

Monitoring Oracle or BEA AquaLogic Service Bus

This topic explains how to configure monitoring for events that are transmitted through Oracle Service Bus (OSB) or BEA Aqualogic Service Bus (ALSB). Sections include:

- [Service Bus Configuration](#)
- [SOAtest Configuration](#)

Service Bus Configuration

1. Ensure that **Message Reporting** is enabled. This is required so SOAtest can draw message events. For details on how to globally enable message reporting in the bus, refer to the [OSB Console Guide](#).
2. Add Message Reporting actions to the desired message workflow components (see [Modeling Message Flow in Oracle Service Bus in the Oracle documentation](#)). See [Proxy Service: Action in the Oracle documentation](#) for details on how to accomplish this.

SOAtest Configuration

Adding Required Jar Files to the SOAtest Classpath

By default, OSB is configured to use the built-in PointBase relational Database for Message Reporting purposes. Parasoft SOAtest uses the OSB message reporting framework to obtain and visualize events (intermediate messages) from the bus by executing SQL queries on the reporting database.

Configuring the Event Source

Double-click the **Event Monitor**. Click the **Event Source** tab and configure the following settings.

- Choose **ORACLE Service Bus** from the platform drop-down menu.
- If you have a default OSB configuration, then add the PointBase JDBC driver to the SOAtest classpath. This is in a single jar found in your OSB /WebLogic installation directory: `${BEA_HOME}/wlserver_10.* /common/eval/pointbase/lib/pbclient5*.jar` You need to use the pbclient51.jar for ALSB 3.0 and pbclient57.jar for OSB 10gR3 (each ships with its own jar).
- If your OSB is configured to use a different database, then you need to provide the database JDBC drivers to the SOAtest classpath.
- Set the Event Monitor URL based on the database. For more information, see [Database Configuration Parameters](#).

Configuring Event Monitoring Options (Recommended)

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	<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>
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)

Specify how long the monitor should wait before obtaining the events. For OSB, we recommend setting to three (3) seconds or longer. By making the event monitor wait for a few seconds before obtaining the events, you can ensure that the events have been logged to the database before the query is executed.

Sharing Configuration Settings

You can click **Export Configuration Settings** to export these configuration settings to a file, then other team members can reference the settings by selecting the **File** button and specifying the path to this file.

▼ OSB Message Reporting Database Configuration

File

Input file: