Creating Tests From Sonic ESB Transactions

This topic explains how SOAtest can monitor transactions that pass through a Sonic ESB system, then generate functional test cases that check the monitored messages. In addition to providing visibility into the systems messages, this allows you replay transactions directly from SOAtest and verify that the monitored functionality continues to work as expected.

Sections include:

Alternative Test Creation Method

Another way to create tests is to have SOAtest's recording proxy monitor traffic at one or more JMS, HTTP, or MQ endpoints as an application is exercised. SOAtest "listens" to traffic requests and responses, then builds a traffic file of legitimate request/response pairs. This traffic is then used to generate a test suite that represents the captured behavior in preconfigured SOAP Client or Messaging Client tools. See Creating Tests From Recorded HTTP, JMS or MQ Traffic for details.

Overview

SOAtest can monitor transactions that pass through a Sonic ESB, then generate functional test cases that check the monitored messages. In addition to providing visibility into the systems messages, this allows you replay transactions directly from SOAtest and verify that the monitored functionality continues to work as expected.

To achieve this, you tell SOAtest how to connect to your Sonic ESB and what destination (topic or queue) messages you want it to monitor, then you prompt it to start monitoring. SOAtest will generate a test suite of Messaging Client tests for each JMS message captured at the specified destination or for all messages within the process flow (if a process tracking topic was used). These tests are preconfigured with the connection parameters, requests, and destination information so that SOAtest can replay the same messages.

SOAtest can generate test clients for the following types of JMS messages:

- javax.jms.TextMessage
- javax.jms.MapMessage
- javax.jms.ObjectMessage
- javax.jms.BytesMessage
- javax.jms.StreamMessage

Prerequisites

The following jar files must be added to your classpath (via **Parasoft> Preferences> Parasoft> System Properties**):

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

Generating Tests from Sonic ESB Transactions

To generate tests:

1. Choose the **Other> Sonic Enterprise Service Bus** option in one of the available test creation wizards. For details on accessing the wizards, see:
   - Adding a New .tst File to an Existing Project
   - Adding a New Test Suite
2. Complete the first page of the Sonic ESB wizard as follows:
   a. In the **Connection** area, specify your Sonic ESB connection settings.
   b. 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.
   c. In the **Destination Type** field, specify whether the tracking destination is a topic or a queue.
   d. (Optional) In the **Message Selector** field, enter a value to act as a message filter. See Using Message Selector Filters for tips.
   e. If you want SOAtest to use the JMS QueueBrowser API in order to trace messages posted on a JMS queue— without removing them from the queue— enable the **Leave messages on the queue** option. 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.

f. In the JNDI properties table, specify any additional JNDI properties you want applied to this deployment.

3. Click **Next**. SOAtest will start monitoring the messages that match the settings specified in the previous wizard page. If you run another application that sends messages to the bus, those messages will be noted in this panel.
4. When you are ready to stop monitoring, click **Finish**. SOAtest will then create test cases based on the verified messages.

> **Monitoring Intermediary Messages**
>
> In addition to automatically generating functional tests from monitoring the transaction messages that touch JMS endpoints in Sonic ESB, you can also visualize and trace the intra-process events that take place as part of the transactions that are triggered by the tests, and then dissect them for validation.
>
> For details on how to do this, see Event Monitoring - ESBs, Java Apps, Databases, and other Systems).