Monitoring IBM WebSphere ESB

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

WebSphere Configuration

IBM WebSphere ESB includes monitoring capabilities that build upon its underlying WebSphere Application Server. Parasoft SOAtest can subscribe to Common Base Events that are fired at points in the processing of service components, and which are managed by WebSphere Common Event Infrastructure (CEI).

For information about monitoring service component events in the WebSphere ESB and enabling the monitoring using WebSphere administrative console, see the IBM Monitoring service component events topic.

To configure WebSphere for event monitoring:

1. Enable the CEI service in the ESB.
2. Choose the level of logging for the service components you are interested in. The steps for performing this task on the ESB can be found at the IBM Configuring service component event monitoring using the administrative console topic.
    - In order to get the full event details in SOAtest, we recommend that you select the "ALL MESSAGES AND TRACES" option and the "FINEST" logging level for the components you are interested in, and which results in the business messages being included in the CEI events. To enable that for all business integration components, the log level string in the WebSphere administrative console would look like this:

      ```
      *=info: WBILocationMonitor.CEI.SCA.com.*=finest
      ```

SOAtest Configuration

Adding Required Jar Files to the SOAtest Classpath

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

- com.ibm.ws.ejb.thinclient_7.0.0.jar
- com.ibm.ws.orb_7.0.0.jar
- com.ws.sib.client.thin.jms_7.0.0.jar
- com.ibm.ws.emf_2.1.0.jar

The jar files can be found under [WAS installation dir]/runtimes.

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 the tool configuration panel. Click the **Event Source** tab and specify the following:

Platform	Choose **IBM WebSphere Enterprise Service Bus** from the Platform drop-down menu.
Connection	Specify your ESB connection settings. The connection URL is the JNDI InitialContext URL for the WebSphere Default JMS provider. The port number is the WebSphere bootstrap port. You can check the correct port number for your WebSphere ESB using the administrative console under Servers section, WebSphere Application Server, then click or expand the "Ports" link under the "Communication" section. The port number to use in SOAtest is the BOOTSTRAP_ADDRESS value. The username and password are the credentials that were configured in the WebSphere ESB (under Security, Business Integration Security on the WebSphere administrative console for the Common Event Infrastructure). The credentials you provide are used by SOAtest to create the JNDI InitialContext of the events JMS topic and to create the JMS connection.

Monitoring Source	Specify the connection factory (default is jms/cei/notification/AllEventsTopicConnectionFactory). Specify the destination name (default is jms/cei/notification/AllEventsTopic). This is the CEI topic that reports all CEI events. Choose **Queue** or **Topic** from the **Destination** Type drop-down menu. (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. **Caution: Leave messages on the queue** For a discussion of potential complications with this option—and how to avoid them—see JMS Queue Options.
JDNI 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 to IBM WebSphere ESB.

Viewing Monitored Events

After the test runs, the Event Monitor will show the XML representation of the Common Base Events it receives from WebSphere, including the event's raw business data if it is present.