Sample 0: Introduction to ESB

- Introduction
- Prerequisites
- Building the sample
- Executing the sample
- Analyzing the output

Introduction

This sample demonstrates how a message can be passed through the WSO2 ESB and logged before it is delivered to its receiver. This is a basic, introductory usecase of WSO2 ESB. In this sample, you will deploy a sample client and a sample back-end service, and then transfer a message to the back-end service from the client through the WSO2 ESB.

Prerequisites

For a list of prerequisites, see Prerequisites to Start the ESB Samples.

Building the sample

The XML configuration for this sample is as follows:

```
<definitions xmlns="http://ws.apache.org/ns/synapse">
  <sequence name="main">
    <in>
      <!-- log all attributes of messages passing through -->
      <log level="full"/>
      <!-- Send the message to implicit destination -->
      <send/>
    </in>
    <out>
      <!-- log all attributes of messages passing through -->
      <log level="full"/>
      <!-- send the message back to the client -->
      <send/>
    </out>
  </sequence>
</definitions>
```

This configuration file synapse_sample_0.xml is available in the <ESB_HOME>/repository/samples directory.

To build the sample

1. Start the ESB with the sample 0 configuration. For instructions on starting a sample ESB configuration, see Starting the ESB with a sample configuration.

   The operation log keeps running until the server starts, which usually takes several seconds. Wait until the server has fully booted up and displays a message similar to "WSO2 Carbon started in n seconds."

2. Start the Axis2 server. For instructions on starting the Axis2 server, see Starting the Axis2 server.

3. Deploy the back-end service SimpleStockQuoteService. For instructions on deploying sample back-end services, see Deploying sample back-end services.
Now you have a running ESB instance and a back-end service deployed. In the next section, we will send a message to the back-end service through the ESB using a sample client.

**Executing the sample**

The sample client used here is the **Stock Quote Client**, which can operate in several modes. For further details on this sample client and its operation modes, see **Stock Quote Client**.

**To execute the sample client**

- Run the following commands from the `<ESB_HOME>/samples/axis2Client` directory, to execute the **Stock Quote Client** in its various modes.

  **To execute the Stock Quote Client in the Smart Client Mode**, run the following command:

  ```
  ant stockquote -Daddurl=http://localhost:9000/services/SimpleStockQuoteService
  -Dtrpurl=http://localhost:8280/
  ```

  **To execute the Stock Quote Client in the Proxy Client Mode**, run the following command:

  ```
  ant stockquote -Daddurl=http://localhost:9000/services/SimpleStockQuoteService
  -Dprxurl=http://localhost:8280/
  ```

  **To execute the Stock Quote Client in the Gateway/ Dumb Client Mode**:

  See **Sample 1: Simple Content-Based Routing (CBR) of Messages**.

  Running each command specified above triggers a sample message to the back-end service. If the message is mediated successfully, an output is displayed on the start-up console of the Axis2 server.

**Analyzing the output**

Analyze the output debug messages to understand the actions that are performed.

**Note**

You can change the line `log4j.category.org.apache.synapse=INFO` to `log4j.category.org.apache.synapse=DEBUG` in the `<ESB_HOME>/repository/conf/log4j.properties` file, to run the ESB in the DEBUG mode, in order to analyze the output debug messages.

When you analyze the output debug messages for the actions in the **Smart Client Mode**, you will see the client request arriving at the ESB with a WS-Addressing `To` header set to EPR `http://localhost:9000/services/SimpleStockQuoteService`. The ESB engine logs the message in the full log level (i.e., all the message headers and the body) and then sends the message to its implicit `To` address which is `http://localhost:9000/services/SimpleStockQuoteService`.

You will see a message as follows on the Axis2 server console confirming that the message is routed to the sample server, and that the sample service hosted at the sample server generates a stock quote for the requested symbol.

```
Sat Nov 18 21:01:23 IST 2006 SimpleStockQuoteService :: Generating quote for : IBM
```
The response message generated by the service is again received by the ESB and flows through the same mediation rules, which logs the response message and then sends it back to the client. On the client console you will see an output as follows, based on the message received by the client.

```
Standard :: Stock price = $95.26454380258552
```

When you analyze the output debug messages for the actions in the **Proxy Client Mode**, you will see exactly the same behavior as in the previous scenario. However, this time the difference is at the client, as it sends the message to the WS-Addressing **To** address `http://localhost:9000/services/SimpleStockQuoteService`, but the transport specifies ESB as the HTTP Proxy.