

Monitoring a Custom API-Based Events Source

This topic explains how to configure monitoring for a custom API-Based Events source.

To configure the Event Monitor tool to monitor a custom API-based event source:

1. Double-click the Event Monitor tool to open up the tool configuration panel.
2. In the **Event Source** tab, choose **Custom API-Based Events Source** from the platform drop-down menu and configure the following settings:

Connection	Specify your connection settings.
Event Retrieval	Specify the event retrieval pattern you want to use (polling at a specified time interval, polling after each test execution, or subscribing to an event producer).
User Code	Specify the location of your custom event monitoring application or scripts. See Extensibility API Patterns for details.

3. In 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	<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 Asserter 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).
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 Event Monitor waits between the time the test ends and the time it retrieves the events.