*FOR:*

**Headstrong**

*CONTENT:*

Teevra Specification for Process Model Graph

*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 |
| 1.0 | 10-Aug-2009 | Viswanath P/Punatar Aakash | Updated for Process Model Graph Information |
| 1.1 | 11-Aug-09 | Aakash/ Viswanath P | Added the class diagram |

**Reference Documents**

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

[1 Introduction 4](#_Toc237779503)

[2 Scope of document 4](#_Toc237779504)

[3 Assumptions / Dependencies 4](#_Toc237779505)

[4 Overview 4](#_Toc237779506)

[5 ProcessGraphVO 6](#_Toc237779507)

[6 ProcessGraphNodeVO 7](#_Toc237779508)

[7 ProcessGraphLinkVO 8](#_Toc237779509)

[8 ProcessServiceEO 9](#_Toc237779510)

[9 ProcessServiceConfigurationEO 10](#_Toc237779511)

[10 Sample Process Model Graph 10](#_Toc237779512)

# Introduction

Process model graph is used to preserve the graph view of the process as constructed by user along with the configurations entered for each component in the graph. Process Model Graph is an XML serialized Object model. Every process has one process model graph which is stored in database as an XML, whenever a process is created and saved. Process model graph is used to render the graphical view of process upon the user request.

# Scope of document

This document describes the design of various entities of Process Model Graph. This document describes the serialized and de-serialized mechanism of Process Model Graph.

# Assumptions / Dependencies

* At Process level user can configure Error Reporting mechanism along with Error handling strategy.
* Nodes can’t be self connecting and Process Graph can’t be cyclic i.e. Destination Node should not connect to the source Node.

# Overview

Process Model Graph contains three entities to store all the information of the process viz.*,*

* Process Graph Node(node)
* Process Graph Links(link)
* Error handling and Reporting

The process id is the identifier for the process graph. On UI, user can create and model all the components and connectors.

* Configurations and graph coordinate details of a component are stored in one Process Graph Node (node).
* Node can also be an error reporting component.
* A map is created to store node id and node as key, value pair. Node Id should be unique within the process.
* Graph coordinate details of a connector are stored in one Process Graph Link. One Process Graph Link (link) connects only two nodes. A map is created with source node as key and the list of map of links as value. Map of links stores the links with corresponding destination id as key and the link as value for all the destination node connected from the source node.
* Error Handling and reporting at process level is stored in a Component Service entity with all its configurations.
* The process Model graph is generated when user fills in all the mandatory and required configurations and saves it. Saving of the process validates the uniqueness and length of the name and length of the description.
* It also checks if the components in the process are configured. Component configurations and uniqueness of service/node ids are validated at component level.

The graph entity is converted to a byte array stream using XML encoding which is finally stored as a string in database. Following is the class diagram of the Process Model Graph



# ProcessGraphVO

ProcessGraphVO is the Object model for the process model graph used in the presentation layer. It has the following elements.



* 1. **Long processId:** Unique identifier for a process graph
  2. **Map<String, ProcessGraphNodeVO> nodes:** List of nodes in the process graph along with geometry
  3. **Map<String, Map<String, ProcessGraphLinkVO>> links:** List of connectors present in the graph
  4. **ProcessServiceEO errorReportNHandlingConfigs:** Error Reporting and Handling information

# ProcessGraphNodeVO

ProcessGraphNodeVO represents a node in the process model graph. It has the following elements.



* 1. **Long componentId**: Identifier of the component the node represents.
  2. **String nodeId**: Optional. dentifier of the
  3. **Long x, y:** The X and Y co-ordinates of the node in the graph. This denotes the top left corner of the smallest rectangle that can be drawn to cover the node in the graph.
  4. **ProcessServiceEO serviceDetails**: Configuration for the node.(refer the section 6 for ProcessServiceEO)

# ProcessGraphLinkVO

ProcessGraphLinkVO represents a connector in the process model graph. It has the following elements.

* 1. **String startNodeId**: Name of the start node for the link
  2. **String endNodeId**: Name of the end node for the link
  3. **String startX, Y**: The starting X & Y co-ordinate of the link
  4. **String endX, Y** : The ending X and Y co-ordinate of the link

# ProcessServiceEO

ProcessServiceEO corresponds to the 'Process\_Component' table. It has the following elements.



* 1. **Long prcsCompId**: Unique identifier for a component instance in process
  2. **Long prcsId:** Long prcsId
  3. **Long compId:** Long compId
  4. **String serviced**: Service id for the component that is unique within a process model
  5. **String** **businessErrorStrategy:** Error strategy for business error
  6. **String systemErrorStrategy:** Error strategy for system error
  7. **List<ProcessServiceConfigurationEO> configurations:** configuration of individual Node. (refer the section 7 for ProcessServiceConfigurationEO)

# ProcessServiceConfigurationEO

ProcessServiceConfigurationEO corresponds to the 'Process\_Service\_Configuration' table.It has the following elements.



* 1. **Long prcsId**: Unique identifier for identifier for a process
  2. String serviceId: Service id for the component
  3. String configKey: Identifier for the component
  4. String simpleValue : Simple config value that needs to be lined in PCM
  5. String complexValue: String complexValue
  6. String serviceType; Identifies the type of component (server alias)

# Sample Process Model Graph

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

<java version="1.6.0\_07" class="java.beans.XMLDecoder">

<object class="com.headstrong.Teevra.services.process.vo.ProcessGraphVO">

<void property="errorReportNHandlingConfigs">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceEO">

<void property="businessErrorStrategy">

<string>ReportAndAbortJob</string>

</void>

<void property="compId">

<long>26</long>

</void>

<void property="configurations">

<object class="flex.messaging.io.ArrayCollection">

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>directory</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>error</string>

</void>

<void property="simpleValue">

<string>c:\temp\error</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>overwriteFile</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>error</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void id="ObjectArray0" property="source"/>

<void property="source">

<object idref="ObjectArray0"/>

</void>

</object>

</void>

<void property="prcsCompId">

<long>0</long>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>error</string>

</void>

<void property="systemErrorStrategy">

<string>ReportAndAbortJob</string>

</void>

</object>

</void>

<void property="links">

<object class="flex.messaging.io.amf.ASObject">

<void method="put">

<string>source</string>

<object class="flex.messaging.io.amf.ASObject">

<void method="put">

<string>destination</string>

<object class="com.headstrong.Teevra.services.process.vo.ProcessGraphLinkVO">

<void property="endNodeId">

<string>destination</string>

</void>

<void property="endX">

<long>320</long>

</void>

<void property="endY">

<long>98</long>

</void>

<void property="startNodeId">

<string>source</string>

</void>

<void property="startX">

<long>183</long>

</void>

<void property="startY">

<long>98</long>

</void>

</object>

</void>

</object>

</void>

</object>

</void>

<void property="nodes">

<object class="flex.messaging.io.amf.ASObject">

<void method="put">

<string>source</string>

<object class="com.headstrong.Teevra.services.process.vo.ProcessGraphNodeVO">

<void property="componentId">

<long>3</long>

</void>

<void property="nodeId">

<string>source</string>

</void>

<void property="serviceDetails">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceEO">

<void property="compId">

<long>3</long>

</void>

<void property="configurations">

<object class="flex.messaging.io.ArrayCollection">

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>directory</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>source</string>

</void>

<void property="simpleValue">

<string>c:\temp\input</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>binary</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>source</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>lock</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>source</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>delete</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>source</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void id="ObjectArray1" property="source"/>

<void property="source">

<object idref="ObjectArray1"/>

</void>

</object>

</void>

<void property="createdDate">

<void property="nanos">

<int>64000000</int>

</void>

</void>

<void property="modifiedDate">

<void property="nanos">

<int>64000000</int>

</void>

</void>

<void property="prcsCompId">

<long>0</long>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>source</string>

</void>

</object>

</void>

<void property="x">

<long>133</long>

</void>

<void property="y">

<long>74</long>

</void>

</object>

</void>

<void method="put">

<string>destination</string>

<object class="com.headstrong.Teevra.services.process.vo.ProcessGraphNodeVO">

<void property="componentId">

<long>3</long>

</void>

<void property="nodeId">

<string>destination</string>

</void>

<void property="serviceDetails">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceEO">

<void property="compId">

<long>3</long>

</void>

<void property="configurations">

<object class="flex.messaging.io.ArrayCollection">

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>directory</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>destination</string>

</void>

<void property="simpleValue">

<string>c:\temp\output</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>binary</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>destination</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>lock</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>destination</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void method="add">

<object class="com.headstrong.Teevra.services.process.eo.ProcessServiceConfigurationEO">

<void property="configKey">

<string>delete</string>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>destination</string>

</void>

<void property="simpleValue">

<string>FALSE</string>

</void>

</object>

</void>

<void id="ObjectArray2" property="source"/>

<void property="source">

<object idref="ObjectArray2"/>

</void>

</object>

</void>

<void property="createdDate">

<void property="nanos">

<int>64000000</int>

</void>

</void>

<void property="modifiedDate">

<void property="nanos">

<int>64000000</int>

</void>

</void>

<void property="prcsCompId">

<long>0</long>

</void>

<void property="prcsId">

<long>0</long>

</void>

<void property="serviceId">

<string>destination</string>

</void>

</object>

</void>

<void property="x">

<long>320</long>

</void>

<void property="y">

<long>72</long>

</void>

</object>

</void>

</object>

</void>

<void property="processId">

<long>0</long>

</void>

</object>

</java>