Adding and Scheduling Tasks

A *task* runs a piece of code triggered by a timer, allowing you to run scheduled jobs at specified intervals. A task can be scheduled in the following ways:

1. Using *count* and *interval* attributes to run the task a specified number of times at a given interval.
2. Giving the scheduled time as a *cron style* entry.
3. Making the task run only once after the ESB starts by using the attribute *once*.

Having deployed a task implementation to the ESB runtime (see Writing Tasks), you can use the ESB Management Console to add a task to the "Tasks" list and schedule various instances of the task. You can use either XML configuration or UI configuration to add and schedule tasks.

**XML Configuration**

Follow the instructions below to add and schedule tasks using XML configuration.

1. Sign in. Enter your user name and password to log on to the ESB Management Console.
2. Click on "Main" in the left menu to access the "Manage" menu.
3. In the "Manage" menu, click on "Source View" under "Service Bus."
4. In the source view, add the task configuration based on your requirement.
Examples

To run every 5 seconds continuously:

```xml
<task name="CheckPrice" class="org.wso2.esb.tutorial.tasks.PlaceStockOrderTask">
  <trigger interval="5"/>
</task>
```

To run every 5 seconds for 10 times:

```xml
<task name="CheckPrice" class="org.wso2.esb.tutorial.tasks.PlaceStockOrderTask">
  <trigger interval="5" count="10"/>
</task>
```

You can also give cron-style values. To run daily at 1:30 AM:

```xml
<task name="CheckPrice" class="org.wso2.esb.tutorial.tasks.PlaceStockOrderTask">
  <trigger cron="0 30 1 * * ?"/>
</task>
```

To run only once after ESB starts:

```xml
<task name="CheckPrice" class="org.wso2.esb.tutorial.tasks.PlaceStockOrderTask">
  <trigger once="true"/>
</task>
```

Injecting the message to a named sequence or proxy service

By default, the message is sent to the Main sequence. To send it to a different sequence or to a proxy service, set the `injectTo` property to sequence or proxy, and then add the `sequenceName` or `proxyName` property to specify the name of the sequence or proxy service to use. For example:
Injecting to a sequence other than Main

```xml
<task name="SampleInjectToSequenceTask"
     class="org.apache.synapse.startup.tasks.MessageInjector"
     group="synapse.simple.quartz">
  <trigger count="2" interval="5"/>

  <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks"
            name="injectTo"
            value="sequence"/>

  <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks" name="message">
    <m0:getQuote xmlns:m0="http://services.samples">
      <m0:request>
        <m0:symbol>IBM</m0:symbol>
      </m0:request>
    </m0:getQuote>
  </property>

  <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks"
            name="sequenceName"
            value="SampleSequence"/>

</task>
```
Injecting to a proxy service

```
<task name="SampleInjectToProxyTask"
    class="org.apache.synapse.startup.tasks.MessageInjector"
    group="synapse.simple.quartz">
    <trigger count="2" interval="5"/>
    <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks" name="message">
        <m0:getQuote xmlns:m0="http://services.samples">
            <m0:request>
                <m0:symbol>IBM</m0:symbol>
            </m0:request>
        </m0:getQuote>
    </property>
    <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks"
        name="proxyName" value="SampleProxy"/>
    <property xmlns:task="http://www.wso2.org/products/wso2commons/tasks"
        name="injectTo" value="proxy"/>
</task>
```

UI Configuration

Follow the instructions below to add and schedule a task in ESB Management Console.

1. Sign in. Enter your user name and password to log in to the ESB Management Console.
2. Click **Main** in the left menu to access the Manage menu.
3. In the Manage menu, click **Scheduled Tasks** under Service Bus.

4. The Scheduled Tasks page appears, where you can add, edit, and delete tasks.

5. Click **Add Task**.
6. The New Scheduled Task page appears. Enter the required details into the fields.

- **Task Name** - Name of a scheduled task.
- **Task Group** - The group name to grouping tasks. The group name `synapse.simple.quartz` belongs to ESB - Synapse. All available groups are displayed as a drop-down menu. If there are tasks belong to some other domains, for example WSO2 Mashups tasks, then those will be shown here as a separate group names.
- **Task Implementation** - The implementation class of the task. To use the default task implementation that is available with the ESB (and therefore can be used without downloading any third-party libraries or custom JARs), specify `org.apache.synapse.startup.tasks.MessageInjector`. This class simply injects a specified message into the Synapse environment. For more information on writing custom task implementations, see Writing Tasks, Sample 300: Introduction to Tasks with Simple Trigger, and Writing Tasks Sample.
- **Trigger Type** - Trigger type for the task. This can be selected as either "Simple" or "Cron."
  - **Simple Trigger** - Defined by specifying a count and an interval, implying that the task will run a count number of times at specified intervals.
  - **Cron Trigger** - Defined using a cron expression.
- **Count** - The number of times the task will be executed.
- **Interval** - The interval between consecutive executions of a task.
- **Pinned Servers** - Provides a list of ESB server names, where this task should be started for the "Pinned Servers" value.

7. Click **Load Task Properties** to see the instance properties of the task implementation.
8. Use the instance properties fields as follows:

- **Property Name** - The unique name of the task property.
- **Property Type** - The type of property, either Literal or XML.
- **Property Value** - The value of the property.
- **Action** - Allows you to delete a property.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Property Value</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>message</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>soapAction</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>proxyName</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sequenceName</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>injectTo</td>
<td>Literal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The org.apache.synapse.startup.tasks.MessageInjector implementation takes the following properties:

- **format** - defines the format of the message similar to Address Endpoint formats: soap11, soap12, pox, get
- **message** - you can provide an XML or literal value depending on message format.

**Note**

When you add a scheduled task, it is mandatory to provide a value for the message property. Therefore, even if you do not want to send a message body, you have to provide an empty payload as the value to avoid an exception being thrown.

- **soapAction** - specify the SOAP Action to use when sending the message to the endpoint.
- **to** - specify the endpoint address.
- **injectTo** - specify whether to inject a message to a proxy service or sequence. This field takes values 'sequence' or 'proxy' and 'main' to inject to main sequence.
- **proxyName** - if injectTo contains 'proxy' then the name of the proxy to inject the message to is specified here.
- **sequenceName** - if injectTo contains 'sequence' then the name of the sequence to inject the message to is specified here.

9. Click **Schedule** to apply the settings.

See an example of task scheduling [here](#).