Message Processors

A message processor is used to deliver messages that have been temporarily stored in a message store. This approach is useful for serving traffic to back-end services that can only accept messages at a given rate, whereas incoming traffic to the ESB arrives at different rates. You use the Store mediator to store messages in the message store, and then you use a message processor to deliver messages from the message store to the back-end service at a given rate. Using message processors and message stores allows you to implement different messaging and integration patterns.

Multiple message processors can use the same message store. For example, in a clustered environment, each of the nodes might have an instance of the same message processor, each of which would connect to the same message store and evenly consume messages from it. The message store acts as a manager of these consumers and their connections and ensures that messages are processed by only one message processor, preventing message duplication.

The ESB ships with the following message processor implementations:

- Scheduled Message Forwarding Processor
- Message Sampling Processor

You can also implement your own message processor by implementing the MessageProcessor interface and adding the message processor to the configuration.

Message Processor Configuration

You can add, edit, delete, and activate/deactivate message processors by filling in the fields in the relevant screens in the Management Console. Note that you must have added a message store before you can add a message processor. You can also configure a message processor in the source view in the Management Console by adding the <messageProcessor> element as a top-level entry in the configuration as follows:

```
<messageProcessor class="classname" name="unique string" targetEndpoint="endpoint name" messageStore="associated message store name">
   <parameter name="string">"string"</parameter>*
</messageProcessor>
```

Sampling and custom processors
<messageProcessor class="classname" name="unique string" messageStore="associated message store name">
  <parameter name="string">"string"</parameter>*
</messageProcessor>

**Note**

Message Processor does not work in tenant mode. It is a limitation in the current implementation.