*FOR:*

**Headstrong**

*CONTENT:*

Teevra Specification for Process Configuration Model

*Version 2.0*

*April 18, 2013*

L



**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Release Date | Author | Description of Change |
| 0.1 | 06-Aug-2009 | Arunkumar.K | Template Creation |
| 0.2 | 11-Aug-2009 | Viswanath P | Added details for all the sections except assumptions, appendix |

**Reference Documents**

|  |
| --- |
| **Document Name** |
| Teevra\_Server\_Design.doc |
| Teevra\_Systems\_Requirements\_Document.doc |

[1 Introduction 4](#_Toc237339143)

[1.1 Scope of document 4](#_Toc237339144)

[2 Assumptions / Dependencies 4](#_Toc237339145)

[3 Overview 4](#_Toc237339146)

[4 [Entity 1 ex. Connector] Schema 4](#_Toc237339147)

[5 [Entity 2 ex. Mediator] Schema 4](#_Toc237339148)

[6 [Entity 3 ex. Route] Schema 4](#_Toc237339149)

[7 Sample Process Configuration Model 4](#_Toc237339150)

[8 Appendix 4](#_Toc237339151)

[8.1 [Component 1] Configuration 4](#_Toc237339152)

[8.2 [Component 2] Configuration 4](#_Toc237339153)

# Introduction

ProcessConfigurationModel (PCM) is an XML file which is used to drive the process in the fusion engine. PCM can be easily hand edited, backed up from the file system, or even included into version control for easier management and control (e.g. moving a configuration from development, through QA, staging and into production).

PCM defines Connectors, Mediators, Routers, Error Handlers, Error Reporters and the Message flow information which is understood by the process modeler.

## Scope of document

This document describes the configurations of various entities in the PCM viz., Connectors, Mediators, Routers, Error Reporters and the process. It also contains one sample process model graph which shows how a PCM looks like.

# Assumptions / Dependencies

*[Specify dependencies if any for the process configuration model]*

# Overview

Every process has one PCM associated with it which gets generated when user publishes the process from UI. User can publish a process once it is saved. A XML file for PCM with process id as its name is generated when the published process is assigned to a JMX server. PCM is generated from process graph using as XML Document.

Process configuration defines connectors, mediators, routers, and error reporters and process as child elements of the XML document. ‘fusion’ is the main child of the document to which all the above mentioned elements would be appended. All the element details and configurations are explained in detail separately.

# Process

* Process is the root level element in the PCM. Process is a collection of routes that would define the complete process. It also defines process level error handling and reporting
* Process is defined under process element.
* Process definition contains Routes for message flow under route element.
* It also contains the type of the process flow. It is given like flow=’ST’. By default, flow is assumed to be SEDA. Process flow can be any one of SEDA,ST,JMS.
* Error handling strategy is defined under onError element.
* Schema for process definition looks as below:

<process id="1" name="test">

<route id="sequence1" start="true">

<from id="src" />

<to id="parser" />

</route>

<route end="true" id="sequence2">

<from id="parser" />

<to id="dest" />

</route>

<onError exceptionType="Business">

<errorHandlingStrategy type="ReportAndAbortJob" />

</onError>

<onError exceptionType="System">

<errorHandlingStrategy type="ReportAndAbortJob" />

</onError>

</process>

# Route

* A Route is a combination of source and target endpoints. It contains following attributes.
* **from:** is the source of message for the route. IDs of Mediation and Binding endpoints can be specified in the from tag.
* **to:** can be any Endpoint including a mediation endpoint. Any message received by the ***from*** Endpoint will be passed on the ***to*** endpoint that in turn can send it to other Endpoint
* Routes are defined under route element
* Route properties are defined in the property element with name attribute as the property name and value as the property value.
* All Routes are assigned a unique id which would be used by the process modeler to determine its sequence in the process
* Routes from where the process starts has attribute start=”true” and the routes at where the process ends has attribute end=”true”
* Schema for the same looks as below:

<route id="sequence3" start="true">

<from id="1" />

<to id="2" />

</route>

# On Error

* On Error element defines the error reporters and error handling strategies where A Route is a combination of source and target endpoints. It contains following attributes.
* Error handling and reporting strategy is defined under onError element.
* Error handling strategies at component and process level are defined under mediator and process tags respectively.
* All onError elements have an exception type, an error handling strategy and an error reporting configuration.
* Exception types can be of two types
  + Business Exception
  + System Exception
* Error Handling strategy can be any one of the following
  + ReportAndContinueJob
  + ReportAndAbortJob
  + ReportAndStopProcess
  + ReportAndAbortJobIfThresholdCrossed
* Error reporter properties are defined in the errorReporter element with an id for the error reporter and a formatter type of the error reporting. Formatter type can be any one of the following
  + fixml
  + xml
  + db
* Schema for the same looks as below:

<onError exceptionType="Business">

<errorReporter formatter="fixml" id="errreporter" />

<errorHandlingStrategy type="ReportAndAbortJob" />

</onError>

# Connector

* Connector is an endpoint which provides or consumes services using some sort of communication protocol, such as HTTP, JMS etc., These components sit at the boundary of the process and communicate with the environment outside the application
* Following attributes and properties are defined for the connector
* **Type:** It’s a mandatory attribute. This is required to resolve the actual component.
* **Id:** It’s a mandatory attribute. This is required to uniquely identify the connector in the process.
* Connectors are defined under Connector element.
* Connector properties are defined in the property element with name attribute as the property name and value as the property value.
* All connectors are assigned a unique id which would be used in the process definition.
* Multiple configurations can be specified for the same type of connector with different IDs. Both the configuration would represent two different connectors.
* Schema for the same looks as below:

<connector id="src" type="FILE">

<property name="directory" value="c:\temp\input" />

<property name="binary" value="FALSE" />

<property name="lock" value="FALSE" />

<property name="delete" value="FALSE" />

</connector>

# Mediator

* Mediator is a functional component that offers a message mediation service. It provides or consumes services locally within the Fusion runtime environment, enabling services such as business logic, processing and transformation services.
* Following are the properties that are defined with the mediator.
* **Type:** It’s a mandatory attribute. This is required to resolve the actual component.
* **Id:** It’s a mandatory attribute. This is required to uniquely identify the mediator in the process.
* Mediators are defined under Mediator element.
* Mediator properties are defined in the property element with name attribute as the property name and value as the property value.
* All mediators are assigned a unique id which would be used in the process definition.
* Multiple configurations can be specified for the same type of mediator with different IDs. Both the configuration would represent two different mediators.
* Error handling strategy is defined under onError element
* Schema for the same looks as below:

<mediator id="InputToOutput" type="TransformerService">

<property name="inputObjectSchemaDescriptor"

value=" SchemaInput " />

<property name="objectSchemaDescriptor"

value="SchemaOutput" />

<property name="schemaMapping" value="" />

</mediator>

# Router

* Router is a functional component which provides ability to determine/change message at run time based on the message content or other parameters. Some of the routing components are.
  + Content Based Router
  + Splitter
  + Aggregator
* Following are the properties that are defined with the router.
* **Type:** It’s a mandatory attribute. This is required to resolve the actual component.
* **Id:** It’s a mandatory attribute. This is required to uniquely identify the router in the process.
* Routers are defined under router element.
* Router properties are defined in the property element with name attribute as the property name and value as the property value.
* All routers are assigned a unique id which would be used in the process definition.
* Multiple configurations can be specified for the same type of router with different IDs. Both the configuration would represent two different routers.
* Error handling strategy is defined under onError element
* Schema for the same looks as below:

<router id="InputToOutput" type="Splitter">

<property name="inputObjectSchemaDescriptor"

value="SchemaInput" />

</router>

# Error Reporter

* An error reporter is an endpoint used to log the error reports at both component and process level. Connectors do not have error reporters connected to them. Following attributes and properties are defined for the error reporter
* **Type:** It’s a mandatory attribute. This is required to resolve the actual component.
* **Id:** It’s a mandatory attribute. This is required to uniquely identify the error reporter in the process.
* Error Reporters are defined under errorReporter element.
* Error Reporter properties are defined in the property element with name attribute as the property name and value as the property value.
* All error reporters are assigned a unique id which would be used in the process definition.
* Multiple configurations can be specified for the same type of connector with different IDs. Both the configuration would represent two different error reporters.
* Schema for the same looks as below:

<errorReporter id="ErrorReporter" type="JMS">

<property name="brokerUrl" value="tcp://localhost:61616" />

<property name="destination" value="errorreporter" />

<property name="destinationType" value="queue" />

<property name="provider" value="Active MQ" />

</errorReporter>

# Sample Process Configuration Model

<?xml version="1.0" encoding="UTF-8"?>

<fusion>

<mediators>

<mediator id="3" type="DbFormatterService">

<property name="objectSchemaDescriptor"

value="BloombergRawSchema" />

<property name="schemaMapping" value="" />

<property name="schemaType" value="Default" />

</mediator>

<mediator id="2" type="XmlParserService">

<property name="objectSchemaDescriptor"

value="BloombergRawSchema" />

<property name="schemaMapping" value="" />

</mediator>

</mediators>

<routers />

<connectors>

<connector id="1" type="DATABASE">

<property name="url" value="jdbc:postgresql://10.200.20.106/Teevradev" />

<property name="userName" value="test" />

<property name="password" value="test" />

<property name="query" value="test" />

<property name="interval" value="test" />

<property name="serverType" value="PostGre SQL" />

<property name="updateQuery" value="test" />

<property name="readSize" value="test" />

</connector>

<connector id="4" type="DATABASE">

<property name="url" value="jdbc:postgresql://10.200.20.106/Teevradev" />

<property name="userName" value="test" />

<property name="password" value="test" />

<property name="query" value="test" />

<property name="interval" value="test" />

<property name="serverType" value="PostGre SQL" />

<property name="updateQuery" value="test" />

<property name="readSize" value="test" />

</connector>

</connectors>

<errorReporters>

<errorReporter id="errreporter" type="FILE">

<property name="directory" value="c:\temp\errreport" />

<property name="overwriteFile" value="FALSE" />

</errorReporter>

</errorReporters>

<process id="74" name="test" flow=”ST”>

<route end="true" id="sequence1">

<from id="3" />

<to id="4" />

</route>

<route id="sequence2">

<from id="2" />

<to id="3" />

</route>

<route id="sequence3" start="true">

<from id="1" />

<to id="2" />

</route>

<onError exceptionType="Business">

<errorReporter formatter="fixml" id="errreporter" />

<errorHandlingStrategy type="ReportAndAbortJob" />

</onError>

<onError exceptionType="System">

<errorReporter formatter="fixml" id="errreporter" />

<errorHandlingStrategy type="ReportAndAbortJob" />

</onError>

</process>

# </fusion>

# Appendix

*[Enlist specific configurations for each of the component present in Fusion]*

## [Component 1] Configuration

*[Specify configuration for component 1, what value it expects etc]*

## [Component 2] Configuration

*[Specify configuration for component 2, what value it expects etc]*