

Monitoring Sonic ESB

This topic explains how to configure monitoring for Sonic ESB. Sections include:

- [Adding Required Jar Files to the SOAtest Classpath](#)
- [Configuring the Event Source](#)

Adding Required Jar Files to the SOAtest Classpath

The following jar files need to be added to the SOAtest classpath:


- broker.jar
- mfcontext.jar
- sonic_Client.jar

The files are located in the [SONIC_MQ_INSTALL]/MQ<version>/lib directory by default. To add these jar files to SOAtest's classpath, complete the following:

1. Choose **Parasoft> Preferences**.
2. Open the **Parasoft> System Properties** page.
3. Click the **Add JARS** button and choose and select the necessary JAR files to be added.

Configuring the Event Source

Double-click the **Event Monitor** tool to open up the tool configuration panel. Click the **Event Source** tab and specify the following:

Platform	Choose Sonic Enterprise Service Bus from the platform drop-down menu.
Connection	Specify your Sonic ESB connection settings.
Monitoring Source	<p>Specify the connection factory. Default is TopicConnectionFactory</p> <p>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 "dev.Tracking" tracking endpoint. For instance, if you want to track all events that occur as part of the process flow, specify the dev.Tracking endpoint, and have the process set to Tracking Level of 4 in the ESB.</p> <p>Choose Queue or Topic from the Destination Type drop-down menu.</p> <p>(Optional) In the Message Selector field, enter a value to act as a message filter. See Using Message Selector Filters for tips.</p> <p>Enable 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.</p> <div style="border: 1px solid #ccc; padding: 5px;"><p> Leave messages on the queue</p><p>For a discussion of potential complications with this option—and how to avoid them—see JMS Queue Options.</p></div>
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.

<p>Wrap monitored messages with CDATA to ensure well-formedness of the XML event output</p>	<p>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 XPath, allowing the message contents to be extracted by XML Transformer or validated with XML Assertor tools.</p> <p>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.</p> <p>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 XPath for ignoring values, even if this option is disabled.</p>
<p>Maximum time to wait for the monitor to start (milliseconds)</p>	<p>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.</p>
<p>Maximum monitor execution duration (milliseconds)</p>	<p>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).</p>
<p>Event polling delay after each test finishes execution (milliseconds)</p>	<p>This field is not applicable to Sonic ESB.</p>