Monitoring Other JMS Systems

This topic explains how to configure monitoring for any JMS system. Double-click the Event Monitor tool to open up the tool configuration panel and configure the following settings to monitor any JMS system.

Configuring the Event Source

Platform	Choose **Generic JMS system** from the platform drop-down menu.
Connection	Specify your JMS connection settings.
Monitoring Source	In the **Initial Context** field, specify a fully-qualified class name string passed to the JNDI javax.naming.InitialContext constructor as a string value for the property named javax.naming.Context.INITIAL_CONTEXT_FACTORY. In the **Connection Factory** field, specify the JNDI name for the factory. This is passed to the lookup() method in javax.naming.InitialContext to create a javax.jms.QueueConnectionFactory or a javax.jms.TopicConnectionFactory instance. In the **Destination Name** field, specify the topic or queue that you want to monitor. You can specify a regular topic or queue (e.g., the entry or exit of a workflow process), or a special process tracking topic. In the **Destination Type** field, specify whether the tracking destination is a topic or a queue. (Optional) In the **Message Selector** field, enter a value to act as a message filter. See Using Message Selector Filters for tips. Enable to the **Leave messages on the queue** option if you want SOAtest to use the JMS QueueBrowser API to trace messages posted on a JMS queue without removing them from the queue. This allows SOAtest to gain visibility into these messages without impacting the transaction. **Leave messages on the queue** For a discussion of potential complications with this option—and how to avoid them—see JMS Queue Options.
JNDI Properties	If you want any additional JNDI properties applied to this deployment, specify them in the JNDI properties table.

Configuring Event Monitoring Options

Click the **Options** tab and 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.