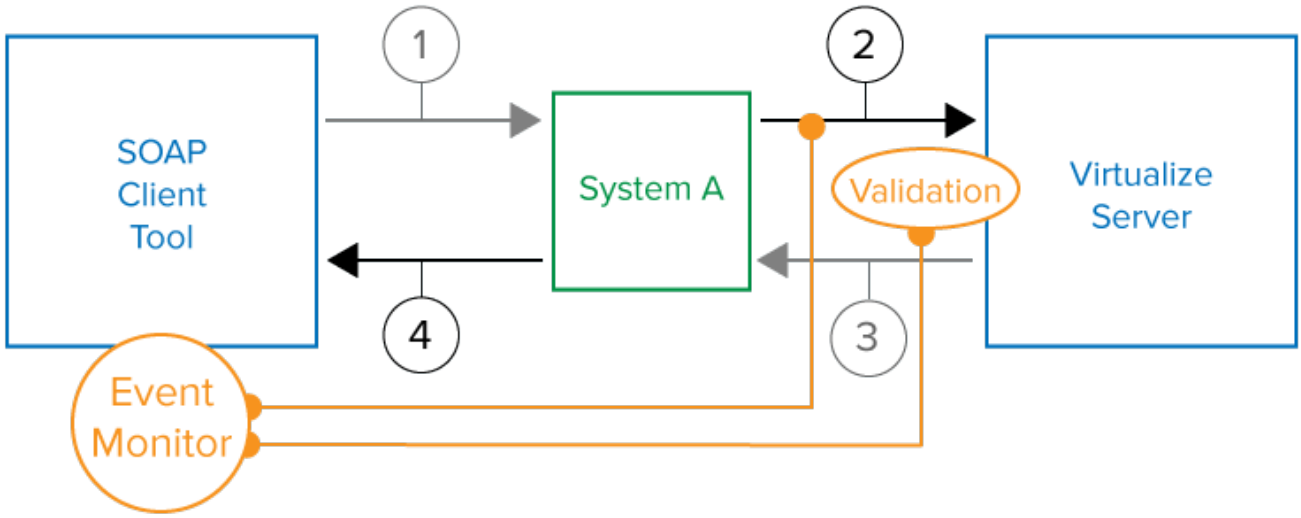
Sections include:

Why Monitor Server Events?

Visibility into what messages are sent to and from Parasoft Virtualize Servers—and what validations and errors occur at the virtual asset level—enables you to:

- Validate the messages that are sent to your Parasoft servers and the virtual assets deployed on them.
- See what errors occur.

For example, in a common test situation, SOAtest will send a message to a system, such as an available service or a web browser, which will then send a message to a virtual asset deployed on a Parasoft Virtualize server (e.g., because the actual resources is not yet available or is not accessible for testing).



By adding an Event Monitor tool to the test suite, you gain insight into messages 2 and 3 and well as messages 1 and 4. You also see the result of any validation tools you may have attached to the virtual asset (for instance, XML Assertor tools) and receive details on any errors that may have occurred (for instance, because the virtual asset was not properly configured to process valid messages, or because invalid messages were sent).

Using an Alternative JMS System for Virtualize Server Event Monitoring

By default, the Virtualize event monitoring service uses a built-in provider based on ActiveMQ. For details on how to use another provider, see U sing an Alternative JMS System for Virtualize Server Event Monitoring.

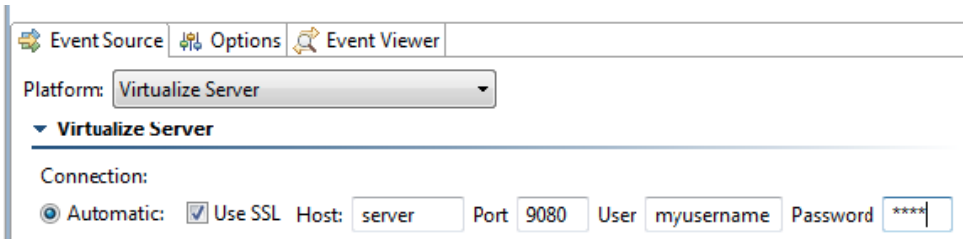
Configuration

Add an Event Monitor tool to the test suite that drives the interaction with the Virtualize Server you want to monitor. Click the **Event Source** tab of tool's configuration panel and choose **Virtualize Server** from the Platform drop-down menu and configure the following settings.

Connection Settings

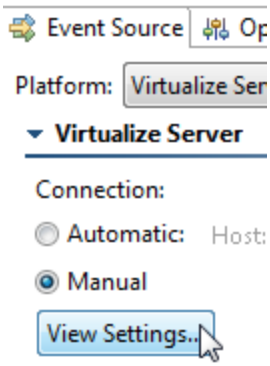
If you are connecting to a Virtualize Server 9.6 or newer and require user authentication:

1. Enable the **Automatic** connection option and complete the **User** and **Password** fields.
2. You may also need to adjust the settings for **SSL**, **Host**, and **Port**.



If you are connecting to a Virtualize Server 9.3 or older that uses a port other than 1080:

1. Enable the **Manual** connection option and click **View Settings**.



2. Provide the following OpenJMS connection settings:

URL	rmi://hostname:portnumber/
Initial Context	org.exolab.jms.jndi.InitialContextFactory
Connection Factory	ConnectionFactory
Destination Name	SOATESTSERVER_STUB_EVENTS
Destination Type	Topic

Event Subscriptions

Under **Event Subscriptions**, enable the type of events you want to monitor.

Request messages	The message sent to a virtual asset (e.g., message #2 in the diagram under Why Monitor Server Events?)
Response messages	The message that the virtual asset returns (e.g., message #3 in the diagram under Why Monitor Server Events?)
Message validation results	The result of any validation tools you may have attached to the virtual asset—such as XML Assertor tools (e.g., the tool is represented by the Validation marker in the diagram under Why Monitor Server Events?).
Errors events	Any errors that may have occurred, such as if virtual assets were not properly configured to process valid messages or because invalid messages were sent.In the diagram under Why Monitor Server Events?, if the virtual asset was not configured to route message #2 to a specific responder, this would be reported as an error. This also includes events from custom extentions that are designated to be at the INFO level.

Info events	Informational events, such as server startup and shutdown events as well as events from custom extentions that are designated to be at the INFO level.
Debug events	Events from custom extensions that are designated to be at the DEBUG level.
Warn events	Events from custom extensions that are designated to be at the WARN level.

Test Failure Criteria

Specify the test failure criteria in the **Test Failure Criteria** section. If a validation tool is chained to the current Event Monitor tool, the following settings not applicable because the outcome of the chained validation will determine this test's success or failure.

Error events	The test will fail if any errors occur (e.g., because responders were not properly configured to process valid messages, or because invalid messages were sent).
Validation failure events	The test will fail if a failure is reported by any validation tool (such as an XML Assertor tool) you may have attached to a responder.
No events received	The test will fail if the virtual asset does not receive any events before the current test suite (the one that includes the Event Monitor tool) completes execution.

Event Monitoring Options

Click the **Options** tab, modify settings as needed.

Clear the event viewer before each event monitor run	Enable this option to automatically clear the Event Monitor event view (both text and graphical) whenever Event Monitor starts monitoring.
Include test execution events in the XML event output to chained tools	Enable this option to show only the monitored messages and events in the Event Viewer tab and XML output display. This option also indicates when each test started and completed. Enabling this option is helpful if you have multiple tests in the test suite and you want to better identify the events and correlate them to your test executions.
Wrap monitored messages with CDATA to ensure well-formedness of the XML event output	Enable this option if you do not expect the monitored events' message content to be well-formed XML. Disabling this option will make the messages inside the events accessible via XPaths, allowing the message contents to be extracted by XML Transformer or validated with XML Assertor tools. Enable this option if the message contents are not XML. This ensures that the XML output of the Event Monitor tool (i.e., the XML Event Output for chaining tools to the Event Monitor, not what is shown under the Event Viewer) is well-formed XML by escaping all the message contents. This will make the content of these messages inaccessible by XPath since the message technically becomes just string content for the parent element. The Diff tool's XML mode supports string content that is XML. As a result, the Diff tool will still be able to diff the messages as XML, including the ability to use XPaths for ignoring values, even if this option is disabled.
Maximum time to wait for the monitor to start (milliseconds)	Specify the maximum length of time the Event Monitor should wait to finish connecting to the event source before SOAtest runs the other tests in the suite. This enables SOAtest to capture events for those tests and prevents SOAtest from excessively blocking the execution of the other tests if the Event Monitor is having trouble connecting to its event source. Increase the value if connecting to the event source takes more time than the default. The default is `3000`.
Maximum monitor execution duration (milliseconds)	Specify the point at which the test should timeout if, for example, another test in the test suite hangs or if no other tests are being run (e.g., if you execute the Event Monitor test apart from the test suite, then use a custom application to send messages to system).
Event polling delay after each test finishes execution (milliseconds)	This field is not applicable.

Event Viewer

The **Event Viewer** tab will display details about the events received. It indicates the virtual asset name, the name of responder that processed that message, the response message, results of validation tools (if available), and errors that occurred. Double-clicking an item opens a dialog with additional details.

Troubleshooting: No Events or Specific Error Messages are Reported

If you no events or specific error messages are reported by Event Monitor, then disable your firewall. Firewalls running where SOAtest is running can sometimes block communication between the Event Monitor and a remote Virtualize Server. If you are using the Windows firewall, it needs to be disabled prior to using Event Monitor with a remote Virtualize Server. Other firewalls may also need to be disabled.

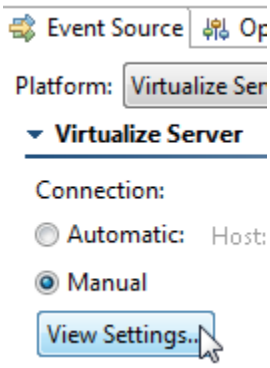
Using an Alternative JMS System for Virtualize Server Event Monitoring

By default, Virtualize Server events are monitored using the built-in ActiveMQ-based provider. Alternatively, you can use another JMS system that you have. To configure this:

1. Open the view for the Parasoft Server you want to monitor (e.g. for a Parasoft Virtualize server, go to Parasoft Virtualize, and choose **Window> Show View> Virtualize Server**).
2. Double-click the node for the machine (local or remote) you want to configure to use the event monitoring provider.
3. In the **Event Monitoring Provider** field, select your preferred server. If you want to use a JMS server that is not specifically listed, choose **Other JMS Provider**.
4. Specify the connection settings.

> **Event Monitoring Destination - Configuration Needed**
>
> Note that a default event monitoring destination and type are specified in the available controls.
>
> You need to either:
>
> - Configure your JMS system to use this default destination, or
>
> - Change the Parasoft settings to another destination that is available on your system.

5. In the Event Monitor tool configuration panel:
    a. Open the **Event Source** tab.
    b. Set connection to **Manual**
    c. Click **View Settings**.



    d. Select the appropriate **Event Monitoring Provider**.
    e. Specify the settings required to connect to your JMS.

For details on the connection settings, see

- Monitoring IBM WebSphere ESB
- Monitoring Oracle or BEA AquaLogic Service Bus
- Monitoring Software AG webMethods Broker
- Monitoring Sonic ESB
- Monitoring TIBCO EMS
- Monitoring Other JMS Systems