

TIBCO® MDM Studio Process Designer User's Guide

*Software Release 5.0
December 2015*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO and Two-Second Advantage TIB, TIBCO Adapter, Predictive Business, Information Bus, TIBCO BusinessConnect, TIBCO ActiveMatrix BusinessWorks, TIBCO Enterprise Message Service, TIBCO MDM, TIBCO MDM Studio, TIBCO MDM Studio Process Designer, TIBCO MDM Studio Rulebase Designer, TIBCO MDM Studio Repository Designer are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2007-2015 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	7
TIBCO Documentation and Support Services	8
Getting Started	9
Start MDM Process Designer	9
For Windows	9
For Linux	9
Welcome Screen	9
Accessing and Installing the Samples	10
Palette	10
Using the Process Designer	13
Creating a Project	13
Project Components	18
Project	18
Process Package	18
Process	18
Project skeleton using the MDM Template	18
Main MDM Process	19
Sub Process	19
MDM Process Templates	20
CIM Services WSDL	20
BOM Files	22
Designing a MDM process	22
Activities	23
Adding Activities	23
Adding a New Activity - Method 1	23
Adding a New Activity - Method 2	24
Modifying an Activity	24
Activity Details	24
Search for activity in a workflow	25
Defining Custom Activities	25
InterpretCommand Activity	28
Creating InterpretCommand activity	28
Example - InterpretCommand	28
Subflows	31
Using Reusable Sub Process	31
Defining Input Parameters	32
Defining Output Parameters	33
Initiate Subflow	33

Parameters	34
Parameters and Variables	34
Local Activity Parameters	34
Global and Local Process Parameters	34
Global Parameters	34
Defining Global Variables	34
Defining Local Variables	35
Input Parameters	36
Defining Values for Input Parameters	36
Example 1 - Setting the eventState	38
Example 2 - Setting the VersionOption	39
Viewing Input Parameters	39
Output Parameters	40
Defining Values for Output Parameters	40
Viewing Output Parameters	40
Add Parameters	40
Adding Parameter for eval mode=Variable	40
Adding Parameter for eval mode=Constant	42
Adding Parameter for eval mode=XPath	42
Using Content Assist	44
Adding Parameter for eval mode=Rule	45
Adding Parameter for eval mode=Lookup	46
Adding Parameter for eval mode=Property	47
Adding Parameter for eval mode=Catalog	48
Adding Parameter for eval mode=Compute	49
Adding Parameter for eval mode=Event	50
Adding Parameter for eval mode=UserProfile	50
Adding Parameter for eval mode=System	51
Adding and Modifying Custom Parameters	52
Tooltip Parameter Descriptions	54
Parameter Descriptions properties file	54
Adding a new parameters description	55
Transitions	56
Transition Types	56
Simple Transitions	56
Conditional Transitions	56
Parallel Transitions	56
Creating a Parallel Transition	57
Parallel Transitions - Limitations	58
Transition Conditions	59

Conditional Transitions	59
Interpreted Transitions	60
Creating an Interpreted Transition	60
Example of a Conditional Transition	62
Example 1	62
Example 2	63
Compiled Transitions	64
Parameters	65
Creating a Compiled Transition	66
Creating a Java Transition Project	66
Using the Condition in the process	69
Compiling and packaging the Java Project	69
Merging Jar With ECM.ear	70
Merging Jar with ECM.ear using Windows	70
Merging Jar With ECM.ear using UNIX	71
Validation	72
Validating	72
Problems Tab	72
Validation Checks	73
Miscellaneous Validations	73
Activity Validations	73
Transition Validations	73
Gateway Validations	74
Parameter Validations	74
Parameter Mapping Validations	74
Data Field Restrictions	74
Import Export and Deploying Processes	75
Import Processes into MDM Process Designer	75
Importing a Process	75
Clean Up an Imported Process	76
Limitations of the Process Import	79
Migrating Processes	80
Directly Deploying Workflows	81
Creating a MDM Deployment Server	81
Editing Server Parameters	84
Deploying Workflows	85
Deploying Workflows through Export	86
Deployment to the Workflow Engine	88
Undeploy workflows	88
Sharing Processes	89

Overview	89
Setting up MDM Process Designer to use the Subversion Repository	89
Sharing a Process Design Project	90
Accessing a shared project	92

Figures

Process Designer Sample Main (Shell) Process and sub process	19
Main Process	19
Sub process	19
Fragment View	20
Parallel Transitions	57
Not Supported: A Split connecting to a single activity.	59
Not Supported: A Join connecting to a multiple activities.	59
Supported: A Split to multiple activities; join to a single activity.	59
Conditional Transition	60
Problems Tab	73

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site:

<https://docs.tibco.com>

Documentation on the TIBCO Documentation site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit <https://docs.tibco.com>.

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

```
TIBCO_HOME/release_notes/TIB_bstudio-mdm_5.0.0_docinfo.html
```

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is C:\tibco. On UNIX systems, the default *TIBCO_HOME* is /opt/tibco.

The following documents for this product can be found on the TIBCO Documentation site:

- TIBCO MDM Studio Release Notes
- TIBCO MDM Studio Installation Guide
- TIBCO MDM Studio Process Designer Tutorial
- TIBCO MDM Studio Process Designer User's Guide
- TIBCO MDM Studio Repository Designer Tutorial
- TIBCO MDM Studio Repository Designer User's Guide
- TIBCO MDM Studio Rulebase Designer Tutorial
- TIBCO MDM Studio Rulebase Designer User's Guide
- TIBCO MDM Studio UI Builder Tutorial
- TIBCO MDM Studio UI Builder User's Guide

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://www.tibcommunity.com>

Getting Started

How to start the Process Designer, what you will see at startup, and information on accessing samples and tutorials are explained.

Start MDM Process Designer

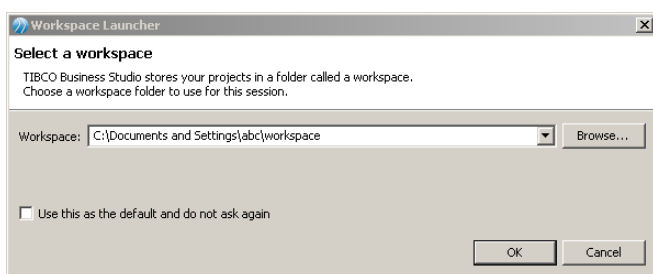
(Short description required.)

For Windows

(Short description required.)

After the installation completes, start MDM Process Designer by selecting **Start->Program Files->TIBCO-><environment name>>TIBCO MDM Studio 5.0-> Studio Designer**.

Provide a workspace location (folder where projects will be saved).



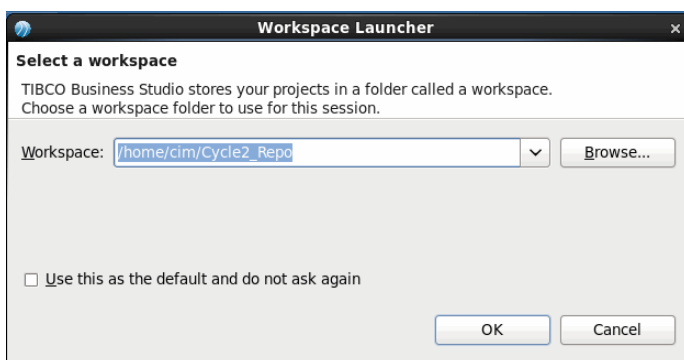
TIBCO MDM Studio opens up and you are ready to start using it. For details on how to use the Process Designer, refer [Using the Process Designer](#).

For Linux

(Short description required.)

After the installation completes, navigate to the path where studio is installed `<installed path>/studio-mdm/5.0/eclipse` and double click on **TIBCOBusinessStudio**.

Provide a workspace location (folder where projects will be saved).

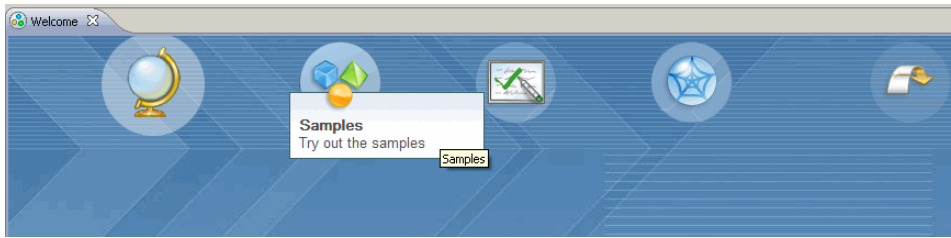


TIBCO MDM Studio opens up and you are ready to start using it. For details on how to use the Process Designer, refer [Using the Process Designer](#)

Welcome Screen

(Short description required.)

After you select the workspace for the first time, Eclipse opens up with the Welcome screen. This contains icons to samples and tutorials among other things.



This Welcome screen shows up only the first time and will not be displayed for subsequent openings of Eclipse. If you want to go to this screen again, you can access it from **Help->Welcome**.

Accessing and Installing the Samples

(Short description required.)

TIBCO MDM Studio Samples are a collection of the MDM standard processes, process modeling tutorials, repository data models, rulebase model, custom import project, MDM Model templates, and Process java transitions.

The sample models are provided to illustrate the modeling capabilities of MDM Studio. Each of these models needs further elaborations for their intended purpose. Install the sample projects to view the MDM processes, data models, and their associated rules. All the samples are available in the TIBCOHome directory.

Follow these steps to install the **Samples**.

Procedure

1. On the **File** menu, click **Import**. The import wizard is displayed.
2. From the **General** folder, select **Existing Studio Projects into Workspace**.
3. Click **Next**. The import wizard for selecting the directory path is displayed.
4. Click **Select archive file** option. Click **Browse** and select the sample project zip archives from `\<TIBCOHome>\studio-mdm\5.0\samples` folder.
5. Click **Finish**. The select project opens in the workspace.

Result

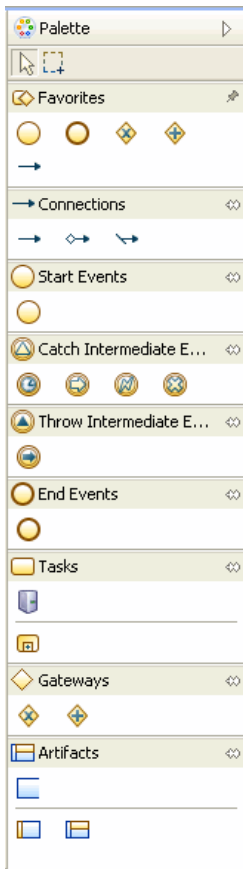


You need to import `ALLCClasses.jar` into java project to refer other mdm classes.











Palette

(Short description required.)

The Palette contains different artifacts to help you build your process model. Select and drop into the main drawing pane to define or modify your process model. You can do the following with the Palette:



Connections	
	Sequence Flow - This connection defines the process flow between objects.
	Conditional Flow - This connection defines the conditional flow between objects.
	Default Flow - This connections is a default flow if all other conditions are false.
Start Event	
	Start Event - This event does not specify any trigger.
Catch Intermediate Event	
	Catch Timer Intermediate Event - This event waits for an event triggered at a specified time.
	Catch Link Intermediate Event - This event links from a Throw Link event.
	Catch Error Intermediate Event - This event catches an error thrown inside a task or a sub process.

	Catch Cancel Intermediate Event - This event catches the cancel sub process.
Throw Intermediate Event	
	Thrown Link Intermediate Event - This event links to a Catch Link event
End Events	
	Message End Event - This is a normal end process.
Tasks	
	Service Task - This task is performed by an automated service or an application.
	Reusable Sub-Process - This process executes the external process.
Gateways	
	Exclusive Data-Based Gateway - This gateway conditionally splits or joins flows based on a data.
	Parallel Gateway - This gateway unconditionally splits or joins flows.
Artifacts	
	Text Annotation - This text allows you to document the process.
	Pool - This represents a logical participant in a process.
	Lane - This represent a sub division of a process.

Using the Process Designer

(Short description required.)

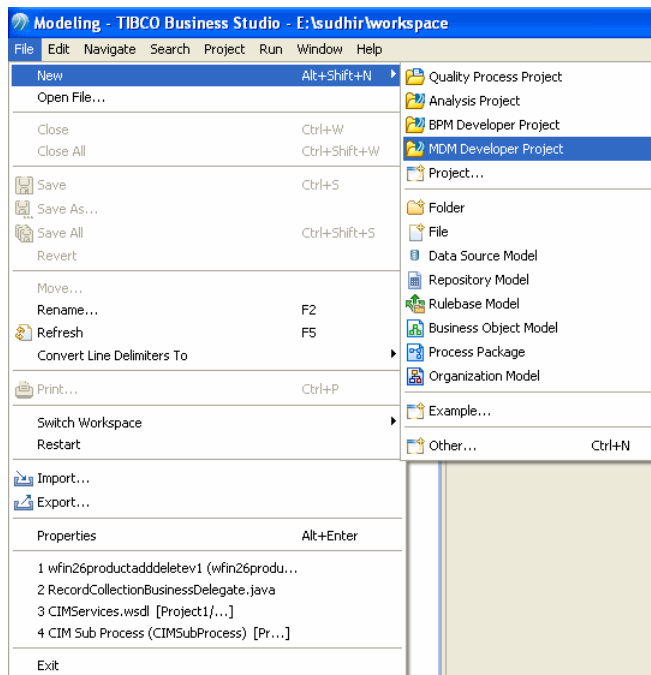
This chapter explains using the Process Designer to graphically design processes.

Creating a Project

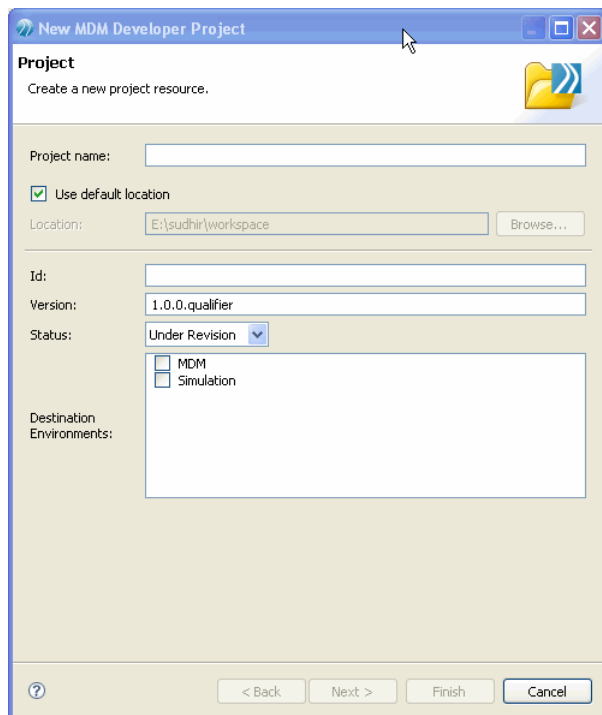
(Short description required.)

Procedure

1. In TIBCO MDM Studio, select **File- > New- > MDM Developer Project**. This starts up the new Project wizard.



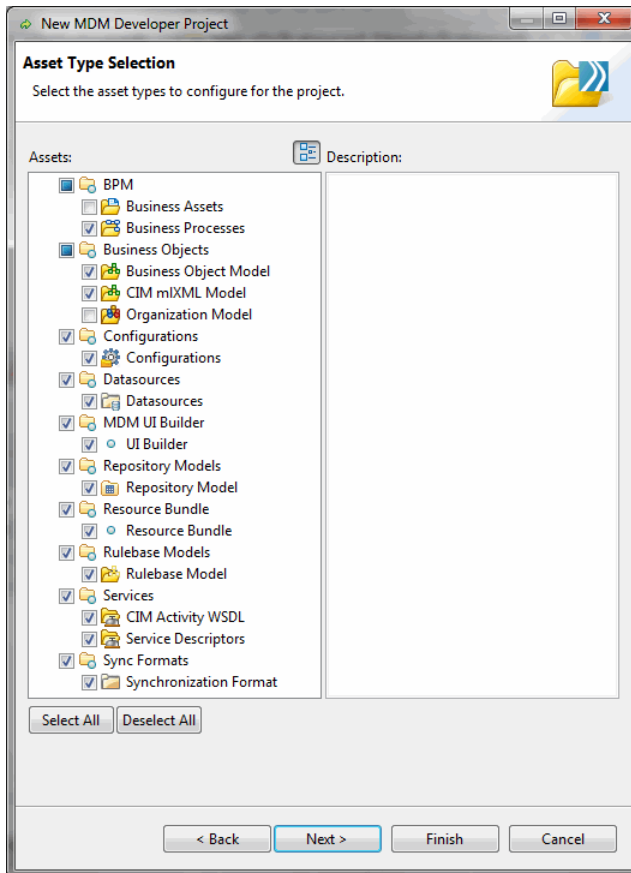
2. Provide a name for the Project and select **MDM** as the Destination Environment. Click **Next**.



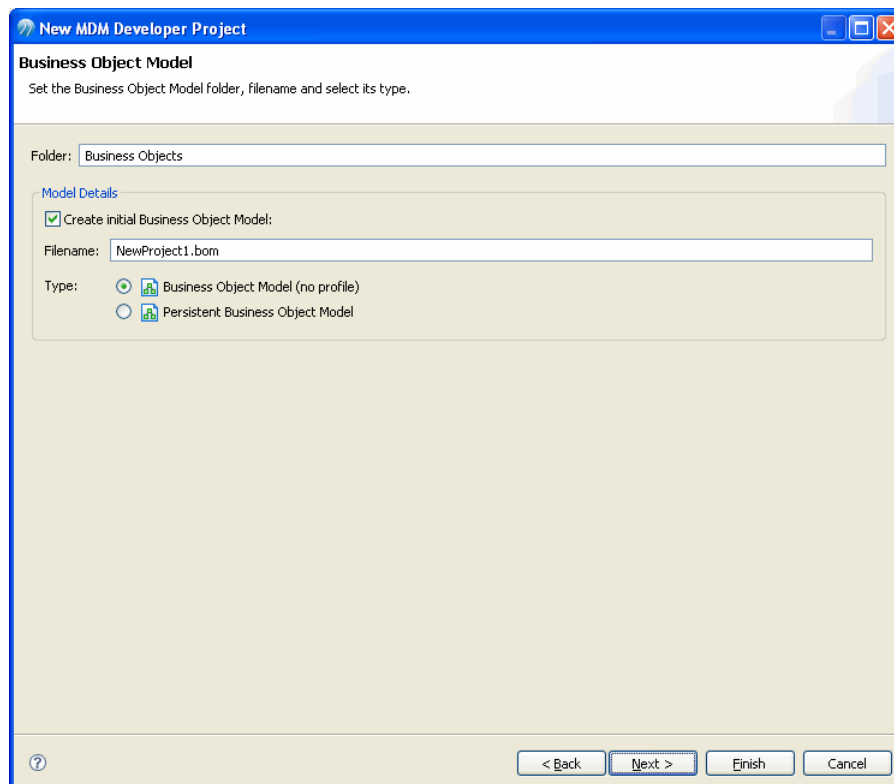
3. The Asset Type Selection dialog is displayed. Ensure that the following is selected:

- **Business Processes** (under BPM)
- **Business Object Model** and **CIM mXML Model** (under Business Objects)
- **CIM Activity WSDL** and **Service Descriptors** (under Services)

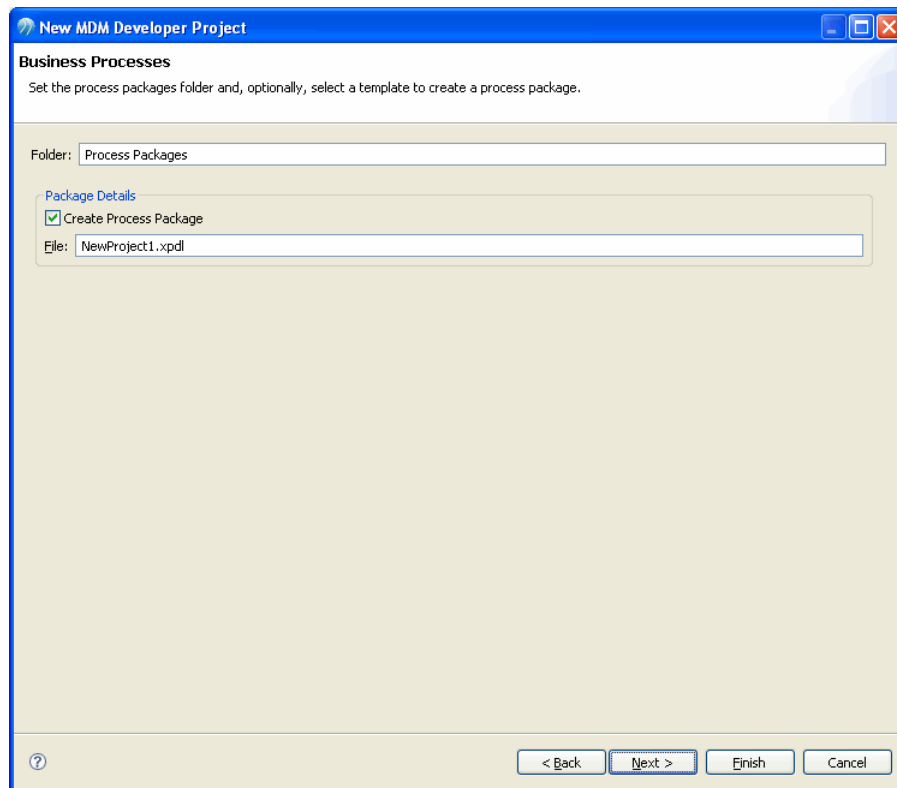
Click **Next**.



4. The Business Object Model dialog is displayed next and prompts you to set the Business Object folder. The new business object model resource that will be created is displayed. Click **Next**.



5. The Business Processes dialog is displayed next and prompts you to set the process packages folder. The new process package resource that will be created is displayed - this is the xpd file. Click **Next**.



6. The process package details are displayed. The following package details will be created:

- **Author** - user who created the Package.
- **Created** - date/time when the Package was created.
- **Description** - text description of the package.
- **Documentation Location** - URL or filename of any supporting documentation.
- **Status** - project life cycle status for informational purposes.
- **Business Version** - version information about the Package (this is user defined versioning, unrelated to any source control system).

Modify the details or leave all the default names and click **Next**.

New MDM Developer Project

Package Information
Package will be created with the following information.

Package

Package Label:

Package Name:

Package Header Information

Author:

Created:

Description:

Document Location:

Status:

Business Version:

Cost Unit:

Language:

< Back Next > Finish Cancel

- The Select Template dialog is displayed next and prompts you to select a package template. Select **MDMPackageTemplate** under **Package Templates** and click **Finish**.

Result

New MDM Developer Project

Select Template
Select template to apply to the package.

Process Details

Create a process

Label:

Package Templates

- MDMPackageTemplate
 - MDM Sub Process
 - MDM Main Process
- Business Process Templates
 - Process Templates
 - Start and End
 - Simple Process
 - Simple Integration Process
 - Case-working process
 - Simple Workflow Process
 - Event Handler
 - External Systems
 - Basic Templates
 - MDM Process Templates
 - Empty Process
 - Main process
 - Internal Edit Process
 - Process Interfaces

Diagram

The diagram shows a process flow starting with a yellow circle labeled 'Start'. An arrow points to a yellow box labeled 'Independent Sub-Process'. From this box, three arrows point down to three yellow boxes: 'SetStatusToCancel', 'SetStatusToError', and 'SetStatusToTimeOut'. From the 'SetStatusToError' box, an arrow points to a yellow circle labeled 'End'. A vertical bar on the left is labeled 'Pool' and 'Lane'.

< Back Next > Finish Cancel

This generates a skeletal MDM process. You now need to add activities and transitions to the main drawing pane expressing the necessary business logic.

Project Components

(Short description required.)

Here is brief description of projects and their components. A Project follows the following hierarchy:
Project -> Process Packages -> Processes



Before you can create a business process, you must create a Project and a Package.

Project

(Short description required.)

To define a process, you first need to create a Project to hold your process. Projects help facilitate sharing and organization of resources. Each Project has a corresponding directory in the file system (specified when you create the Project).

A Project contains Process packages and a process package contains processes.

Process Package

(Short description required.)

A project contains one or more process packages. Each package is an xpdL; each package contains - typically related - processes.

Each Process Definition (or each independent subflow) ships in a single package (xpdL file) with exactly two processes (main process and business logic).

Process

(Short description required.)

Processes are contained in a process package; each Process contains one or more Activities, which are linked together by transitions.

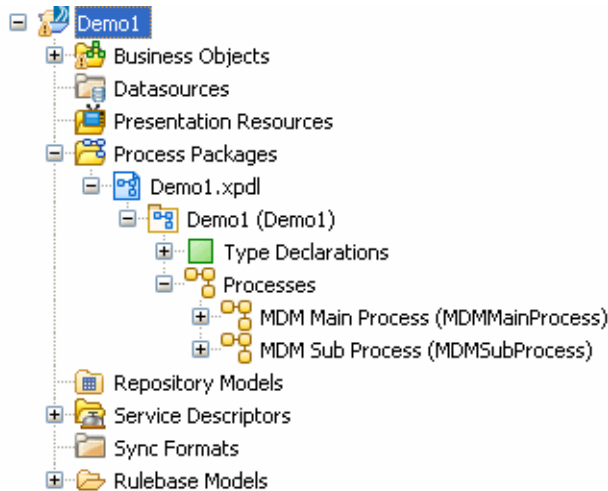
Project skeleton using the MDM Template

(Short description required.)

The MDM template allows you to create a basic process skeleton without any business logic. When you create a new Project using the MDM template, a new package with two processes (a main process and a sub process) is created.

When you use the MDM template to create a new project, you will see the project details and components in the Project Explorer. The project contains a Process packages folder, which in turn contains an XPDL file. You will see two separate processes, a main MDM process, and a sub process.

Process Designer Sample Main (Shell) Process and sub process



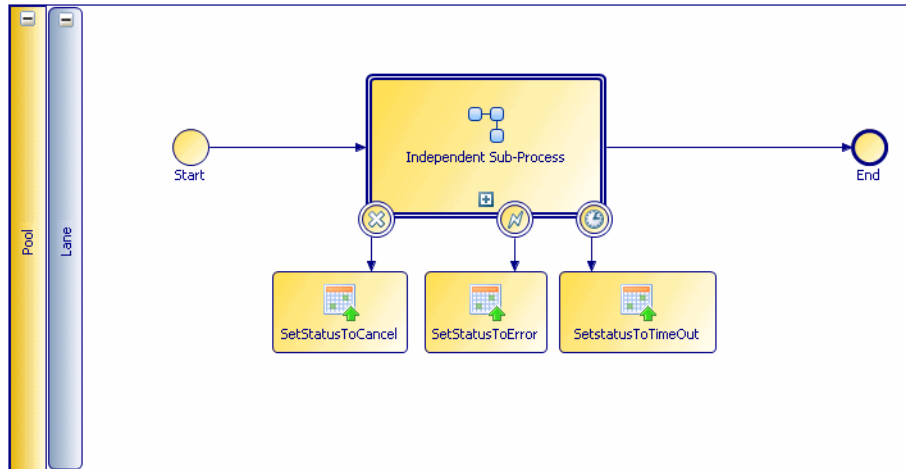
If you double-click the Main MDM process in the Project Explorer, the flow details are displayed in an editor that comes up on the right; this is the primary editor that allows you to graphically layout your process.

Main MDM Process

(Short description required.)

The Main MDM process is called the shell process and its purpose is to catch global exceptions. There are three global exceptions to handle time-outs, errors, and cancellation.

Main Process



Sub Process

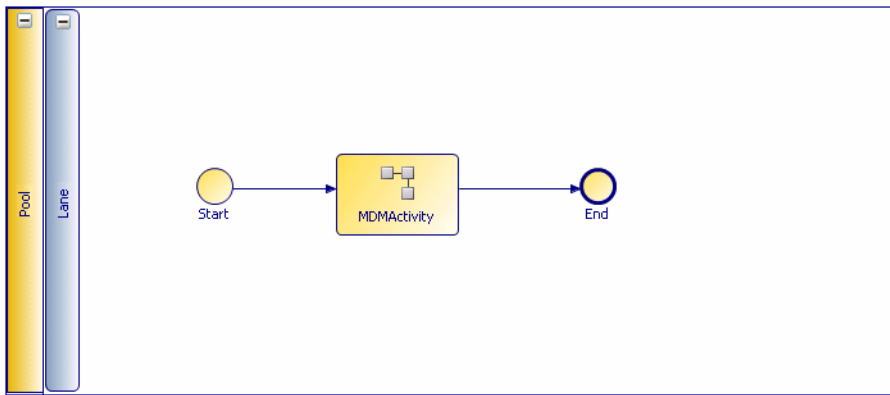
(Short description required.)

The second process is an independent sub process. This is displayed by clicking the plus sign on the Independent Sub Process task. You can also double-click **MDM Sub Process** under **Processes** in the Project Explorer.

This sub process contains a single activity, and a start and end event. There is no business logic here. This is the place where you start adding the logic to your process.

Sub process

s



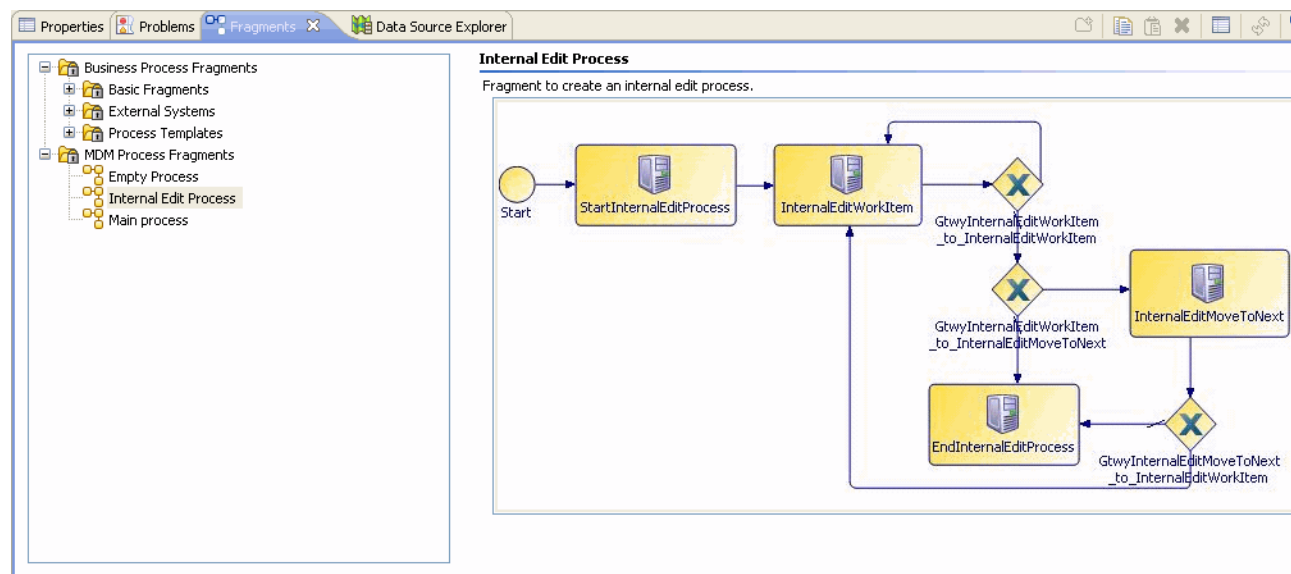
MDM Process Templates

(Short description required.)

Rather than reusing an entire process, you can use the template to quickly create new processes. For example, there may be process patterns that you frequently use. By storing these patterns or "templates" you can easily use them to construct new processes. The MDM Process Templates available are:

- Empty Process
- Internal Edit Process
- Main Process

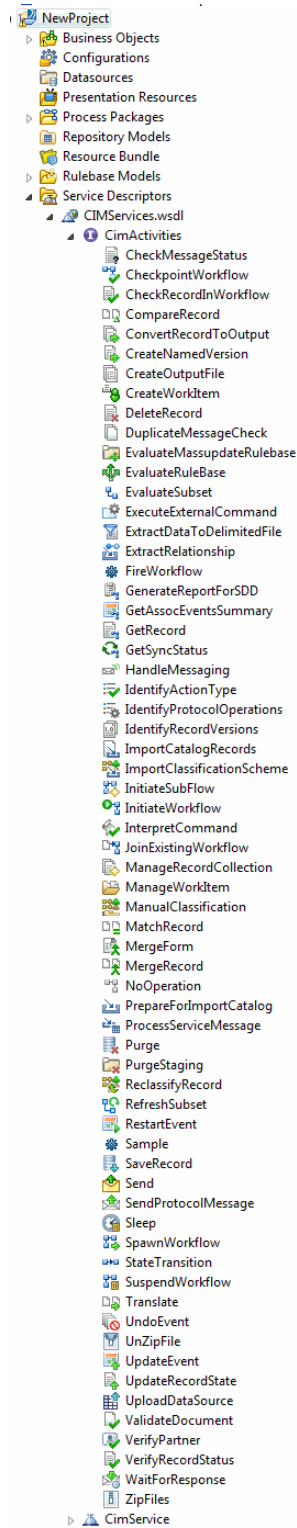
Fragment View



CIM Services WSDL

(Short description required.)

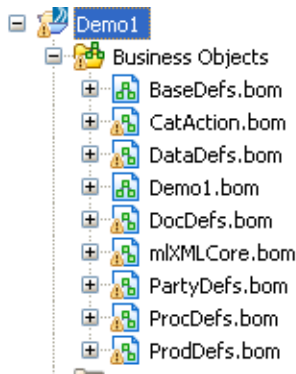
The CIMServices.wsdl contains all the MDM activity definitions. You just drag and drop these activities into your flow diagram to design your process. You can also add or modify activities and parameters in the WSDL.



For more details, see [Using the Process Designer](#) and [Parameters](#).

BOM Files

(Short description required.)



Do not change the name of the following folders and files:

- The **Business Objects** folder and all files under it.
- The **Service Descriptors** folder.
- The **CIMServices.wSDL** file.

Designing a MDM process

(Short description required.)

You can use the Palette to add Activities, Sequence flows, Events, and Annotations to your process. You can create Activities and define all related details such as the action, parameters, and so on.

The following are the high level steps in designing your process:

- Add Activities
- Map Parameters
- Add Transitions
- Validate your process
- Export and Deploy

The subsequent chapters will guide you through each of these steps.

Activities

(Short description required.)

The MDM Process Designer uses Web Service Description Language (WSDL) for definition of MDM activities. Each MDM activity is expressed as a web service operation.

For more details, see [Activity Details](#).

Adding Activities

(Short description required.)

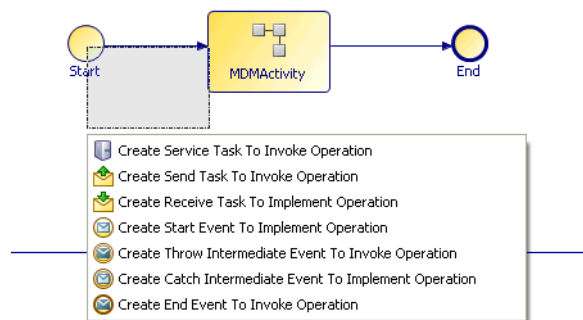
There are two ways you can insert activities:

Adding a New Activity - Method 1

(Short description required.)

Procedure

1. Go to **Service Descriptors** -> **CIMServices.wsdl** -> **CimActivities** in the Project Explorer.
2. From **CimActivities**, select the Activity you want to add. Drag and drop this activity into your flow diagram (click at the point where you want to place the activity) in the right pane.
3. Select **Create Service Task To Invoke Operation** from the drop down that is displayed.



4. Your activity will then get inserted into your flow diagram.

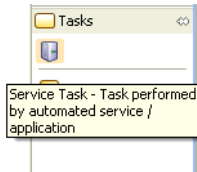


Adding a New Activity - Method 2

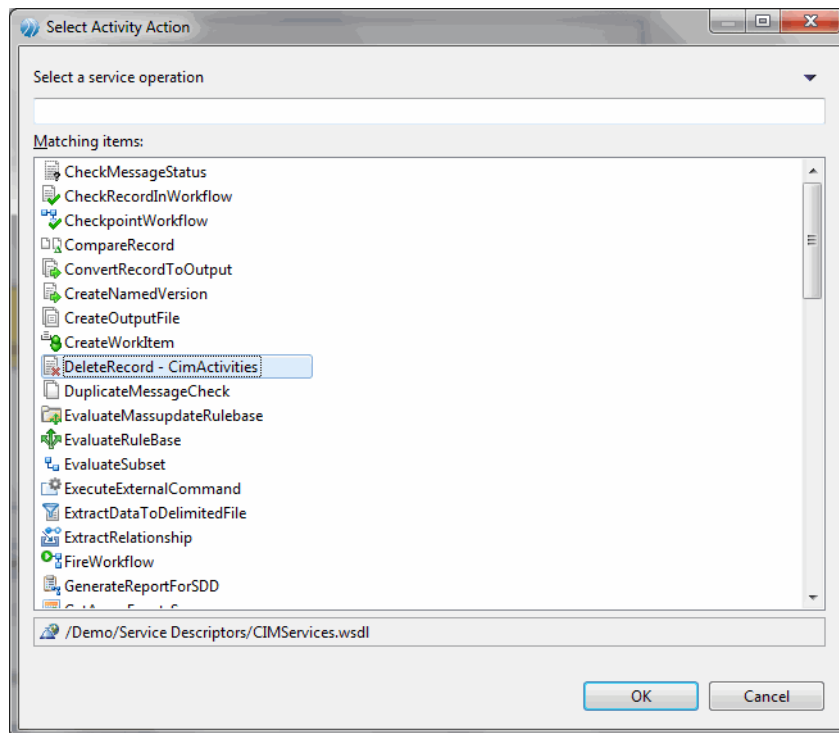
(Short description required.)

Procedure

1. Add a **Service task** to your flow diagram from the **Tasks** section in the Palette.



2. In the **Properties Window**, **General** tab, set the **Name** and **Execution mode**.
3. Click the **Select** button(...) against **Select Action**.
4. In the pop up dialog, select a MDM activity from the operation picker. The list of MDM Activities are displayed in the Matching items. Select the appropriate MDM Activities and click **OK**.



Modifying an Activity

(Short description required.)

You can modify an activity by editing the parameters from the property section.

Activity Details

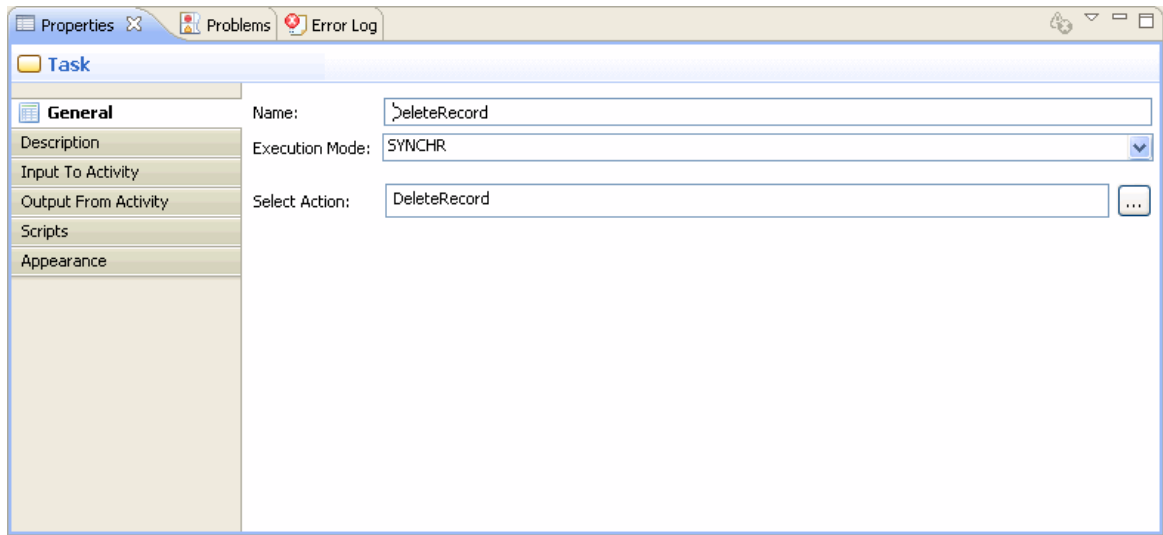
(Short description required.)

Activity details are displayed in property tab if you click on a MDM Activity, and then click the **General** tab in the **Properties Window**.

For instance, if you added a DeleteRecord activity, in the **General** tab you will see details such as:

- The **Name** set to DeleteRecord.

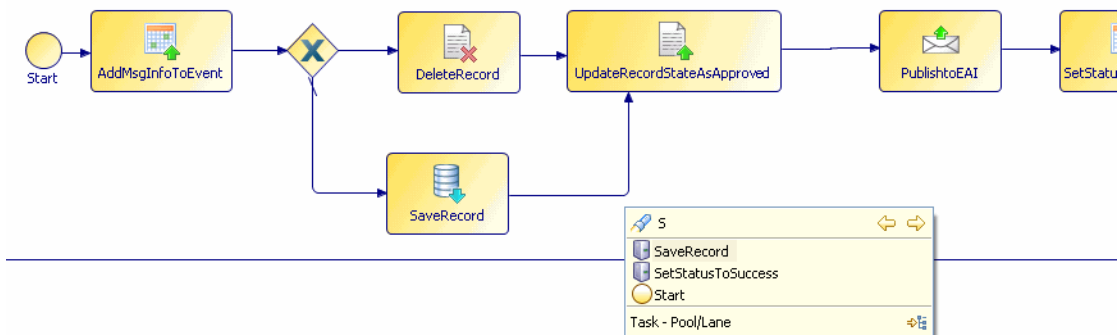
- The **Execution Mode** as SYNCHR.
- The **Select Action** as DeleteRecord.



Search for activity in a workflow

(Short description required.)

Use Ctrl+F to search for a specific activity in a workflow. Ensure that the drawing pane is selected.



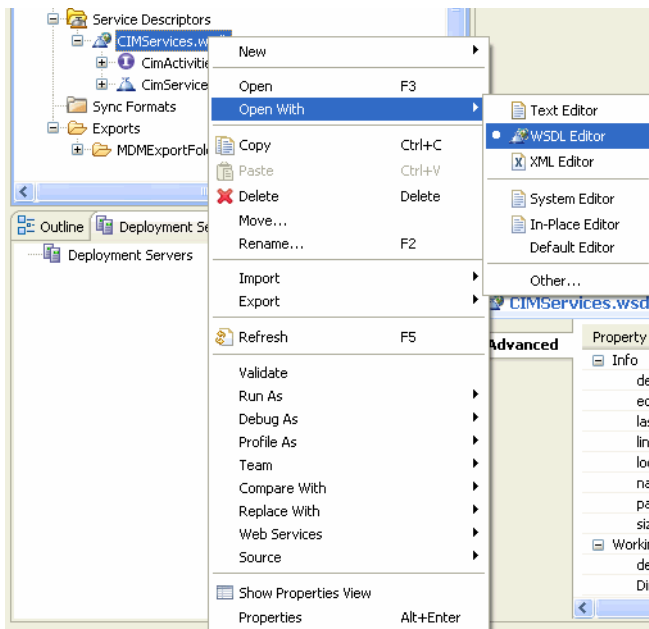
Defining Custom Activities

(Short description required.)

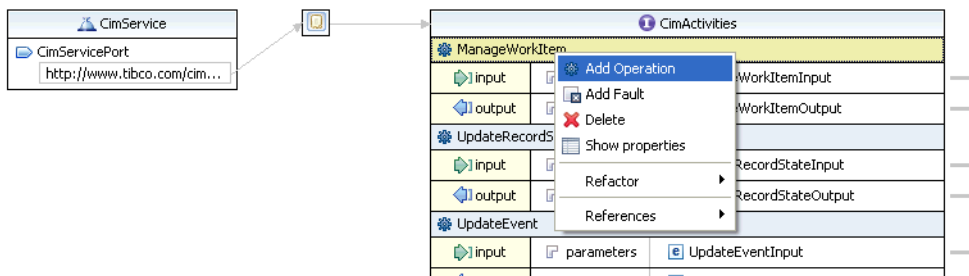
Use the WSDL editor supplied with the MDM Process Designer to add custom activities.

Procedure

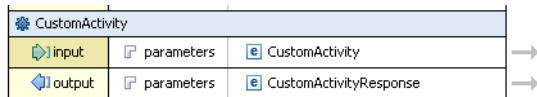
1. Open the CIMServices.wsdl in design mode by right clicking **CIMServices.wsdl** -> **Open With**-> **WSDL Editor** .



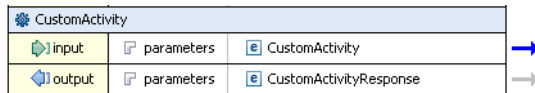
2. The list of MDM activities is displayed - right click any activity and select **Add Operation**.



3. The new activity (called **NewOperation** by default) gets added at the bottom; you can rename the activity, for example **CustomActivity** as below.



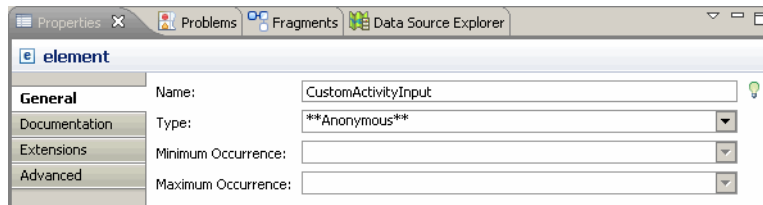
4. Click the arrow to the right of **input** in the newly added custom activity.



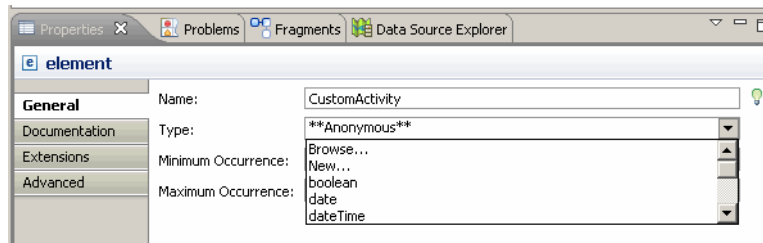
5. The inline schema will be displayed.



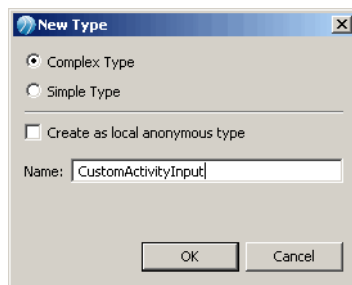
6. In the **Properties** window, **General** tab, change the **Name** to <ActivityName>Input. For example, **CustomActivityInput**.



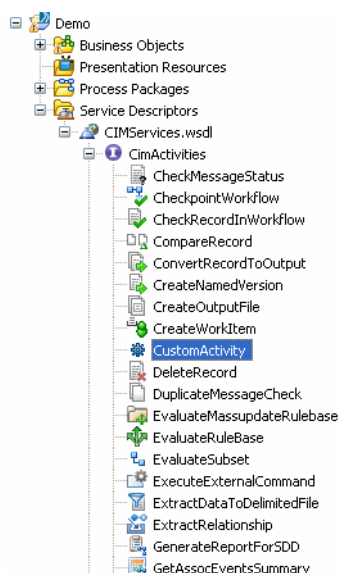
- Next, select **New...** from the **Type** dropdown.



- In the New Type dialog displayed, provide <ActivityName>InputType as **Name**. For example, CustomActivityInputType. Click **OK**.



- Follow the same steps to set the custom activity output type by clicking the arrow to the right of output, and using <CustomActivity>Output as name and <CustomActivity>OutputType as type.
- You can now add input and output parameters to the custom activity.
For more information on adding parameters, see [Adding and Modifying Custom Parameters](#).
- Once you save the wsdl, the newly defined activities will be displayed in the list of activities in **CimActivities** under **Service Descriptors**.



InterpretCommand Activity

(Short description required.)

The InterpretCommand activity allows you to execute a script and retrieve results. Input parameters are passed to the script, and output parameters are assigned values equal to the variables inside the script.

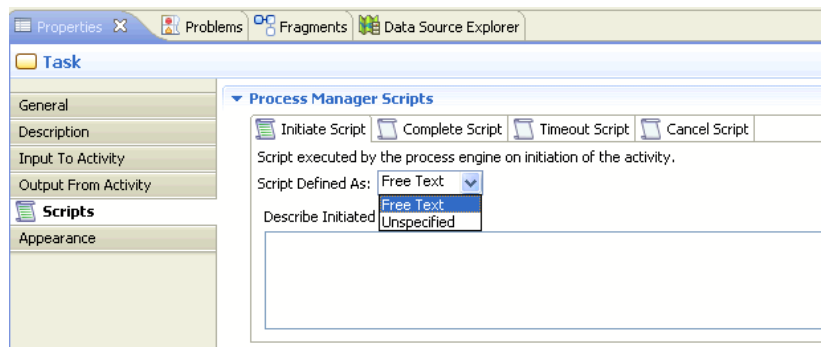
Creating InterpretCommand activity

(Short description required.)

Create an InterpretCommand activity in the Process Designer.

Procedure

1. Select the InterpretCommand activity from the list of activities in **CIM Activities** under **Service Descriptors**. Drag the activity into your flow diagram.
2. In the **Properties** window, go to the **Scripts** tab, and select **Free Text** in the **Script Defined As** dropdown. You can then enter your script.

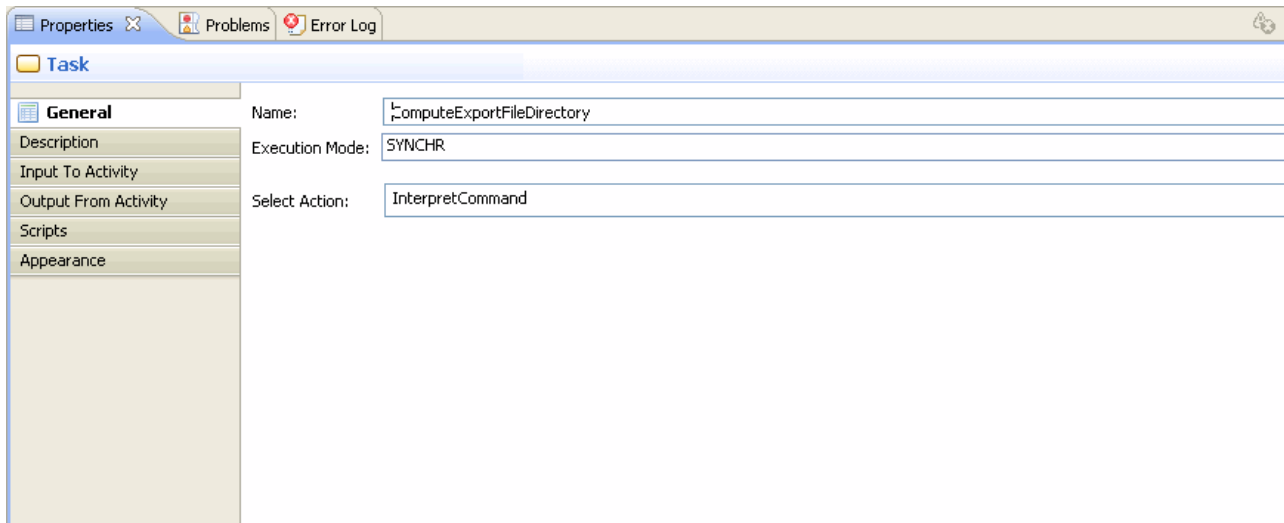


Example - InterpretCommand

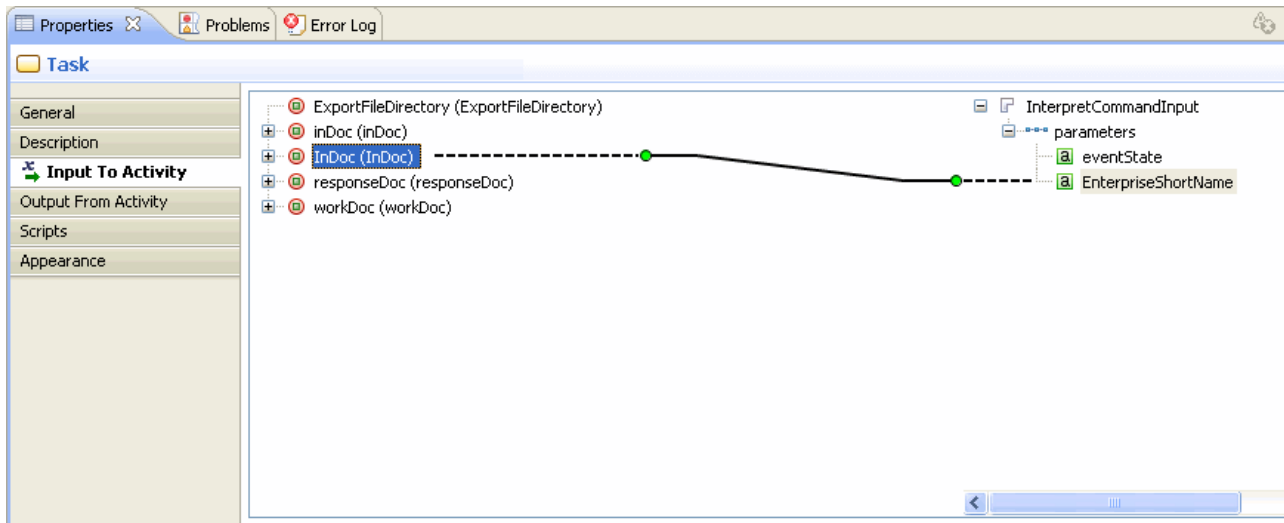
(Short description required.)

Procedure

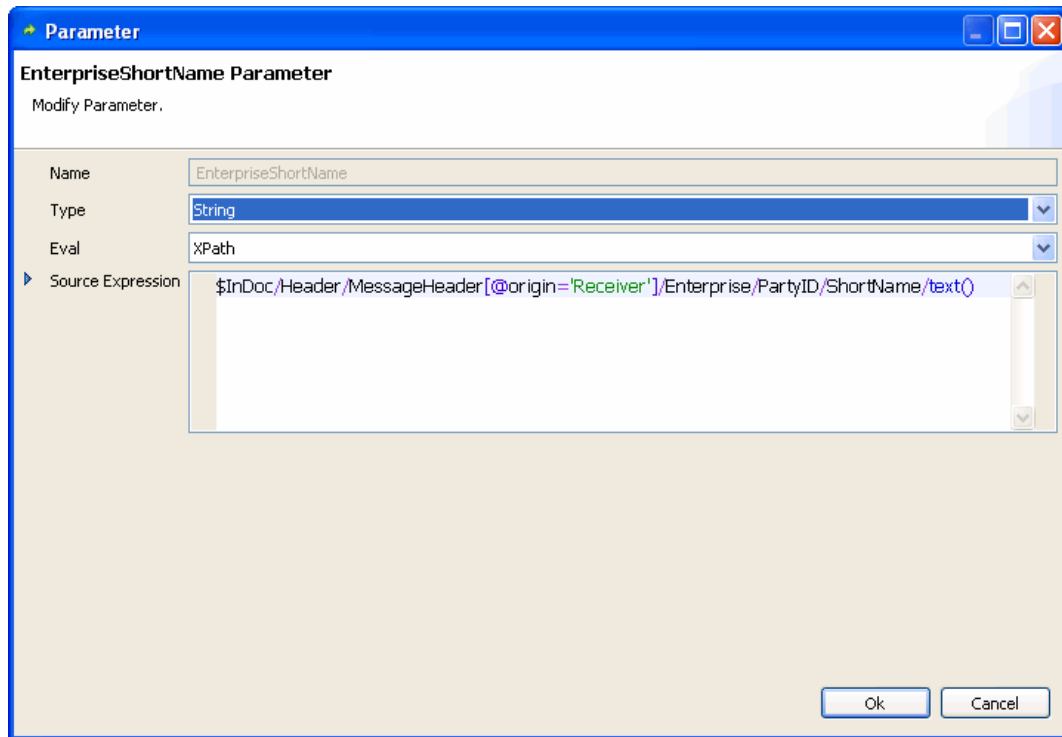
1. Select the **InterpretCommand** activity from the list of activities in **CimActivities** under **Service Descriptors**. Drag the activity into your flow diagram.



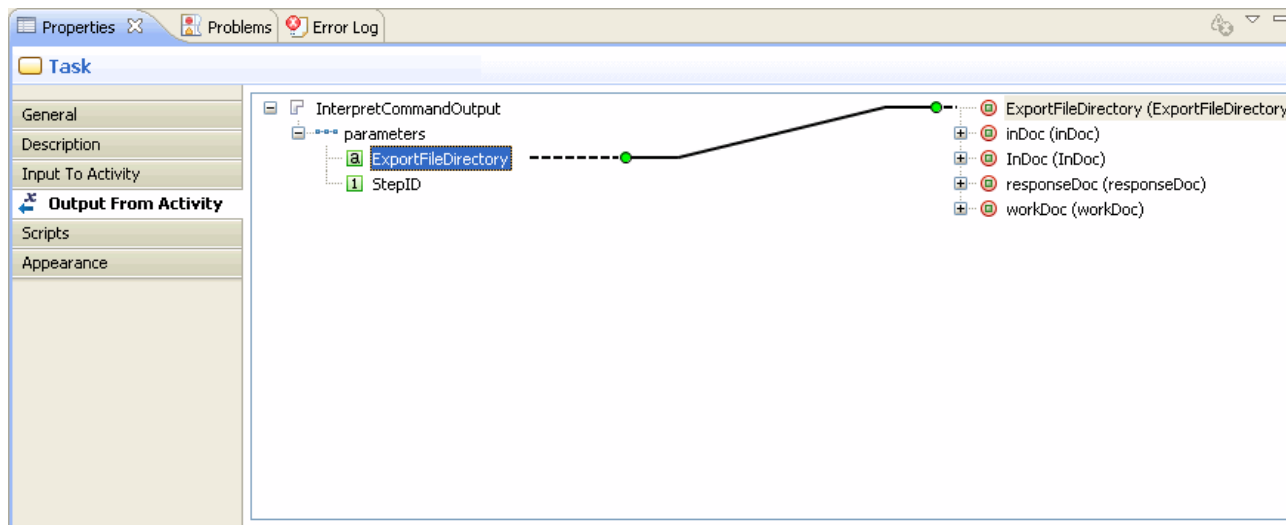
2. In the **Properties** window, go to the **Input To Activity** tab, drag the **InDoc** variable to the **EnterpriseShortName** input parameter. To create parameter with **EnterpriseName**, see [Adding and Modifying Custom Parameters](#).



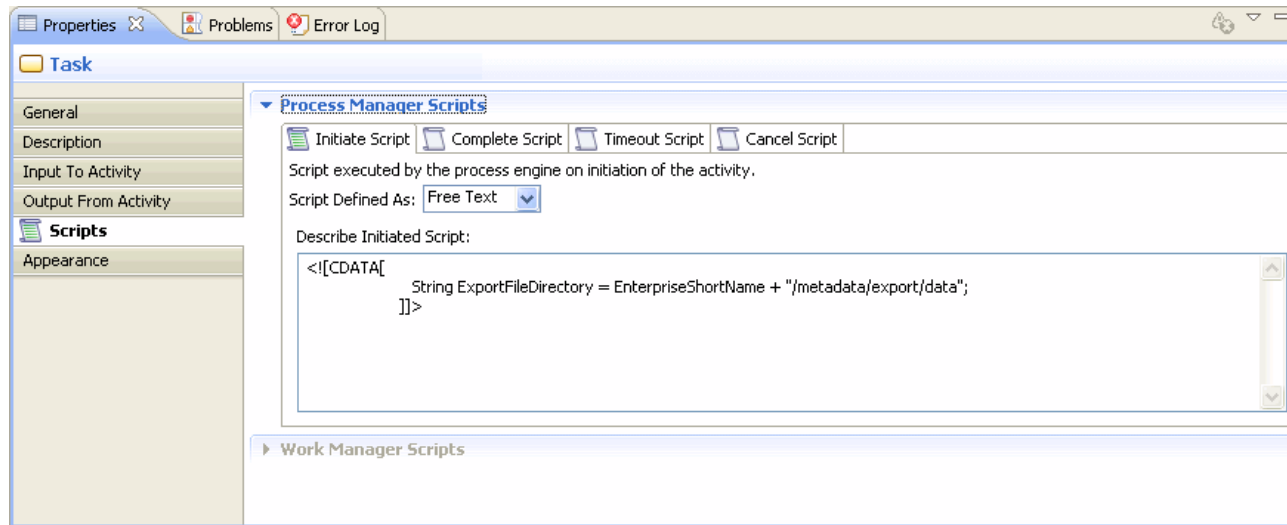
3. Double-click on the **EnterpriseShortName** parameter and select **Xpath** from Eval drop-down list. Enter the source expression in the source expression editor and click **OK**.



4. In the **Properties** window, go to the **Output From Activity** tab, drag the **ExportFileDirectory** variable to the **ExportFileDirectory** output parameter.



5. In the **Properties** window, go to the **Scripts** tab, and select **Free Text** in the **Script Defined As** dropdown. You can then enter your script.



Subflows

(Short description required.)

A subflow refers to a sub process which can be invoked from a main process. Subflows are useful when you want to break up a larger MDM workflow into manageable smaller workflow segments; then these workflow segments can be used together to compose bigger workflows.

There are two ways you can define subflows in the Process Designer:

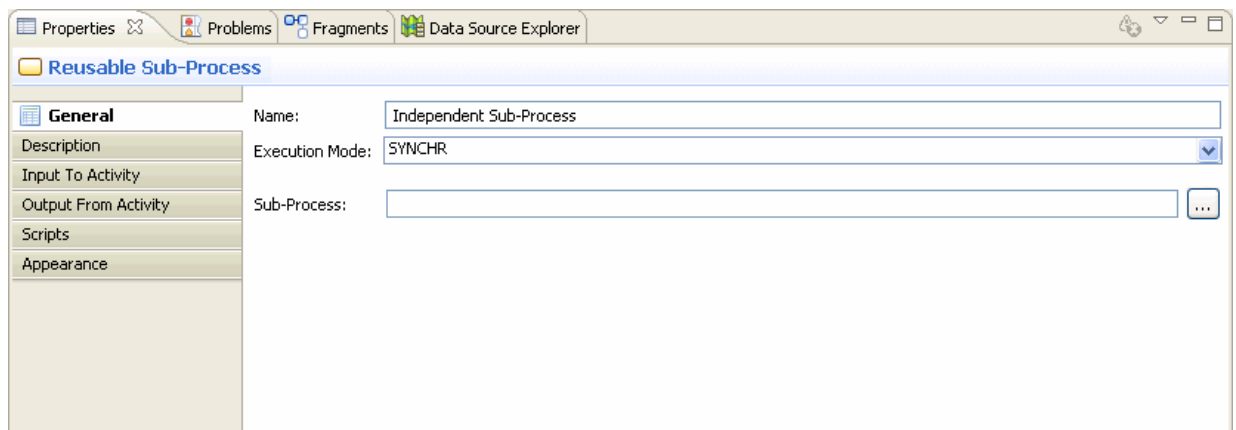
- Use the Reusable sub process palette
- Use the InitiateSubflow activity

You can have multiple Subflow activities defined for a workflow.

Using Reusable Sub Process

(Short description required.)

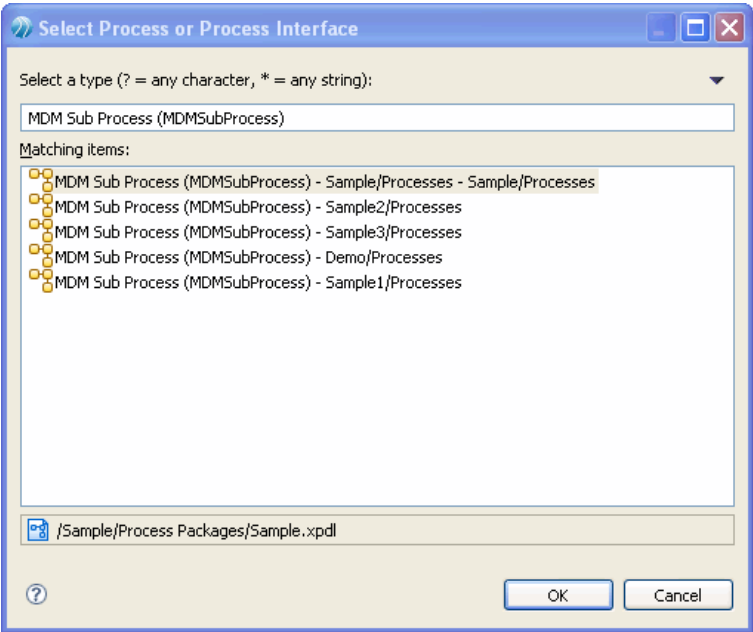
You can define a subflow by using the **Reusable Sub Process** task type and then linking that task to the subflow process.



Procedure

1. Select **Reusable Sub-Process** from the Palette and place it in your flow.
2. For the selected task, set the **Execution Mode** to **SYNCHR** in the **Properties** window, **General** tab.

3. In the **General** tab, browse to select the subprocess location by clicking the (...)button against **Sub-Process**.



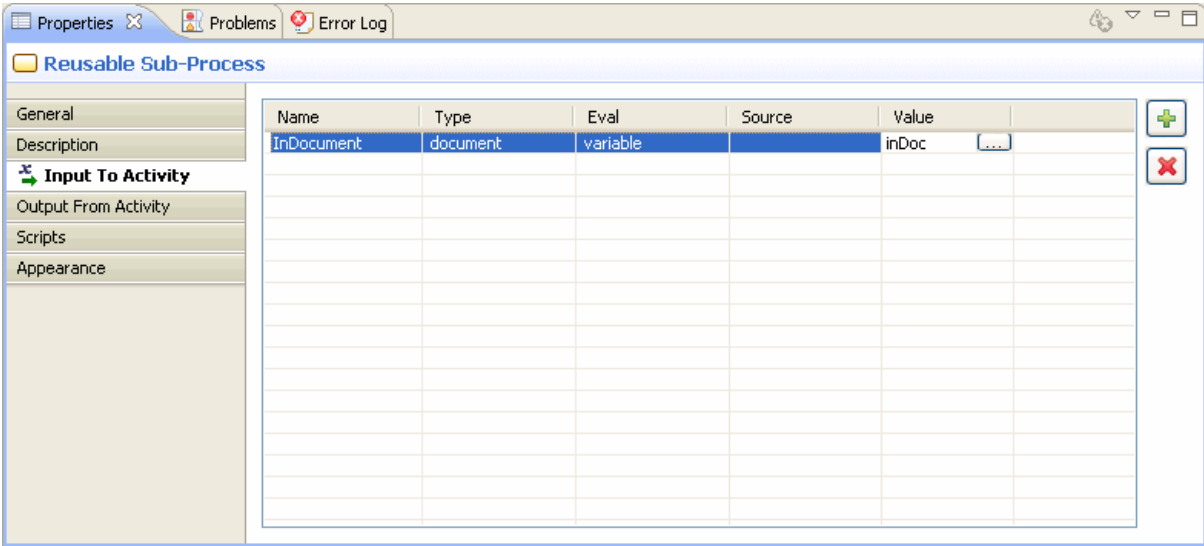
4. Click **OK**.

Defining Input Parameters

(Short description required.)

To define input parameters, go to the **Input To Activity** tab in the **Properties** window, and click the

 button. You can then provide the name and value.



To change the default eval (which is constant), go to the **Input To Activity** tab, click on the **Eval** column and select the **eval** value from the drop-down list, for instance **constant**.

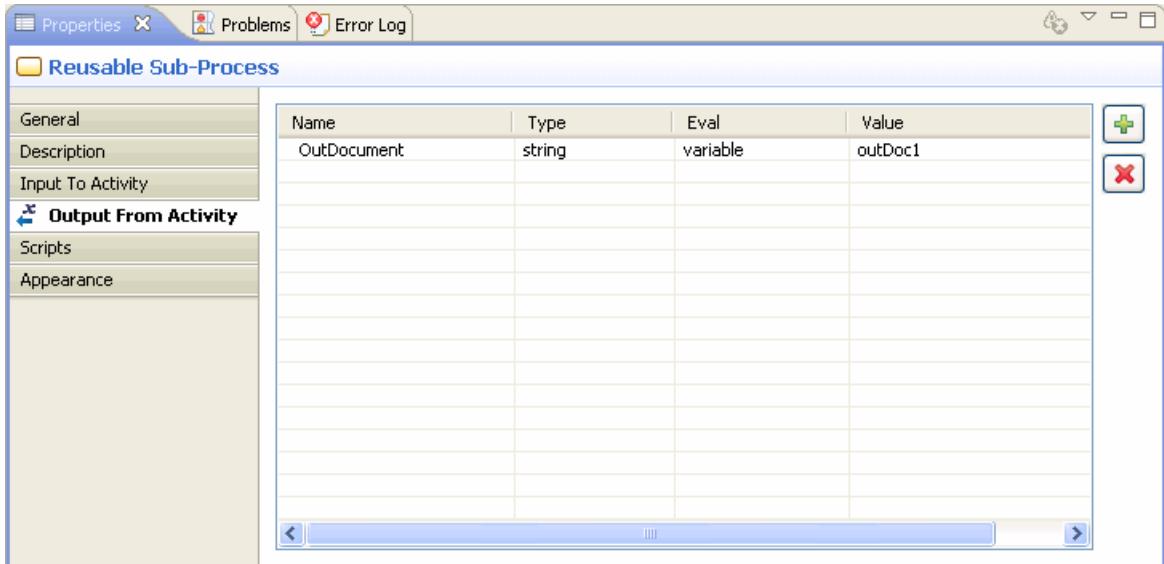
Defining Output Parameters

(Short description required.)

To define output parameters, go to the **Output From Activity** tab in the **Properties** window, and click the



button. You can then provide the name and value.



Click on **Name** column and select the value, Similarly to add the eval type, click on the **Eval** column and select the **eval** value from the drop-down list, for instance **constant**.

Initiate Subflow

(Short description required.)

The other way to use a subflow is to use the **InitiateSubflow** activity.

Activity details such as name, description, parameters can be defined as for other MDM activities, however, this method does not allow you to navigate to the subflow process from the parent process.

In case you need to use InitiateSubflow activity, it is recommended to use a clear activity naming convention so the callout to a different process is immediately clear.

Parameters

(Short description required.)

This chapter explains activity parameters in the Process Designer.

Parameters and Variables

(Short description required.)

Parameters are arguments passed into activities and outputs that are returned from activities.

Local Activity Parameters

(Short description required.)

Each MDM Activity has a predefined set of input and output parameters (mandatory and optional) that can be defined.

For more details, see:

- [Input Parameters.](#)
- [Output Parameters.](#)

Global and Local Process Parameters

(Short description required.)

You can also define global variables and local variables (data fields). Global variables are defined at the process package level, while local variables are defined at the process level.

For more details, see:

- [Defining Global Variables.](#)
- [Defining Local Variables.](#)

Global Parameters

(Short description required.)

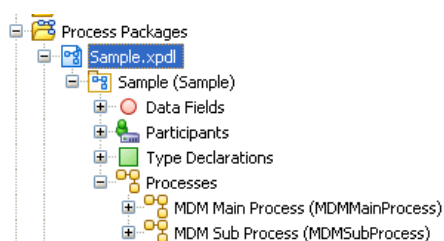
In the MDM workflow format, a set of variables can be defined within the scope of the entire workflow definition. Any activity or transition inside the workflow process has access to the global parameters. An activity typically uses a global parameter by mapping it onto a local parameter and a transition uses it in the code section of its condition.

This concept is synonymous with the Data field concept in Business Studio, and in the MDM Process Designer, global variables are defined through data fields.

Defining Global Variables

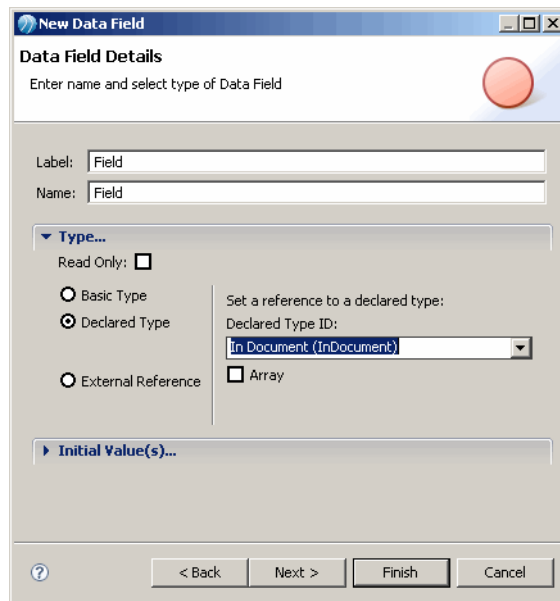
(Short description required.)

In MDM Process Designer, global variables are added by defining data fields at the package level (and not at the process level).

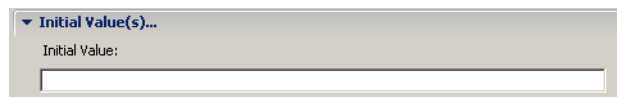


Procedure

1. To add a global variable (new data field), right-click on Package (For example, Sample) and select **New--> Data field** . This starts up the new data field wizard.
2. Define a data field for **inDocument**:
3. Enter a **label** and **name**.
4. Set the type to **Declared Type**.
5. Select **In Document** from the **Declared Type** dropdown.



6. If you want to provide an initial value to the variable, click the **Initial Value** tab and provide the value in the text box.



7. Click the **Next** button.
8. You can enter a description here. Click **Finish**.

Result

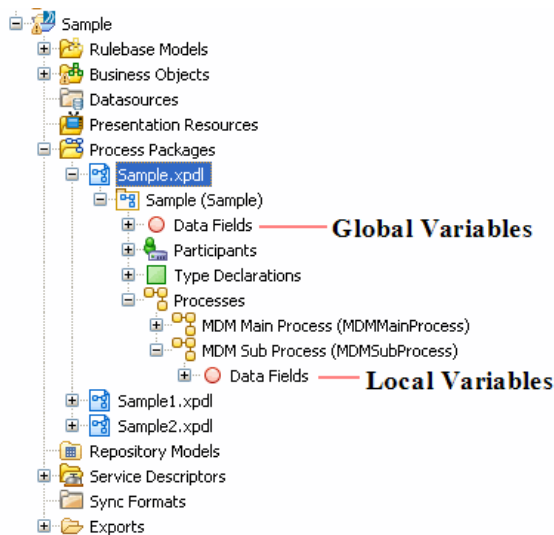
You can map parameters to global variables as required.

Defining Local Variables

(Short description required.)

Local variables are defined exactly the same way as global variables, except that these are defined at the process level.

To add a local variable (new data field), right-click Data Fields (contained within the process), and select **New-->Data field**. Follow the wizard to define the local variable.



Input Parameters

(Short description required.)

Defining Values for Input Parameters

(Short description required.)

You need to map input parameters to provide values for them.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties** Window, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Click corresponding to the parameter you want to provide a value and type the value in the text field.

Result

Each MDM Activity type (Action) has a set of mandatory and optional parameters. Depending on the eval mode selected, the parameters varies. The MDM workflow accepts the following eval modes:

Eval Mode	Description
Constant	A constant value assigned to the parameter, need not to define source attribute of the parameter
Variable	Supplied a value from the global workflow state
XPath	You need to define source attribute to evaluate parameter value. Source should be the XPath expression to be evaluated.
Rule	The source attribute should point to a valid business process rule name.
Property	The property evaluation executes a get method on the specified object.

Eval Mode	Description
Lookup	This looks up the value in the DOMAINVALUE table for a given ownerID, domaintype, domainvalue. If no ownerxpath is specified, the ID of the context organization is used.
Catalog	You need to define source attribute to evaluate parameter value. Source attribute is the property of the catalog. For example, SourceOrganizationID, VersionOption, IsXMLFormat, FileGenerationOption, MasterCatalogID, MasterCatalogVersionDateTime, CatalogID, CatalogName, CatalogDescription, OutputMapID, ChannelID, ClassificationSchemeID, SubsetRuleID, TransformRuleBase, TransferMode, ItemAdd, ItemDelete and so on. For more information, refer to <i>TIBCO MDM Workflow guide</i> .
Compute	<p>COMPUTE is a generic way to calculate work item expiry date based on record attributes. The Rulebase parameter should be defined; it will be used to compute the target expiry date. The actual expiry date will be expiry date</p> <p>= Launch Date(output from rulebase) - RemindBeforeNumberOfDays</p> <p>Based on the value of ReminderNumber (reminders generated so far), the 'Reminder Setup' rule can be configured to get values of RemindBeforeNumberOfDays and email addresses to send the reminder email to. Rules should be set such that for each Reminder number, only one value for RemindBeforeNumberOfDays would be returned (although any number of email addresses can be returned). When a record is edited or a work item revived, expiry date is recomputed and updated into the work item. If the recomputed expiry date differs from the old expiry date, the ReminderNumber is set to zero.</p>
Event	<p>The Event eval is used to get attribute value of the event. You need to define event attribute as a source to evaluate parameter value. For example, Event Descriptor, Event ID, Event Date, Event State, Event Status, Parent Event ID, and Process ID.</p> <p>The recommended data type to be used is, for Event Descriptor is string, Event ID is long, Event Date is date, Event State is string, Event Status is string, Parent Event ID is long, and Process ID is long</p> <p>If the recommended data type is not used, the system will convert the data type to the specified data type. For example, if EventDate data type is specified as "string", data will be output as string instead of date.</p>

Eval Mode	Description
UserProfile	<p>The UserProfile eval is used to get attribute value of the user profile. You need to define user profile attribute as source to evaluate parameter value. For example, Enterprise ID, Enterprise Internal Name, Enterprise Name, Enterprise Vertical, Formatted User Name, Organization ID, Organization Name, Organization Type, User ID, and User Name.</p> <p>The recommended data type to be used is, for Enterprise ID is long, Enterprise Internal Name is string, Enterprise Name is string, Enterprise Vertical is string, Formatted User Name is string, Organization ID is long, Organization Name is string, Organization Type is string, User ID is long, and User Name is string.</p> <p>If the recommended data type is not used, the system will convert the data type to the specified data type. For example, if EnterpriseID data type is specified as "string", data will be output as string instead of long.</p>
System	<p>The System evaluation is used to get attribute value of the system. You need to define system attribute as source to evaluate parameter value. For example, Common Directory, Home Directory, Host Address, Host Name, Log Directory, Node ID, System Date, System Timestamp, and System Timestamp Long.</p> <p>The recommended data type to be used is, for Common Directory is string, Home Directory is string, Host Address is string, Host Name is string, Log Directory is string, Node ID is string, System Date is date, System Timestamp is Timestamp, and System Timestamp Long is long</p> <p>If the recommended data type is not used, the system will convert the data type to the specified data type. For example, if System Date data type is specified as "string", data will be output as string instead of date.</p>

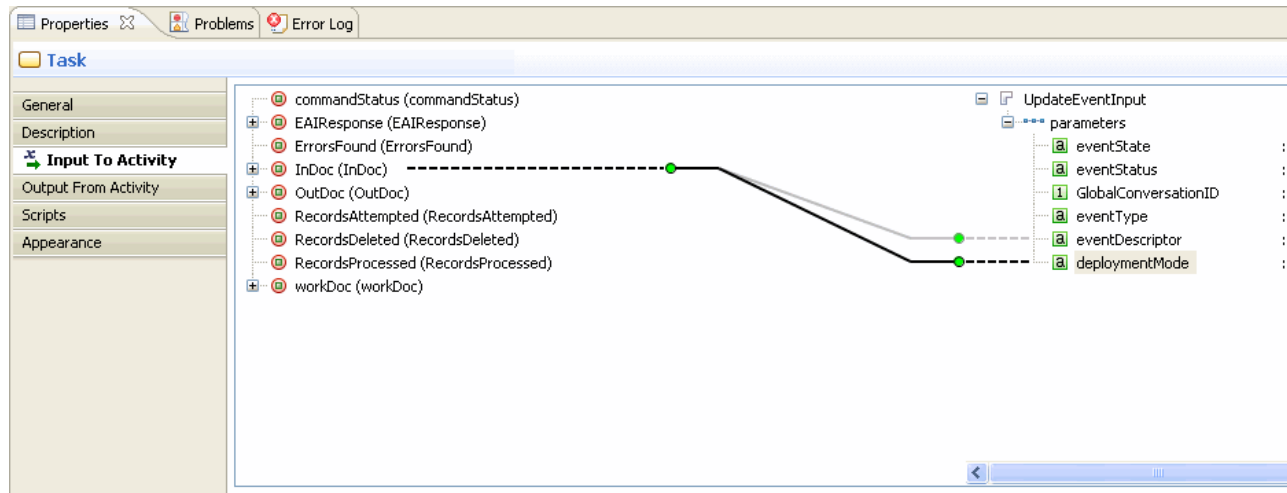
Example 1 - Setting the eventState

(Short description required.)

For instance, if you want to set the eventState to START:

Procedure

1. Select the eventState parameter under **parameters**.
2. Click on the text field corresponding to the eventState parameter and type START.



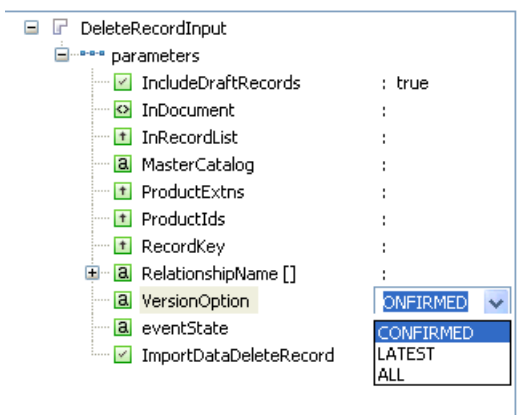
Example 2 - Setting the VersionOption

(Short description required.)

For instance, if you want to set the VersionOption to LATEST:

Select the VersionOption parameter under **parameters**.

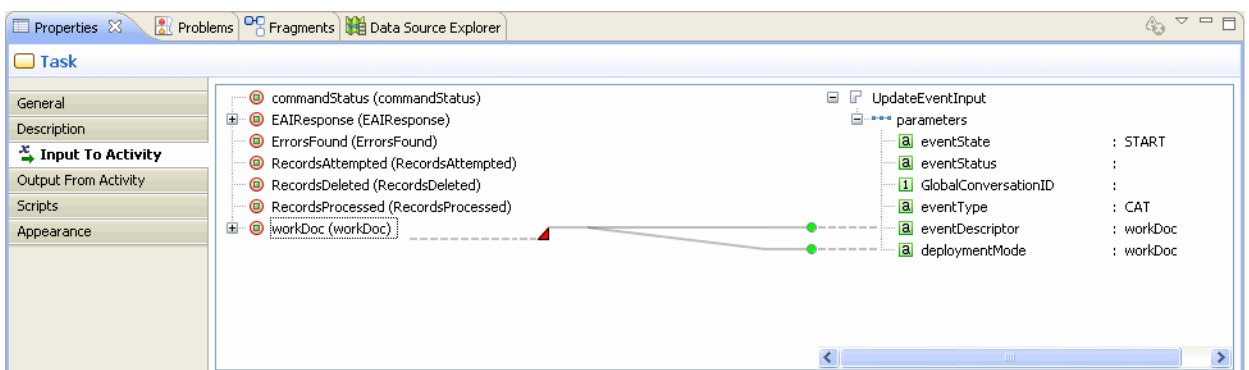
Click on the drop-down list corresponding to the VerisonOption parameter and select LATEST.



Viewing Input Parameters

(Short description required.)

The interface to add input parameters is displayed if you click a MDM Activity, and then click the **Input to Activity** tab in the **Properties Window**.



A complete list of parameters for the selected activity displayed to the right of the dialog.

The list of input parameters to the selected activity are listed under **parameters**. Each element type is also marked - for instance, **eventState: START**.

Output Parameters

(Short description required.)

Defining Values for Output Parameters

(Short description required.)

Procedure

1. Select the activity in the process flow.
2. In the **Properties Window**, go to the **Output From Activity** tab.
3. Expand the <activityname> output parameters on the left.
4. Select the parameter that you want to provide a value to.
5. Drag it onto a data field on the right to map it.

Result

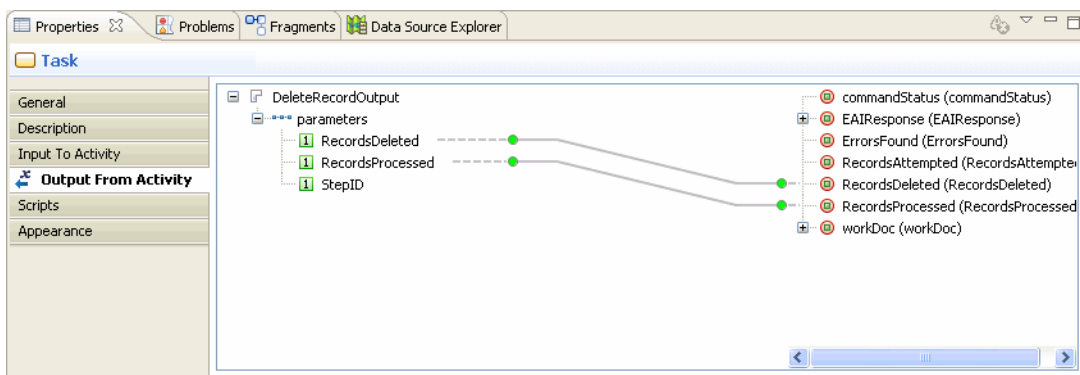


Output parameters can only be mapped to Data Fields.

Viewing Output Parameters

(Short description required.)

The interface to view output parameters is displayed if you click a MDM Activity, and then click the **Output From Activity** tab in the **Properties Window**.



Add Parameters

(Short description required.)

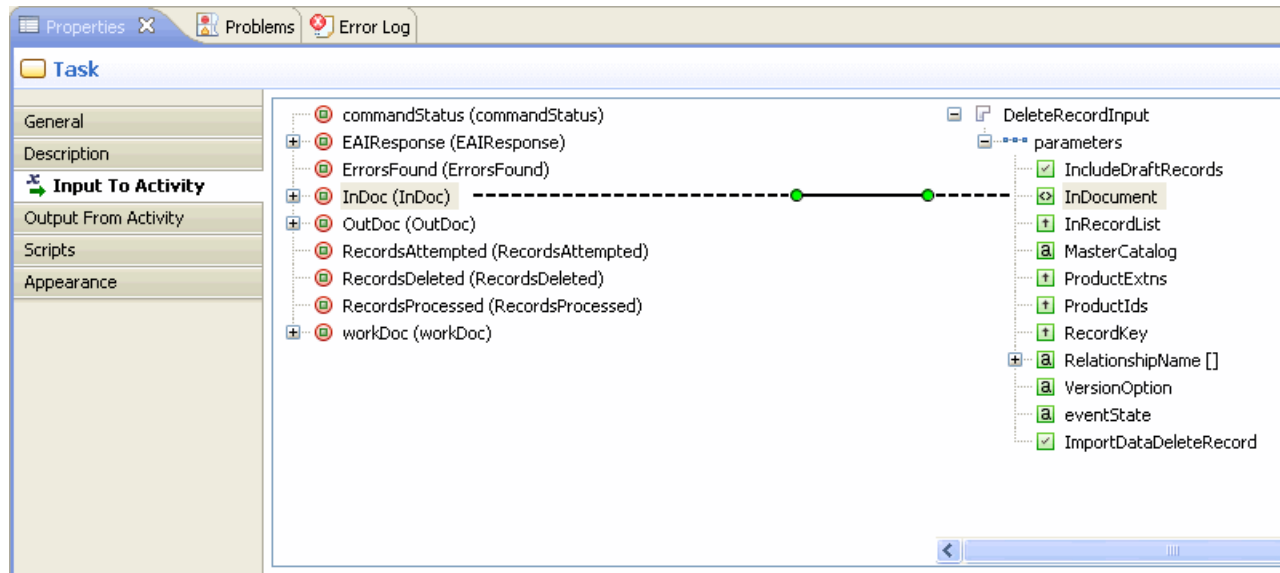
Adding Parameter for eval mode=Variable

(Short description required.)

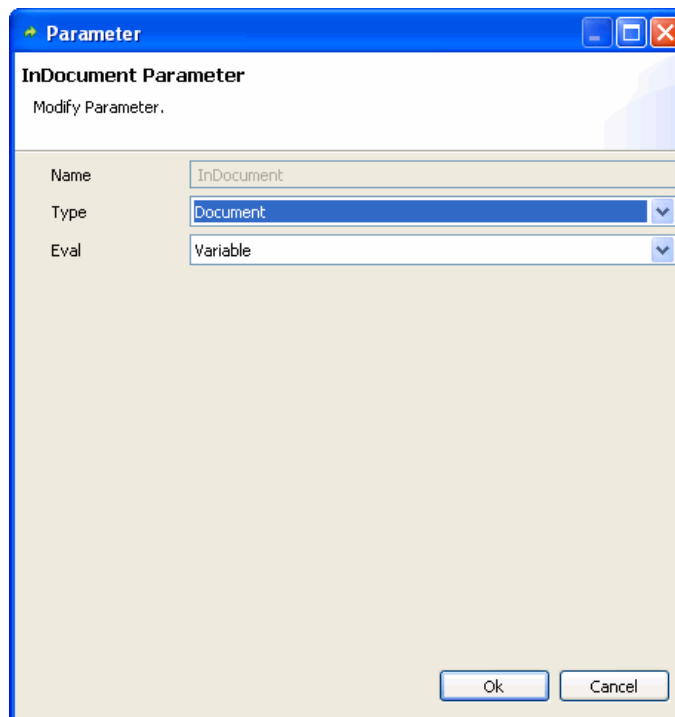
A Data Field variable can be mapped to data types such as `StringType`, `BoolType`, `LongType`, and so on. For example, the screenshot below shows an Document type data field mapped to the variable type `InDocument` parameter of the `DeleteRecordInput` activity.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties** Window, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField (Workflow state variable) on the left.



5. Double-click on parameter to view or change the parameter default type.



6. Click **OK**.

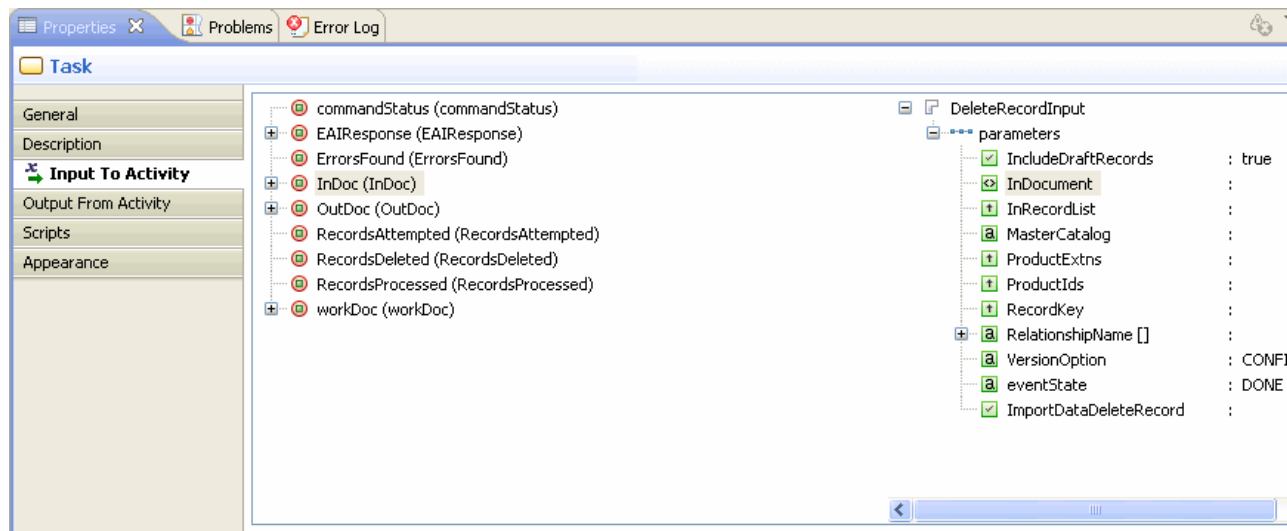
Adding Parameter for eval mode=Constant

(Short description required.)

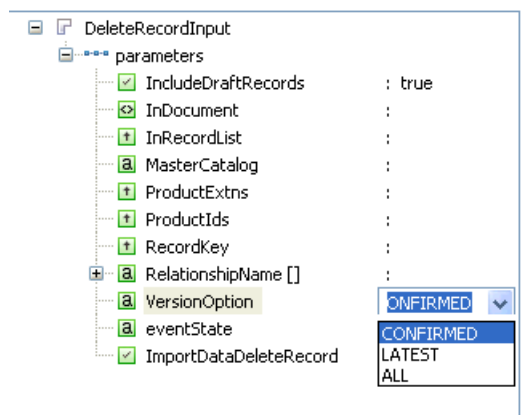
Constant Strings are mapped with a literal expression containing 'Confirmed' or 'Latest' or 'All'. For example, the screenshot below shows the string type VersionOption from the DeleteRecordInput activity mapped to a String constant set to LATEST.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Select the parameter that you want to provide a value to and type in constant value in the text box against parameter.



5. For enumeration parameter a drop-down appears for the parameter.



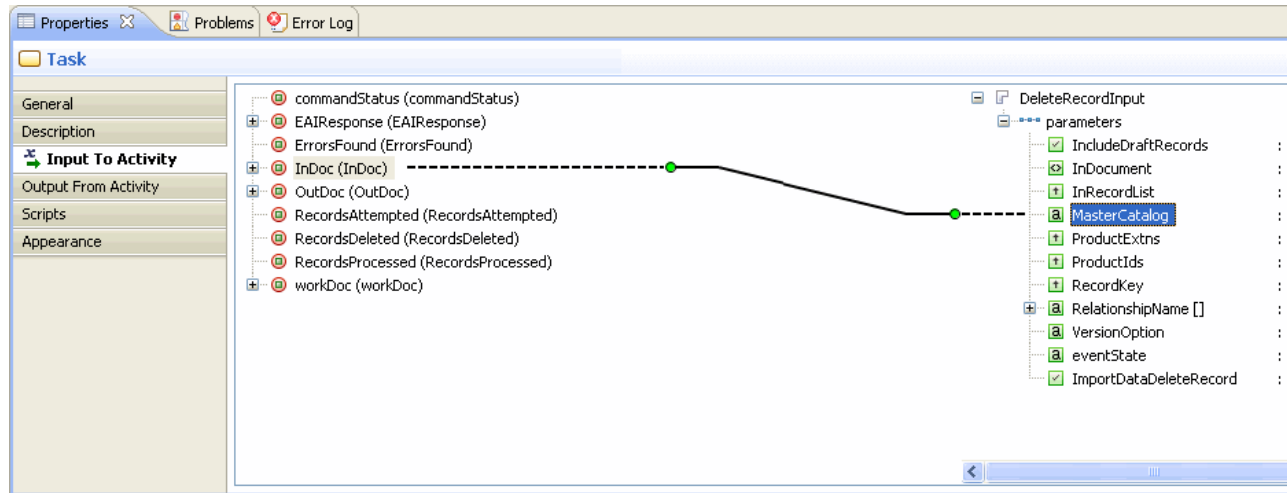
Adding Parameter for eval mode=XPath

(Short description required.)

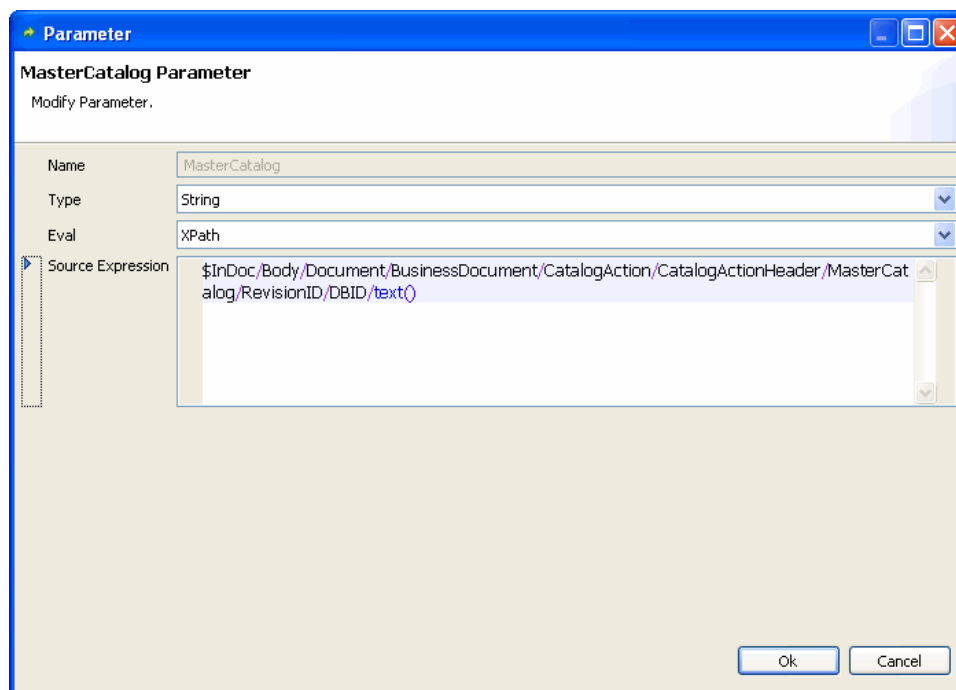
Procedure

1. Select the activity (that you want to define parameters for) in the process flow.

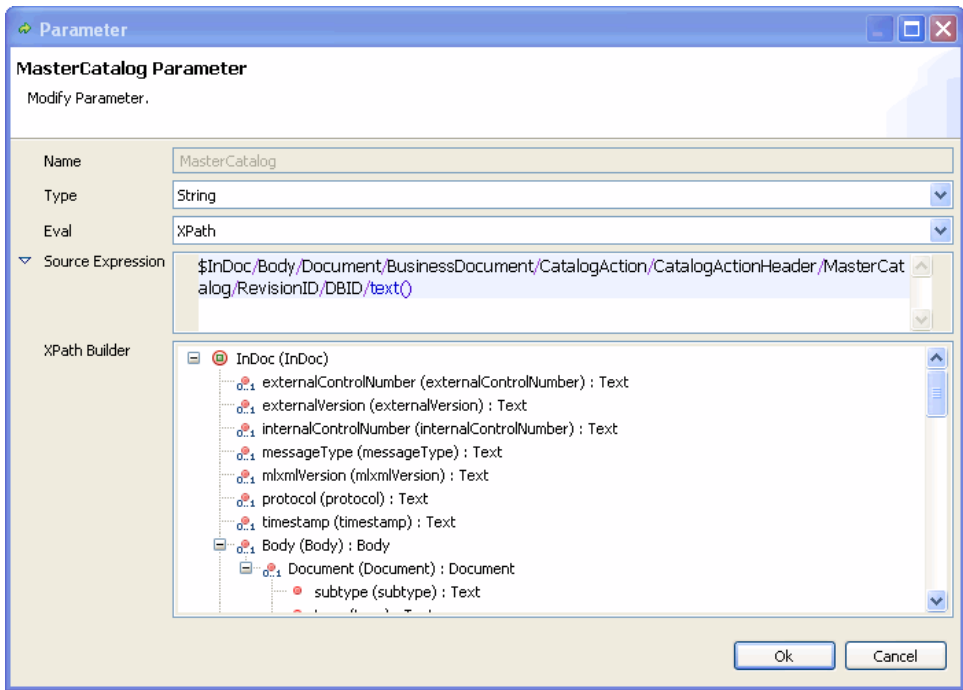
- In the **Properties Window**, go to the **Input to Activity** tab.
- Expand the <activityname> input parameters on the right.
- Select the parameter that you want to provide a value to and drag it onto DataField (mXMLDocument type) on the left.



- Double-click on the parameter and select **XPath** from the **Eval** drop-down list. Use the **Source Expression** to write the xpath expression.



- Use the **XPath Builder** to write the XPath expression.



7. Click **OK**.

Result

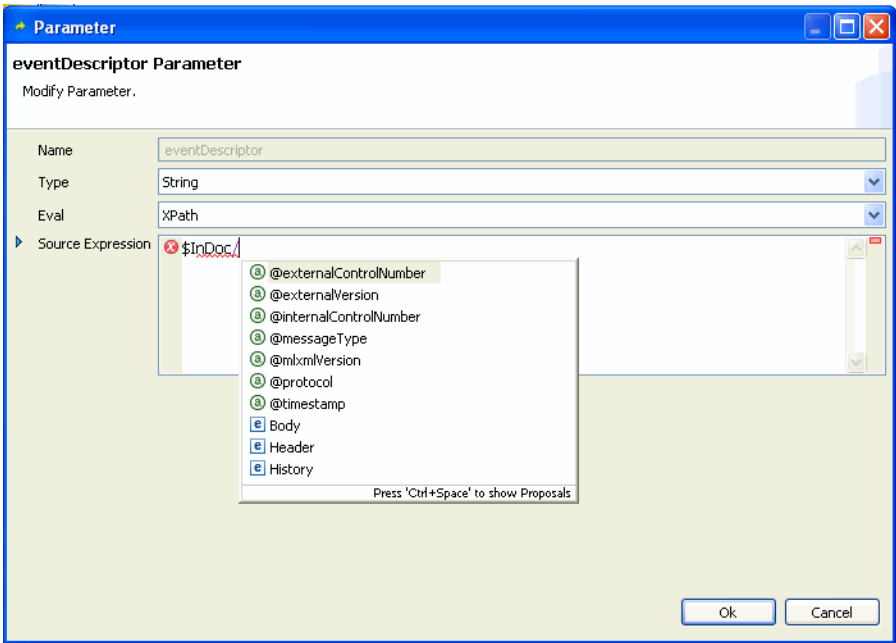


Use the '\$' character followed by the variable name (mapped data field) to access the mXMLDocument.

Using Content Assist

(Short description required.)

You can use the content assist when building your Xpath source expression. For instance, if you go to the Xpath source expression and press '**Ctrl+Space**' you get a list of all the XPath proposals that you can select from.



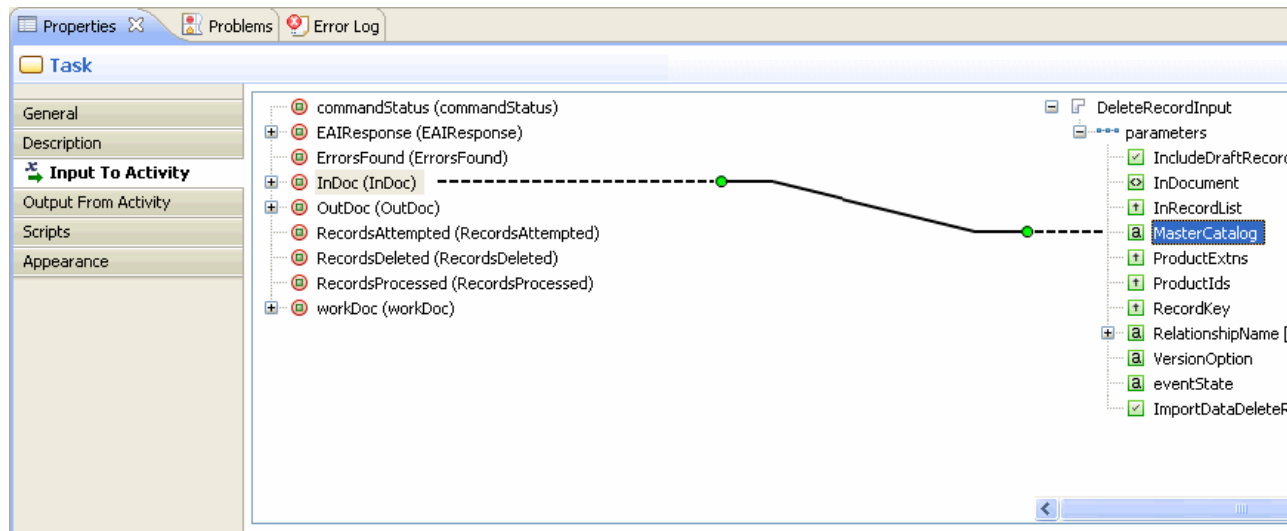
Adding Parameter for eval mode=Rule

(Short description required.)

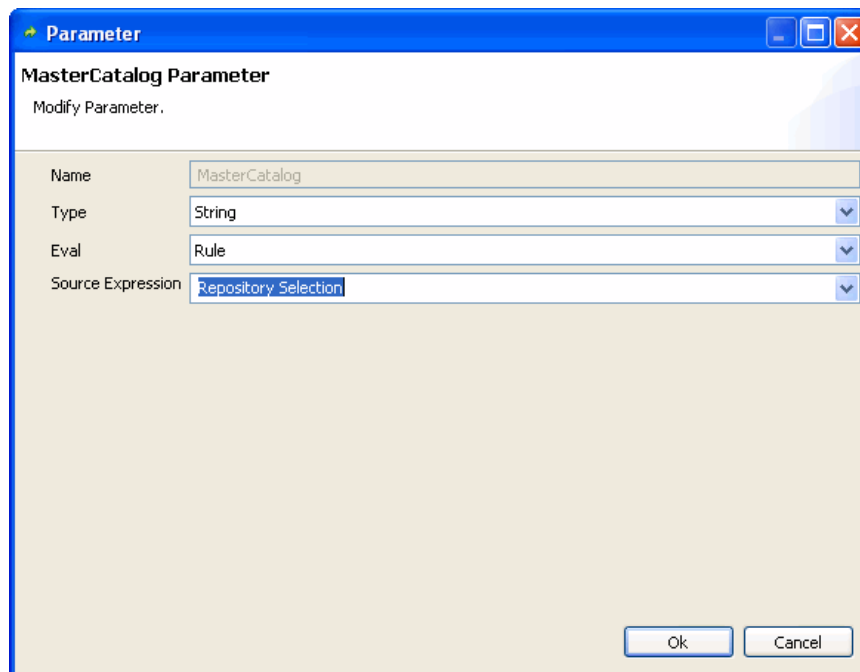
A business rule is assigned by mapping a literal 'rule' into a parameter's **eval** attribute followed by mapping the rule name as a literal into the parameter's **source** attribute.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField on the left.



5. Double-click on the parameter and select **Rule** from the **Eval** drop-down list. Select a rule from the **Source expression** drop-down list.



- Click **OK**.

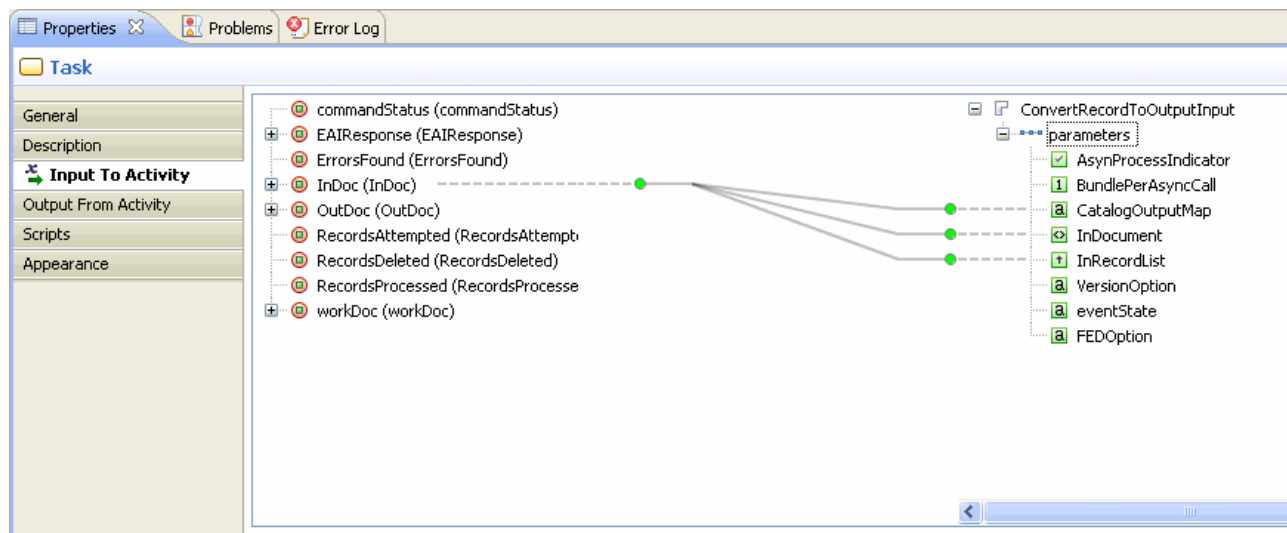
Adding Parameter for eval mode=Lookup

(Short description required.)

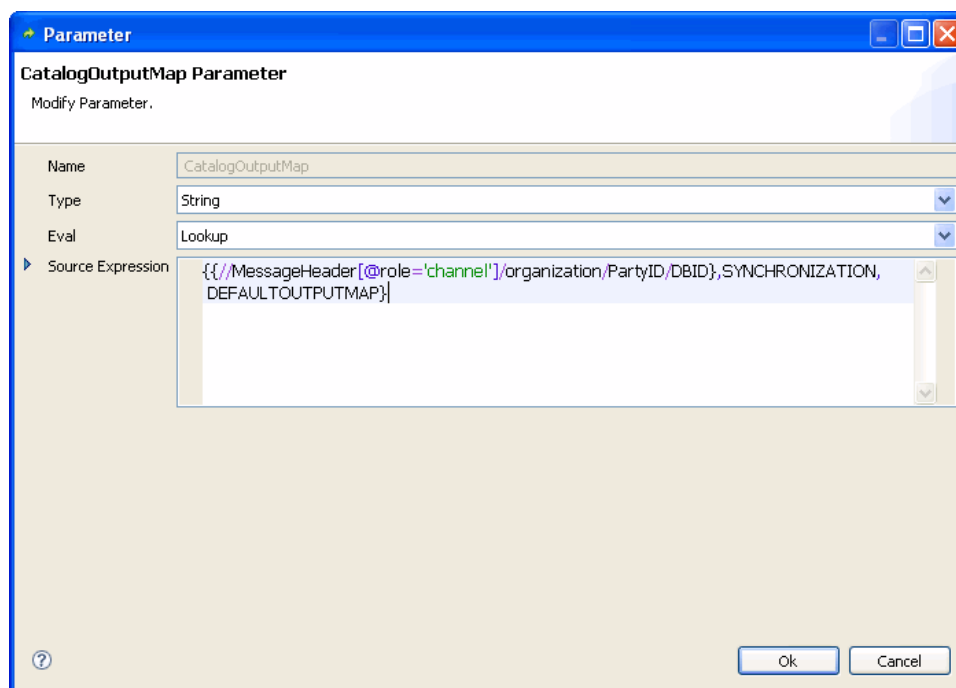
A lookup is assigned by mapping a literal 'lookup' to the eval attribute of a parameter, and mapping the source of the parameter to the required value, and the parameter itself to the required value.

Procedure

- Select the activity (that you want to define parameters for) in the process flow.
- In the **Properties Window**, go to the **Input to Activity** tab.
- Expand the <activityname> input parameters on the right.
- Select the parameter that you want to provide a value to and drag it onto DataField on the left.



- Double-click on the parameter and select **Lookup** from the **Eval** drop-down list.



- Click **OK**.

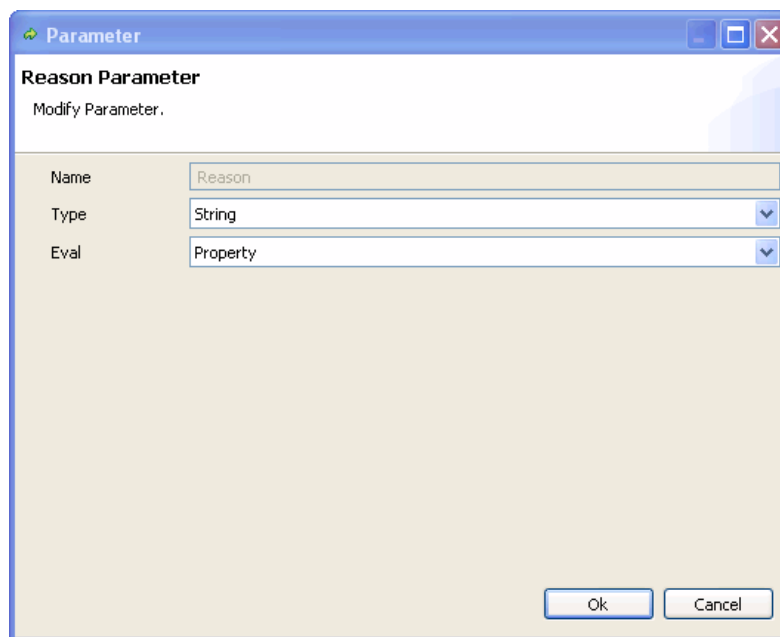
Adding Parameter for eval mode=Property

(Short description required.)

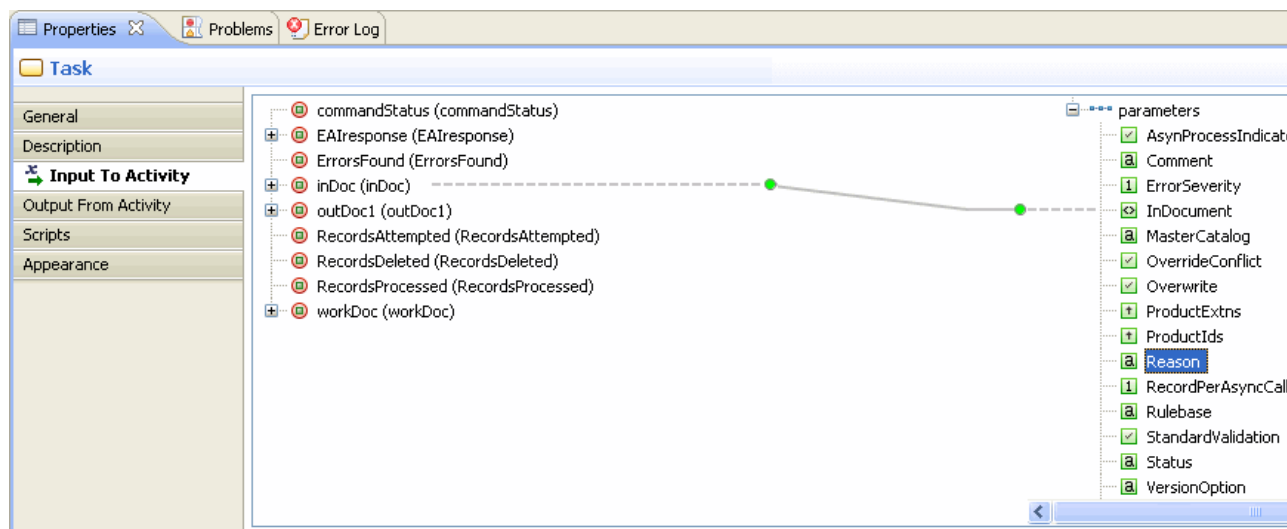
A property is assigned by mapping a literal 'property' into a parameter's **eval** attribute followed by mapping the property name to the value of the parameter.

Procedure

- Select the activity (that you want to define parameters for) in the process flow.
- In the **Properties Window**, go to the **Input to Activity** tab.
- Expand the <activityname> input parameters on the right.
- Double-click on the parameter and select **Property** from the **Eval** drop-down list.



- Select the parameter and enter the property value in the text box against parameter.



- Click **OK**.

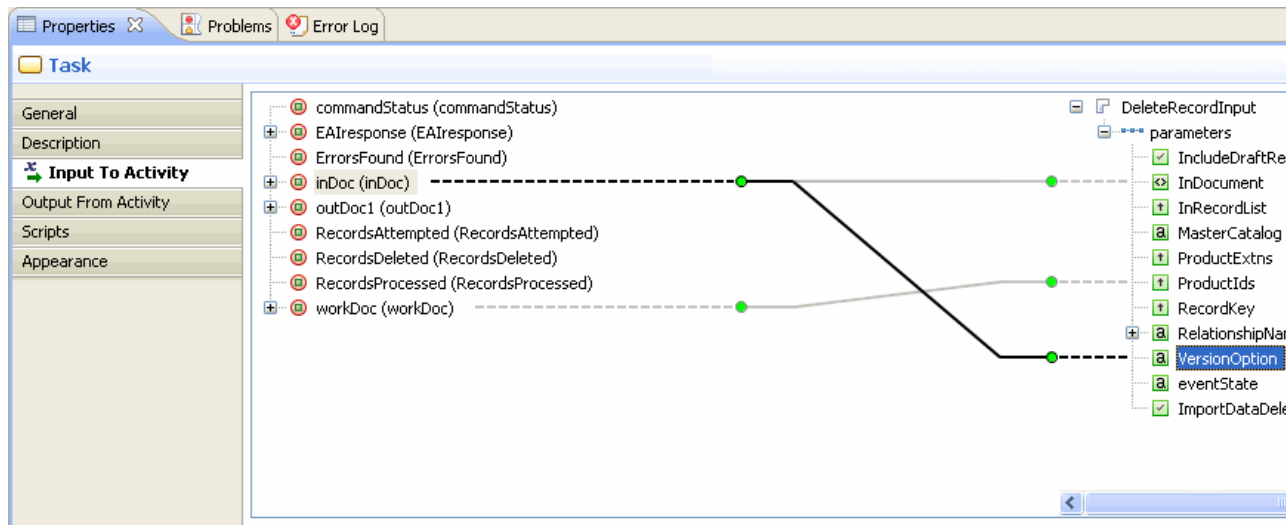
Adding Parameter for eval mode=Catalog

(Short description required.)

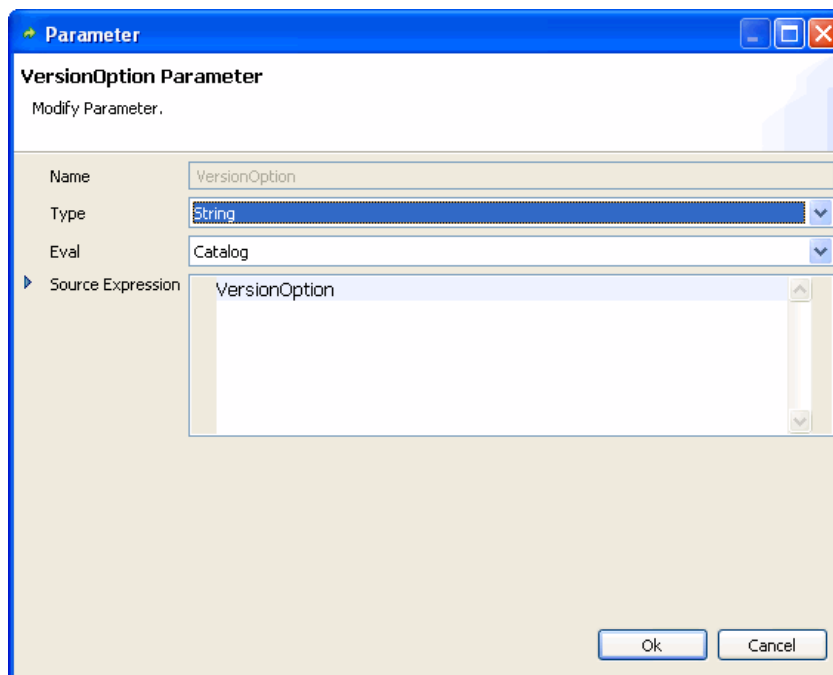
A catalog is assigned by mapping a literal 'catalog' to the eval attribute of a parameter, and mapping the source of the parameter to the required value, and the parameter itself to the required value.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField on the left.



5. Double-click on the parameter and select **Catalog** from the **Eval** drop-down list.



6. Enter the source expression in the source expression editor and click **OK**.

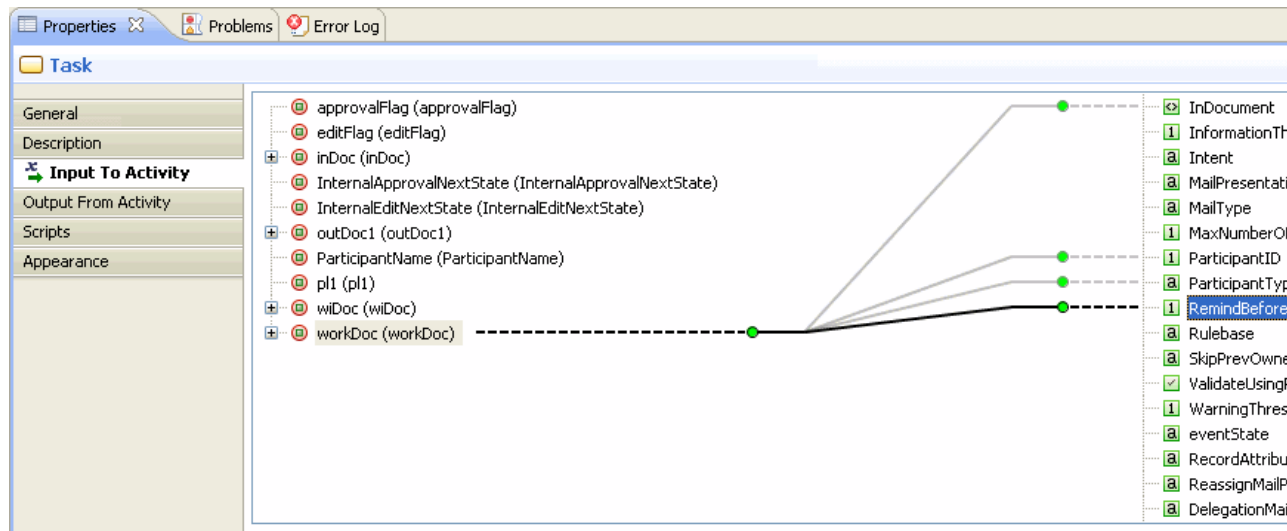
Adding Parameter for eval mode=Compute

(Short description required.)

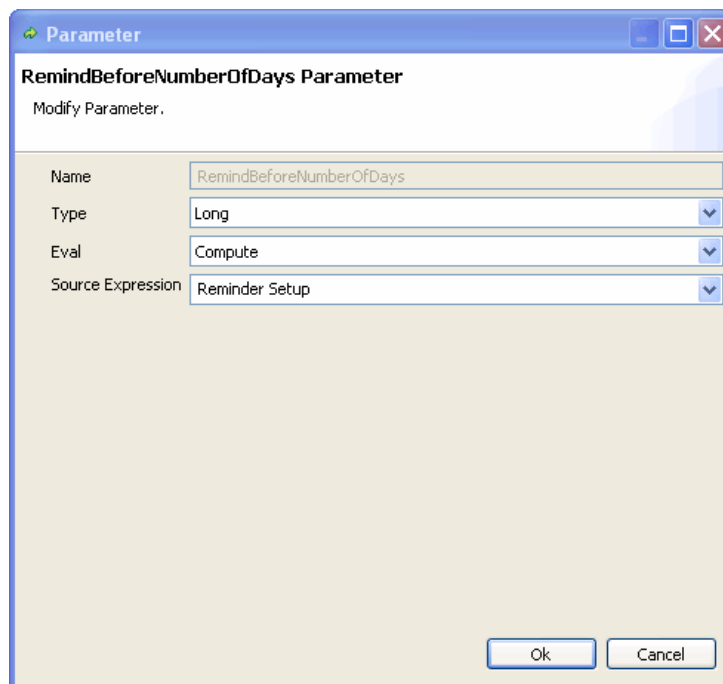
A compute is assigned by mapping a literal 'compute' to the eval attribute of a parameter, and mapping the source of the parameter to the required value, and the parameter itself to the required value.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField on the left.



5. Double-click on the parameter and select **Compute** from the **Eval** drop-down list.



6. Click **OK**.

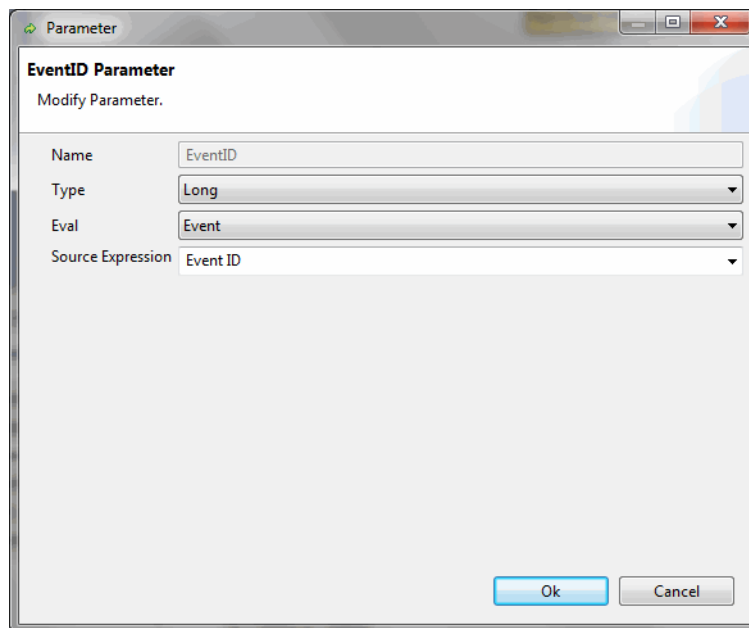
Adding Parameter for eval mode=Event

(Short description required.)

A event attribute is assigned by selecting Event from Eval drop-down list followed by selecting attribute from Source drop down.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Double-click on the parameter (for example, EventID), it will open Parameter dialog window. Select **Event** from the **Eval** drop-down list.
5. Select the appropriate value from the **Source Expression** drop-down list.



6. Click **OK**.

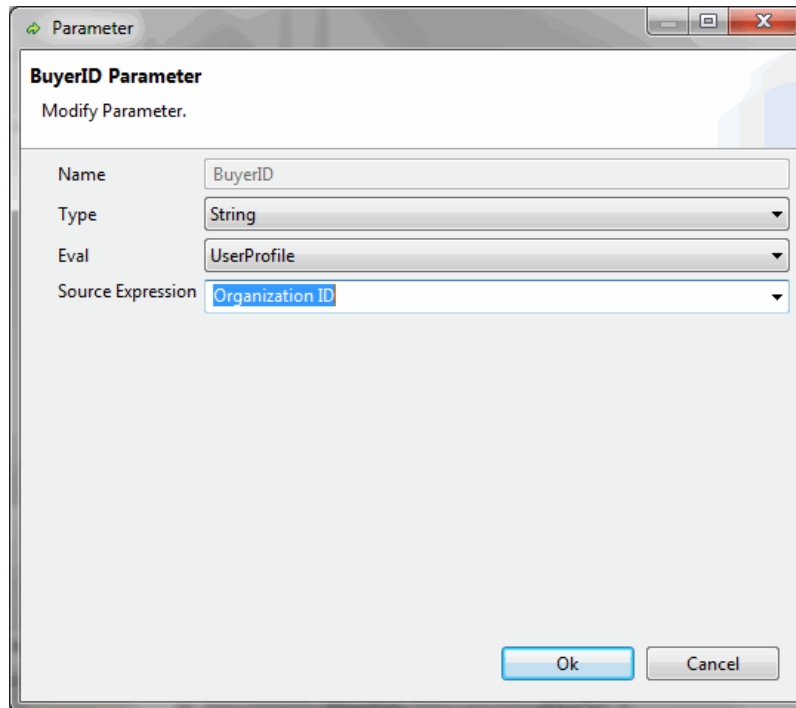
Adding Parameter for eval mode=UserProfile

(Short description required.)

A user profile attribute is assigned by selecting User Profile from Eval drop down list followed by selecting attribute from Source drop-down.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Double-click on the parameter (for example, BuyerID), it will open Parameter dialog window. Select **UserProfile** from the **Eval** drop-down list.
5. Select the appropriate value from the **Source Expression** drop-down list.



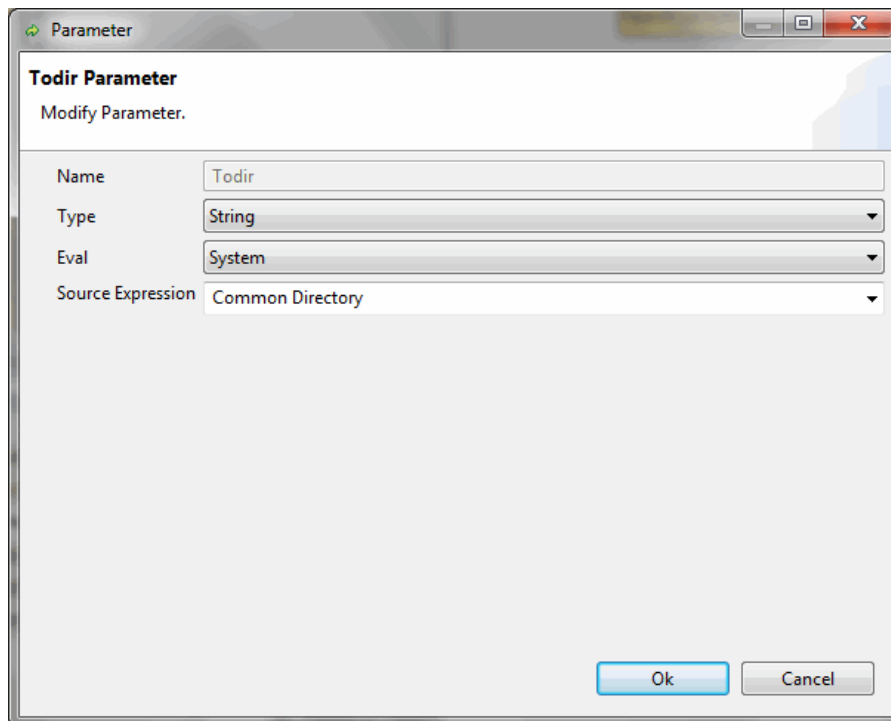
6. Click OK.

Adding Parameter for eval mode=System

A system attribute is assigned by selecting System from Eval drop down list followed by selecting attribute from Source drop-down.

Procedure

1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties Window**, go to the **Input to Activity** tab.
3. Expand the <activityname> input parameters on the right.
4. Double-click on the parameter (for example, Todir), it will open Parameter dialog window. Select **System** from the **Eval** drop-down list.
5. Select the appropriate value from the **Source Expression** drop-down list.



6. Click OK.

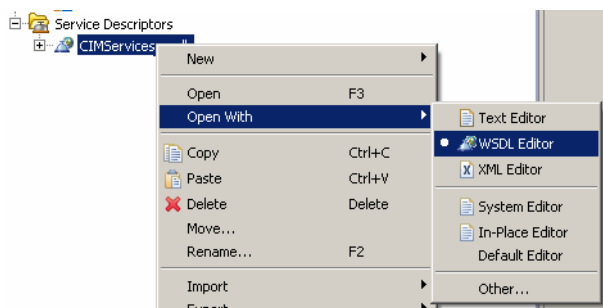
Adding and Modifying Custom Parameters

To add input parameters to an activity, you need to modify the WSDL (this is because the WSDL is the basis on which all the parameters are defined to a web service operation).

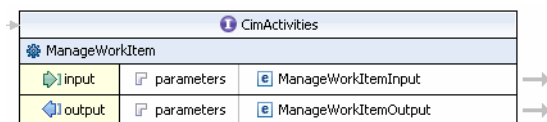
To add or modify a parameter, use the WSDL editor supplied with the MDM Process Designer.

Procedure

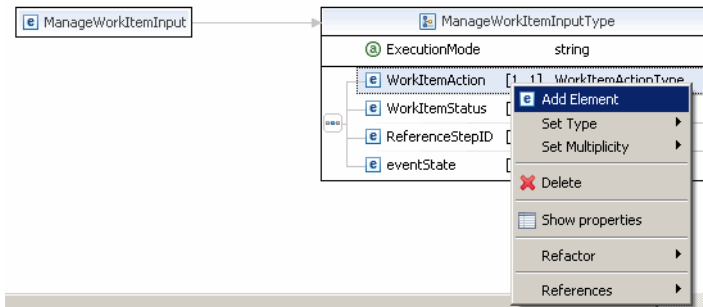
1. Right click **CIMServices.wsdl** in the Project Explorer and select **Open With -> WSDL Editor**. This opens CIMServices.wsdl in design mode.



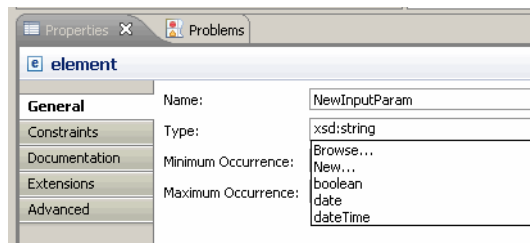
2. Go to a MDM Activity and click the arrow symbol next to <ActivityName>Input.



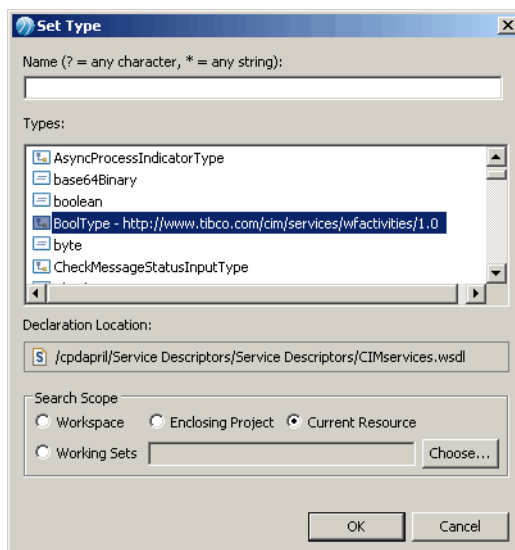
3. Right click the list of elements and select **Add Element**.



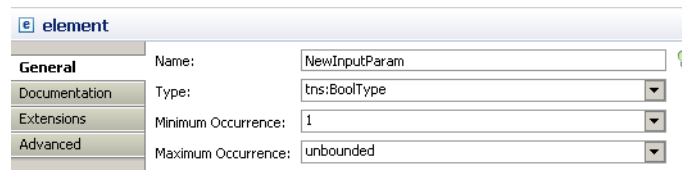
4. Select the new element and go to the **Properties** tab.
5. Change the element name to the new input parameter name and select **Browse** from the Type dropdown.



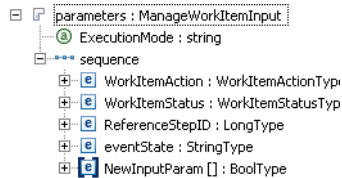
6. In the Set Type dialog that is displayed, select one of the following types and click **OK**.
 - BoolType
 - LongType
 - StringType
 - RecordListType
 - ArrayListType
 - mlXMLDocumentType
 - DateType
 - FileType
 - TimestampType



7. Select the **Minimum Occurrence** and **Maximum Occurrence**. If minimum occurrence is zero, the parameter is optional and if minimum occurrence is one, the parameter is mandatory.



8. Once you save the wsdl, the added parameters will also be saved.
9. When you drag and drop the modified MDM Activity to the drawing pane, and view the parameters in the General tab, you will see the newly added parameter.



Tooltip Parameter Descriptions

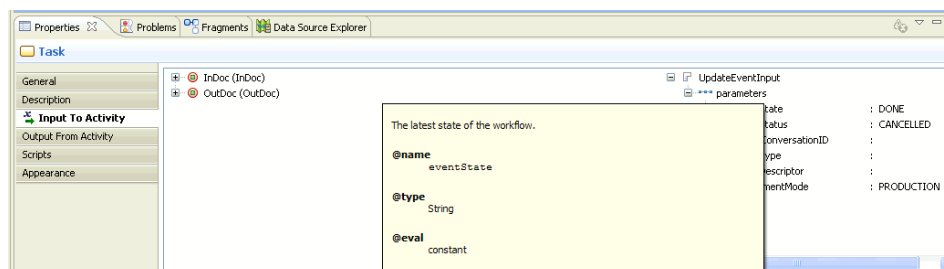
(Short description required.)

For each service task an activity action is defined. Each activity action has input and output parameters. The input parameters are available in Input to Activity tab whereas the output parameters are available in Output from Activity tab. On mouse hovering on these parameters, a tooltip with the parameter descriptions are displayed.

Parameter Descriptions properties file

(Short description required.)

The parameter description are externalized and located in <Installation Directory>\eclipse-platform \bundlepool\1.0\org.eclipse.equinox.p2.touchpoint.eclipse\plugins \com.tibco.xpd.cim.properties_4.0.0.xxx\com\tibco\xpd\cim\properties\tooltip \parameterdescriptions.properties file.



The parameter descriptions properties file consists of common parameters and activity specific parameters. It contains of a set of key value pairs which is stored as a resource. Each key uniquely identifies a property and the value represents a description of the parameter in the local language.

There are two types of keys used in parameter descriptions properties file

Key for common parameters: The common parameters are common to several activities. For example, AsynProcessIndicator and InDocument are the common parameters. The parameter name is used as a key for storing common parameters description.

For example,

the InDocument will be the key for the InDocument parameter. There can currently be only one document.



The property will be represented in local language (For example, English) as InDocument=The document to process in the workflow.

Key for activity specific parameters: The key for an activity specific parameter is formed by concatenating the activity action type and the parameter name separated by a period namely, <Action type>.<Parameter name>.

For example,

If the parameter name is ParticipantName and the action type is CreateWorkItem, the key will be CreateWorkItem.ParticipantName.



The property will be represented in local language (for example, English) as CreateWorkItem.ParticipantName= Specifies the name of participants for whom workitems were created.

You can add or edit the parameter descriptions for an existing or new parameter in the parameter descriptions properties file.

Adding a new parameters description

(Short description required.)

To add a new parameter description in properties file:

Procedure

1. Navigate to <Installation Directory>eclipse-platform\bundlepool
 \1.0\org.eclipse.equinox.p2.touchpoint.eclipse\plugins\
2. Open the compressed archive file com.tibco.xpd.cim.properties_4.0.0.004.jar with a zip file tool such as WinZip.
3. Take backup of com.tibco.xpd.cim.properties_4.0.0.004.jar file.
4. Extract the com.tibco.xpd.cim.properties_4.0.0.xxx\com\tibco\xpd\cim\properties\tooltip\parameterdescriptions.properties file.
5. Using any text editor, open the parameter description properties file and add new parameters description as described in Parameter Descriptions properties file section.
6. Compress the parameter description properties file using any zip file tool such as Winzip and save it with the same name as original namely com.tibco.xpd.cim.properties_4.0.0.004.jar. This will overwrite the earlier properties jar file.
7. Restart TIBCO MDM Studio and view the newly added parameter description.

Transitions

(Short description required.)

This chapter explains transitions in MDM Process Designer.

Transition Types

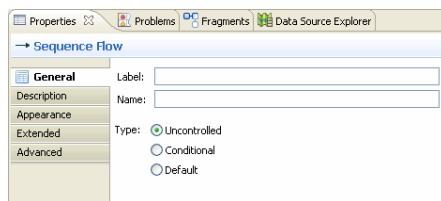
(Short description required.)

Local transitions simply connect one activity to another; they represent a link between two activities to indicate that when one activity completes executing, the next one should start. Local transitions can be either simple or conditional.

Simple Transitions

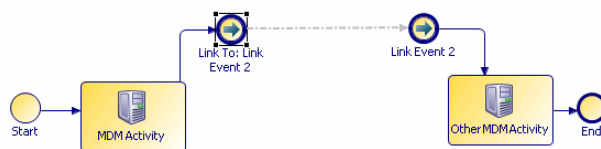
(Short description required.)

A simple transition is not governed by any conditions, and it unconditionally connects one activity to another. In a simple transition, the **Sequence Flow** is set to type **Uncontrolled**.



Use link events to avoid drawing long transitions which may overlap with other items on the drawing pane. You can add link events from the palette (Link Intermediate Event: Link to/from other Link Event).

So for instance, if you want to link ActivityX and ActivityY, you can create a link event under each activity, connect both link events to each other and each of the events to one of the activities.



Conditional Transitions

(Short description required.)

Conditional transitions move to a specific activity depending on whether or not a specified condition has been met.

For more information on conditional transitions and creating them, refer [Conditional Transitions](#).

Parallel Transitions

(Short description required.)

Parallel transitions refer to transitions that execute simultaneously; from a given activity or depending on the outcome of a given activity, you can have more than one activity executing in parallel.

For example, consider the case of a workflow which functions as follows:

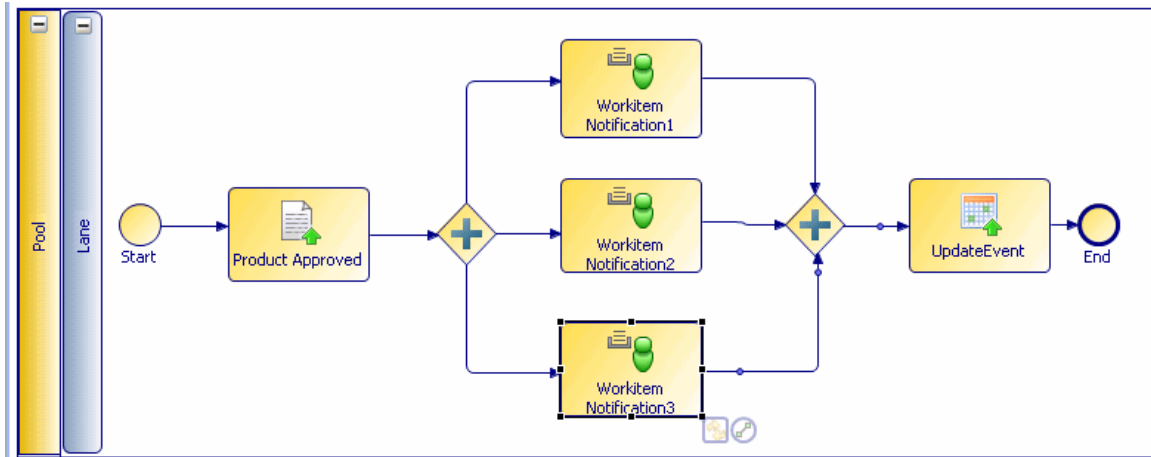
Once the status of a product is set to "approved", workitem notifications need to get sent to three different people with different roles. Each of these notifications are created in parallel, and after the necessary action is taken by all three, the workflow proceeds to the next task.

The following figure illustrates this example with a task in the process forking out into three parallel activities and then all three joining together into a single thread of execution.



MDM always requires a start activity for a parallel process and an end activity to join the parallel process.

Parallel Transitions



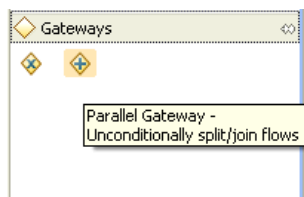
Creating a Parallel Transition

(Short description required.)

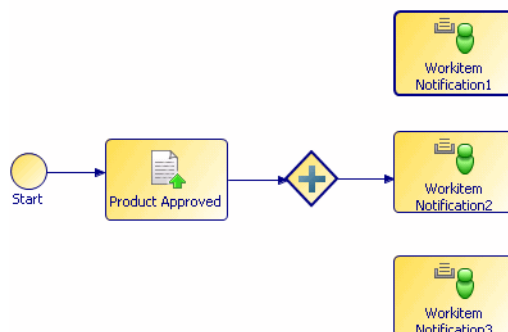
To create a parallel transition, you need to use **Parallel Gateway: Unconditionally split/join flows** to link one activity to two or more activities. To do this:

Procedure

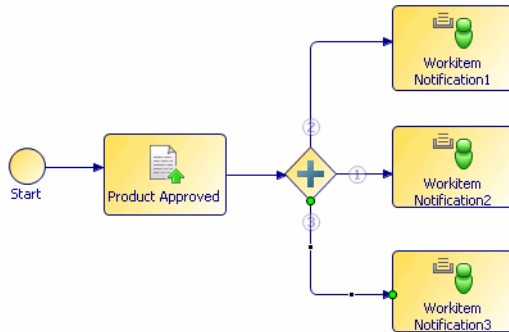
1. Create an activity **ProductApproved** and create three workitem notification activities.
2. Next, select **Parallel Gateway: Unconditionally split/join flows** in the Palette.



- Place it between the **ProductApproved** activity and one of the workitem notification activities (this represents the split for the parallel transition).



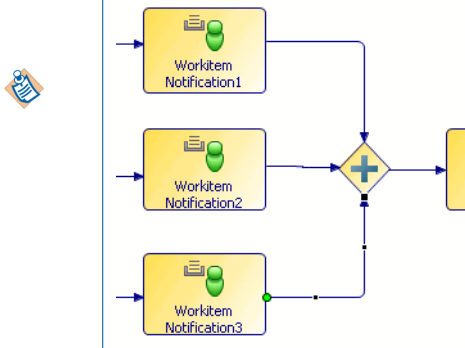
3. Use the sequence flow to link the gateway with the other **Workflow Notification** tasks.



4. Create another Parallel Gateway (Join) sequentially after the **Workflow Notification** tasks and use the **Sequence flow** to connect each of the Workflow Notification tasks with it - this represents the point where the three parallel transitions join into a single thread. You can now connect that gateway to another activity and continue defining your process.

Result

Since split and join both use the same Parallel gateway (and since a Parallel gateway can be either a split or join), the application distinguishes a split from a join in the following manner: if there is a single FromActivity for a given Parallel Gateway, it is considered a split, if there is more than one FromActivity, it is considered a join.



Parallel Transitions - Limitations

(Short description required.)

Procedure

1. MDM supports the concept of a "group" in parallel transitions. What this means is ,you can group a join transitions together by specifying a group attribute. The "group limit" attribute can be used to indicate the number of transitions to be completed, so that the join can proceed. However, MDM Process Designer does not currently support group and group limit attributes for parallel transition.
2. Every parallel context has to join via a parallel gateway. A process cannot end while the process is split into multiple contexts. Before the process ends, all contexts have to join together.
3. It is advised to have the flow of control between the Parallel gateway split and the Parallel gateway join. Returning the flow of control after a context split, for example, before the split gateway, and re-executing the split possibly multiple times leads to complicated (hard to understand) logic.

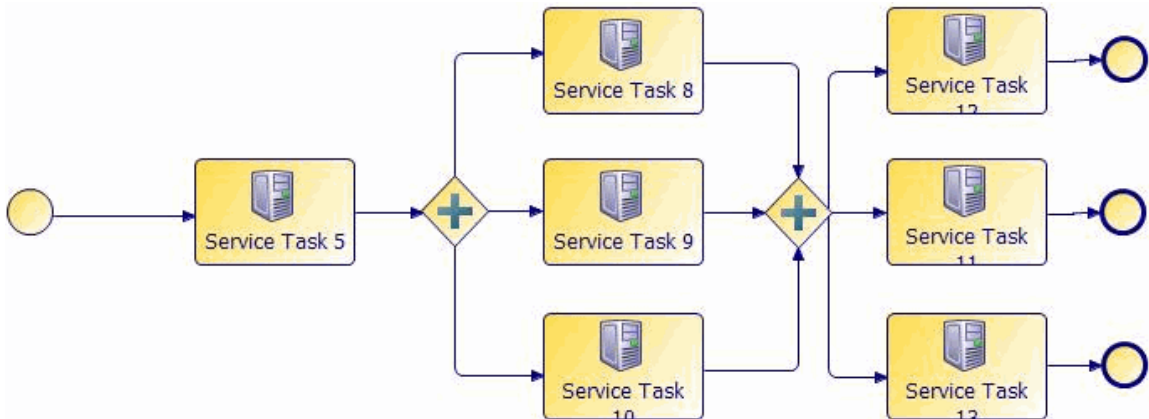
Result

In a split/join for a Parallel transition, the Join must connect to a single MDM activity and cannot connect to multiple activities (another split). Conversely, a Split must connect to more than one activity.

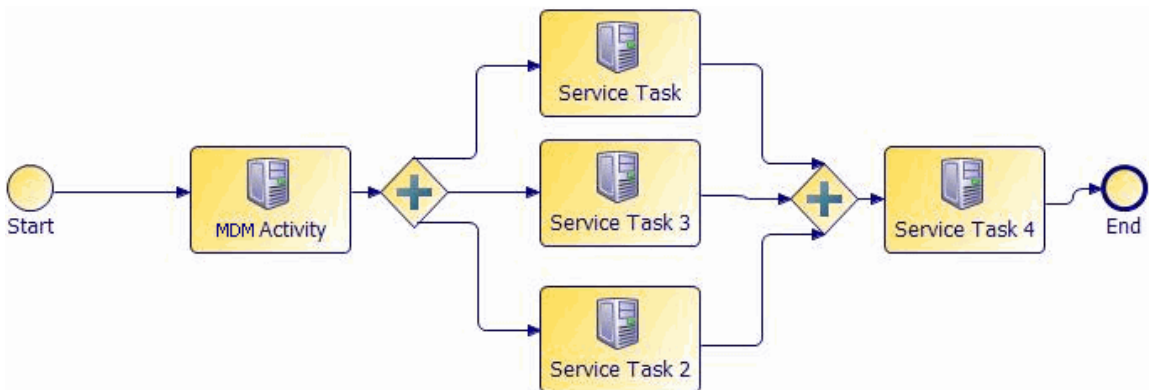
Not Supported: A Split connecting to a single activity.



Not Supported: A Join connecting to a multiple activities.



Supported: A Split to multiple activities; join to a single activity.



Transition Conditions

(Short description required.)

This chapter explains how to create conditions for simple transitions.

Conditional Transitions

(Short description required.)

Conditional transitions determine (or branch) the flow of a process from a particular point depending on the outcome of a specified condition.

To create a conditional transition, you can use:

- **Interpreted Transitions**
Where you enter the condition code manually into a text area.

- **Compiled Transitions**

Where you design the condition code within a java project, check syntax, compile, and just provide a reference to the java class and method inside the condition.



Text area (free text) based conditions get embedded into the workflow file and can be quickly turned around since the MDM engine interprets the code.

Compiled conditions created using a java project perform significantly better, but require an additional jar file to be exported. Compiled conditions are also easier to write since the java code is syntax highlighted and classes and their methods are resolved, avoiding orthographic mistakes.

Interpreted Transitions

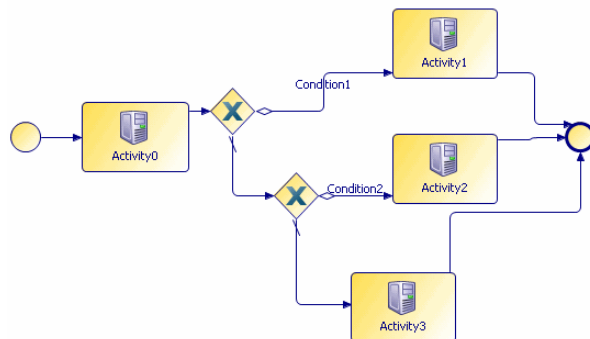
(Short description required.)

Interpreted transitions use a condition entered into a text area and move to a specific activity depending on whether or not a specified condition has been met.

For example, the following figure demonstrates an Activity (Activity0) which has two gateways that map to two conditions; there is also a default condition. The logic is as follows:

- If the first condition (Condition1) is true, after completion of Activity0, it will move to Activity1.
- If the second condition (Condition2) is true, after completion of Activity0, it will move to Activity2.
- Activity3 is the default condition, and it is considered to be true.

Conditional Transition



Creating an Interpreted Transition

(Short description required.)

To create a conditional transition, you need to use Gateway: Conditionally split/join flows based on data to link one activity to two or more activities. (You can also define multiple conditions for the same activity by using multiple Gateways).

To define a simple conditional transition:

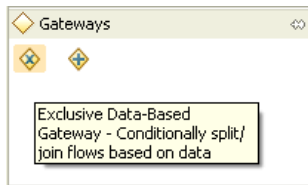
Procedure

1. **Create the activities**

Create two activities Activity1 and Activity2, which Activity0 can optionally move to depending on the condition fulfilled.

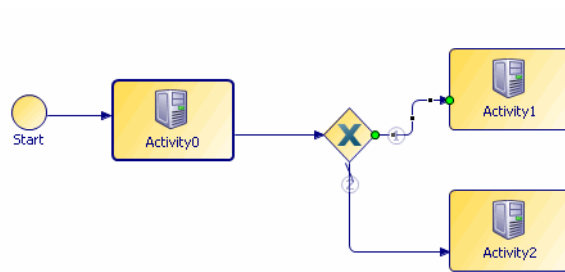
2. **Select the conditional gateway**

Next, select Gateway: Conditionally split/join flows based on data from the Palette and place it between Activity0 and Activity1 and Activity2.

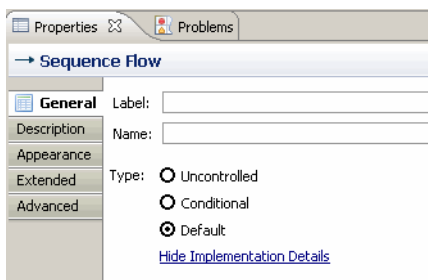


3. Create conditions for the activity

Now create two conditions for Activity0 - to move to Activity1 or Activity2, depending on the condition fulfilled:



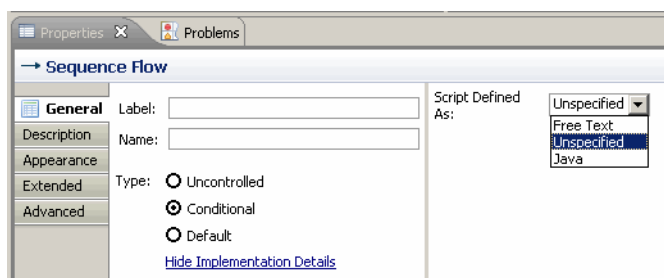
- Use a sequence flow to link Activity0 to the Gateway
- Use a sequence flow to connect the Gateway to Activity 1
- Use a conditional flow to connect the Gateway to Activity 2.
- Mark the flow to Activity2 default by selecting it and then selecting the **Default** option in Sequence Flow in the **Properties** tab.



4. Add condition logic

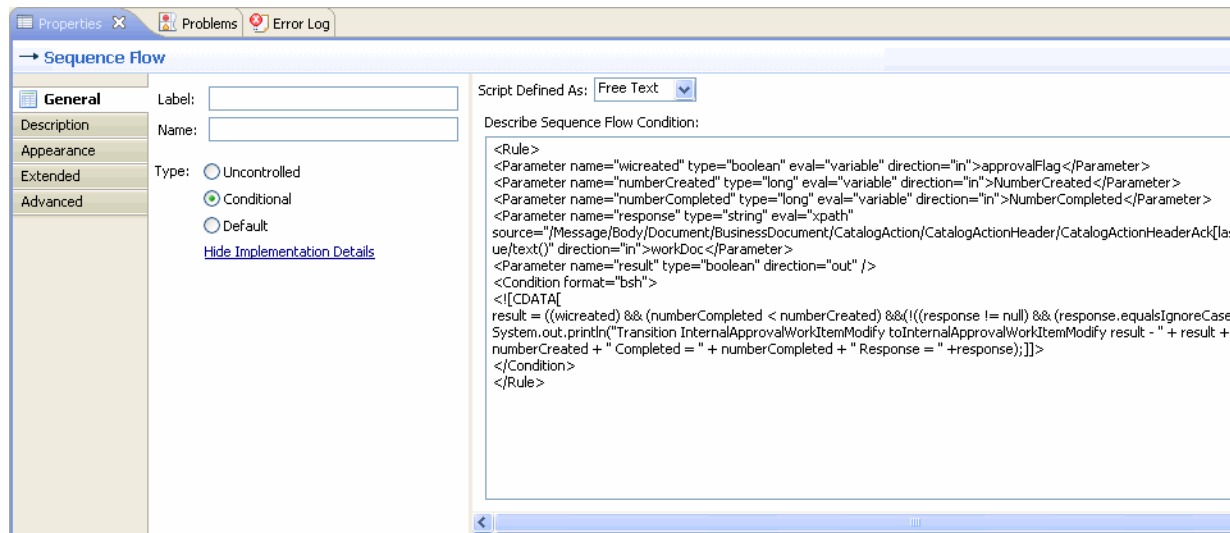
Next, add the condition logic for the non default activity.

- Select the Sequence flow between Gateway and the non default activity (Activity1 in this case).
- In the Sequence flow options in the **Properties** tab, select **Conditional**.
- Select **Free Text** from the **Script Defined As** dropdown.



Selecting the Free Text option makes this an interpreted condition. The Java option enables creation of a compiled java transition - for more information refer [Compiled Transitions](#).

- Manually enter (or cut and paste a previously defined) condition into the **Describe Sequence Flow Condition** box.



Result

Similarly, you can create multiple conditions, and enter the condition code for each.

The inserted condition has to follow a strict format to be valid. Be careful while entering the expression and double check the validity.

The format of the condition is:

```
<Rule>
  <Parameter direction="..." type="..." eval="..." name="...">...</
Parameter>
  <Parameter direction="..." eval="..." type="..." name="..."
source="...">...</Parameter>
  ...
  <Condition format="bsh"><![CDATA[ ...</Condition>
</Rule>
```

Inside the CDATA section, include the java code to be executed. The only requirement is that it needs to have to assign a boolean variable name 'result' a value. This value of this variable will decide whether the transition will be taken or not. The corresponding 'result' parameter must always be present in the Rule tag.

Example of a Conditional Transition

(Short description required.)

Example 1

(Short description required.)

This conditional transition example loops through the `InternalApprovalWorkItemModify` activity, until all participants approve the modification or one of them rejects the modification.

```
<Rule>
  <Parameter name="wicreated" type="boolean"
eval="variable"
direction="in">approvalFlag</Parameter>
  <Parameter name="numberCreated" type="long"
eval="variable"
direction="in">NumberCreated</
Parameter>
  <Parameter name="numberCompleted" type="long" eval="variable"
direction="in">NumberCompleted</Parameter>
  <Parameter name="response" type="string" eval="xpath"
source="/Message/
Body/Document/BusinessDocument/CatalogAction/CatalogA
ctionHeader/
```

```

CatalogActionHeaderAck[last()]/AcknowledgmentCode/Code/Value/text()"
direction="in">workDoc</Parameter>
  <Parameter name="result" type="boolean" direction="out" />
  <Condition format="bsh">
    <![CDATA[
      result = ((wicreated) && (numberCompleted < numberCreated) &&
      ((response != null) && (response.equalsIgnoreCase("Reject"))));
      System.out.println("Transition InternalApprovalWorkItemModify to
      InternalApprovalWorkItemModify result - " + result + " Created
      = " + numberCreated + " Completed = " + numberCompleted + " Response = " +
      response);
    </Condition>
  </Rule>
<Rule>
  <Parameter name="workitemcreation" type="boolean" eval="variable"
  direction="in">approvalFlag</Parameter>
  <Parameter name="result" type="boolean" direction="out" />
  <Condition format="bsh">
    <![CDATA[
      result = workitemcreation;
      System.out.println("In transition from Modify to MoveToNext. The
      result is:"+result+ ", WI created is:"+workitemcreation);
    </Condition>
  </Rule>

```

Example 2

(Short description required.)

This conditional transition example does the following:

- If a ChangeProduct action is taken on a record, it is sent for edit approval (to the users authorized to approve record modifications).
- If a DeleteRecord action is taken on a record, it is sent for delete approval.

```

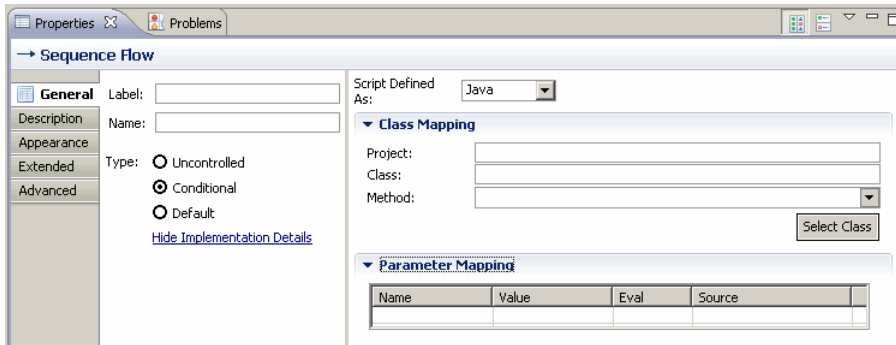
<Rule>
  <Parameter name="action" type="string" eval="xpath" source="/Message/
  Body/Document/BusinessDocument/CatalogAction/CatalogActionDetails/
  CatalogItem[LineNumber=1]/ActionCode/Code/Normal/text()"
  direction="in">workDoc</Parameter>
  <Parameter name="result" type="boolean" direction="out" />
  <Condition format="bsh">
    <![CDATA[
      result = (action != null) &&
      action.equalsIgnoreCase("ChangeProduct") ;
      System.out.println("In transition from MoveToFirst to
      InternalApprovalWorkItemModify:"+result);
    </Condition>
  </Rule>
<Rule>
  <Parameter name="action" type="string" eval="xpath" source="/Message/
  Body/Document/BusinessDocument/CatalogAction/CatalogActionDetails/
  CatalogItem[LineNumber=1]/ActionCode/Code/Normal/text()"
  direction="in">workDoc</Parameter>
  <Parameter name="result" type="boolean" direction="out" />
  <Condition format="bsh">
    <![CDATA[
      result = (action != null) &&
      action.equalsIgnoreCase("DeleteProduct") ;
      System.out.println("In transition from MoveToFirst to
      InternalApprovalWorkItemDelete:"+result);
    </Condition>
  </Rule>

```

Compiled Transitions

(Short description required.)

When you define a conditional transition using Java, the code is not defined in the process flow itself, but the java class loading mechanism is used. The condition only contains a class and method reference to the code to be invoked and the parameters needed for invocation.

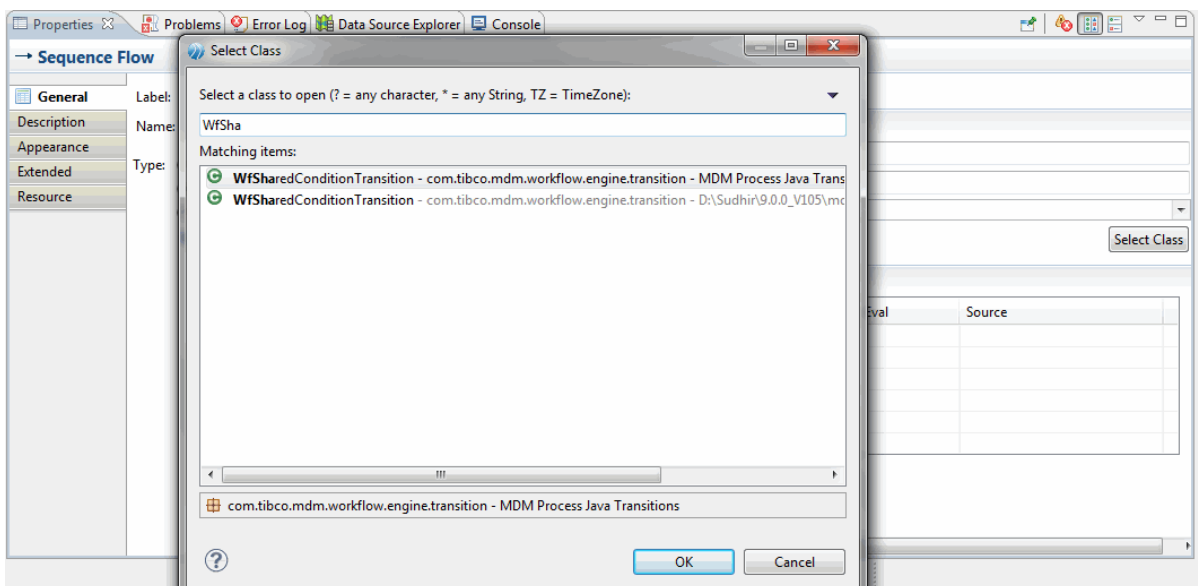


The Java Grammar option for a conditional transition allows you to input a java class containing the condition code, which should be predefined in an eclipse java project. You can create a java project to hold the transition using the New Java Project wizard (**File -> New -> Project -> Java Project**) in Business Studio.

Once your condition (class) is ready, you can reference it from within the **Properties** tab of the Conditional transition by using the Class Picker (the **Select Class** button) to select the class from an existing java project.

Before using the class picker, ensure that you have defined the java code in an eclipse java project. You can then select this class through the Select Class dialog;

The selected class will be embedded into the XPDL document only by name. The code has to be explicitly exported and merged in the .EAR file.

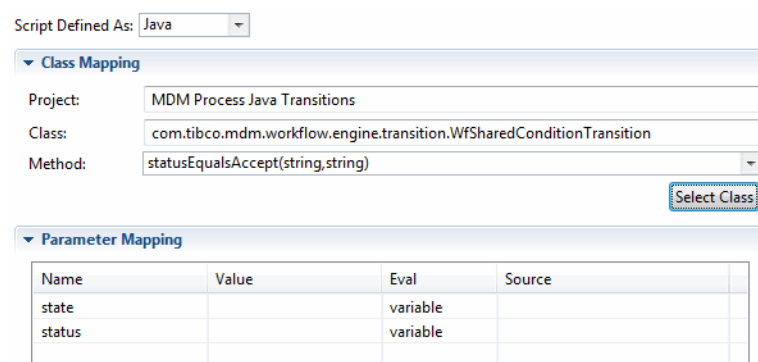


Once you select the class, the **Project**, **Class**, and **Method** textboxes get populated with read only values. Condition methods will be populated (read-only) as a string in the **Method** drop down box.

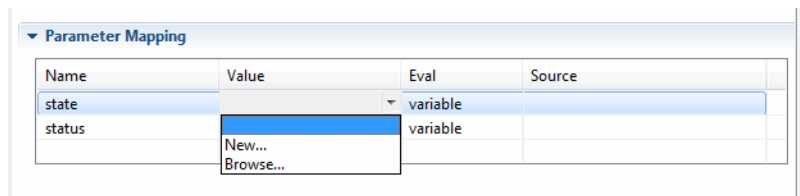
Parameters

(Short description required.)

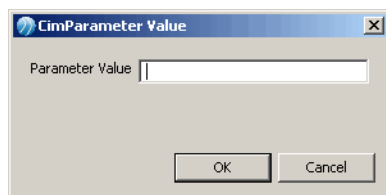
Parameters are displayed in the **Parameter Mapping** section below the **Class Mapping** section.



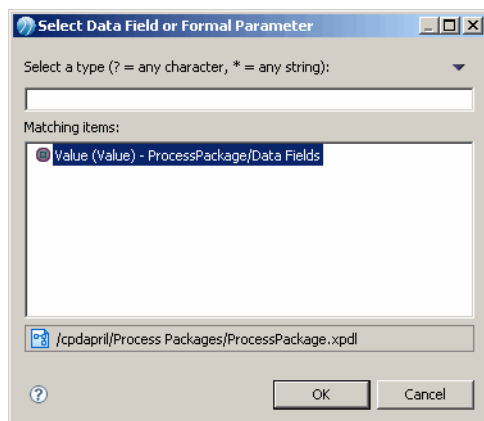
You can modify the parameter value by clicking in the **Value** column against the parameter. You get two options here: **New** and **Browse**.



Click **New** if you want to enter a constant value.



Click **Browse** if you have defined a global or local variable (data field) that you want to map.



Creating a Compiled Transition

(Short description required.)

The following tasks explain the entire process of creating a compiled transition using Java step by step.

Procedure

1. [Creating a Java Transition Project](#)
2. [Using the Condition in the process](#)
3. [Compiling and packaging the Java Project](#)
4. [Merging Jar With ECM.ear](#)

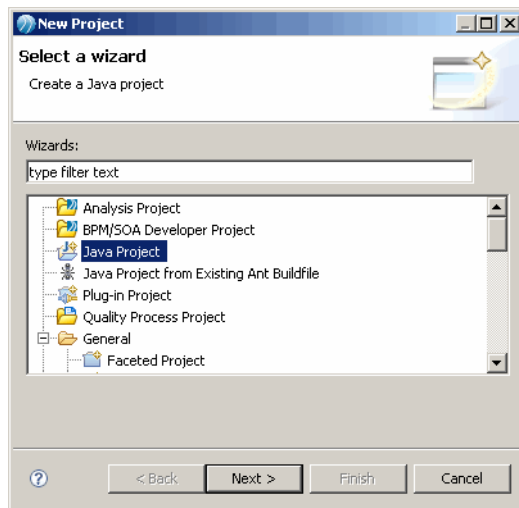
Creating a Java Transition Project

(Short description required.)

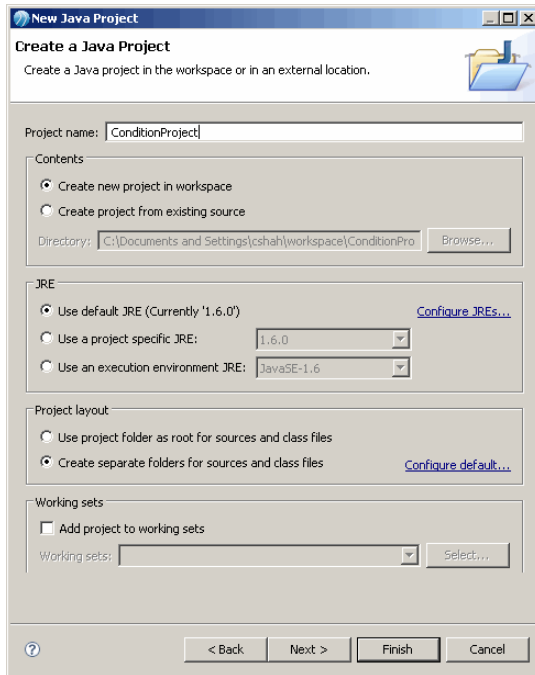
Create a new Java Project for the transition condition.

Procedure

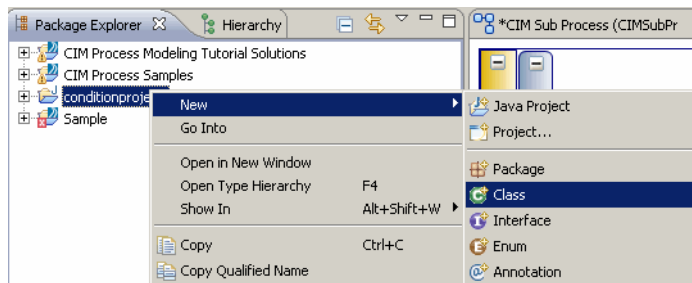
1. Use the Project wizard **File -> New -> Project -> Java Project** . Click **Next**.



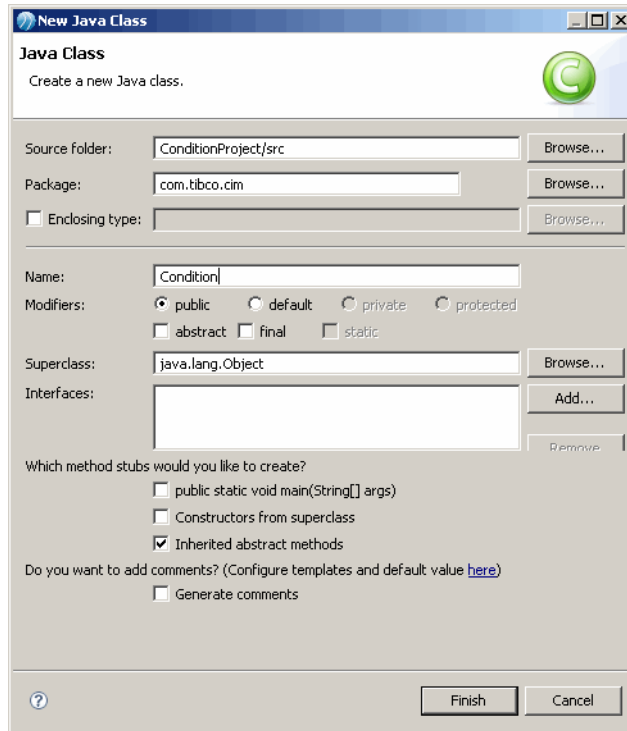
2. Enter a project name and click **Finish**.



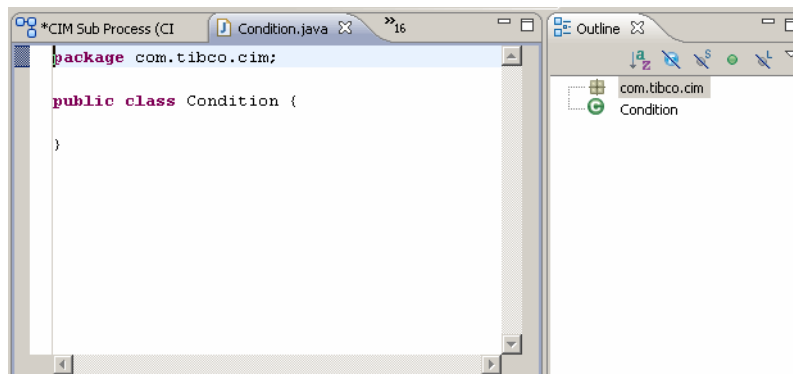
3. Add a new java class to the project by right clicking the project and selecting **New -> Class**.



4. Provide a name for the class (for example, **Condition**) and the package (for example, **com.tibco.cim**).



5. Click **Finish**. The eclipse java class editor will be displayed with an empty stub of the class.



6. Enter your transition logic here or as an example, copy the following class text and replace the empty stub:

```
package com.tibco.cim;
public class Condition {
    static public boolean evaluateCondition(String a, String b){
        return true;
    }
    static public boolean evalRule(String subtype,String splitImportBatch){
        boolean result = ((splitImportBatch != null) &&
(splitImportBatch.equalsIgnoreCase("Split/Approval Required")));
        return result;
    }
    static public boolean evaluateResult(Long ParticipantID){
        boolean result = (ParticipantID != null);
        System.out.println("Transition from ConflictResolutionGoToFirstC To
ConflictResolutionWIConfirmed : id= " + ParticipantID + "----result - "+result);
        return result;
    }
}
```

Using the Condition in the process

(Short description required.)

Procedure

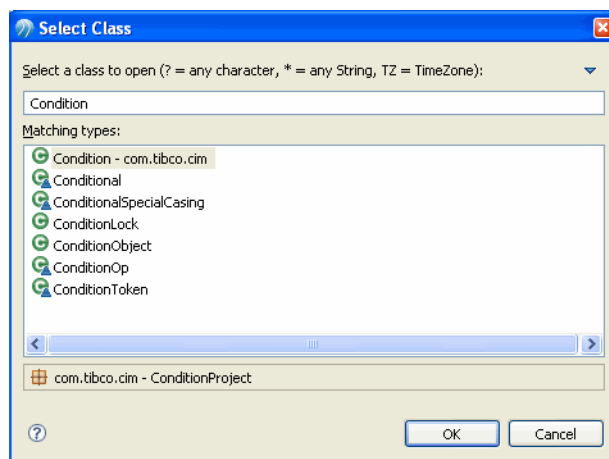
1. Switch to the Modeling view and select the transition to which you want to apply the java condition.



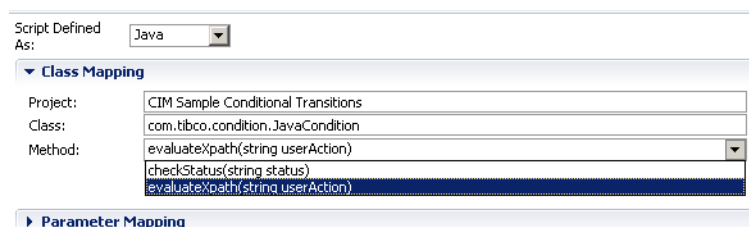
2. Select **Java** in the **Script Defined As** dropdown.



3. Click the **Select Class** button and enter the name (or the initial characters) of the previously defined condition.



4. Select the java type **Condition** and click **OK**.
5. Any of the three static methods inside the Java class **Condition** can be selected.



6. Define parameters in the **Parameter Mapping** section.
7. Save the Process Project; the project can now be exported.

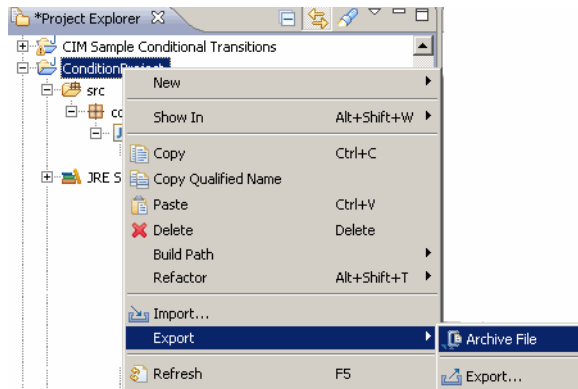
Compiling and packaging the Java Project

(Short description required.)

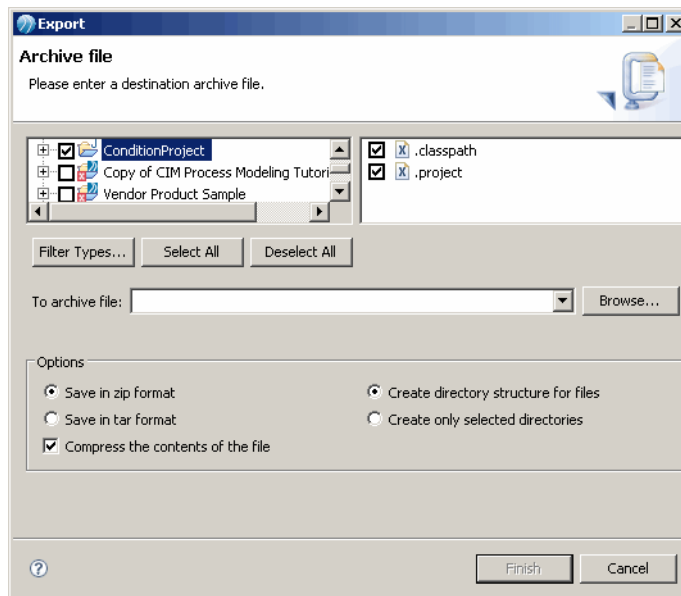
The compilation of the java file should be automatic.

Procedure

1. To export, right click the Project and select **Export -> ArchiveFile**



2. Select the java project (here, ConditionProject), select the type of archive to create, and provide a name for the archive file. Click **Finish**.



3. The file must now be merged with the ECM.ear file of the core MDM server.

Merging Jar With ECM.ear



The following steps explain how to merge your exported condition jar into ECM.ear.

(Short description required.)

Merging Jar with ECM.ear using Windows

(Short description required.)

On Windows, to merge an external jar file with `ECMClasses.jar`, you can use a packaging software such as winzip or winrar.

The location for package from `ECMClasses.jar` to merge the custom transition is `classes\com\tibco\com\tibco\mdm\workflow\engine\transition*.class`.

Merging Jar With ECM.ear using UNIX

(Short description required.)

Procedure

1. Extract the `Condition.jar` file (maintaining the directory structure when extracting) and place the extracted directory into the `classes/commonClasses` directory. For example: `classes/commonClasses/com/tibco/com/tibco/mdm/workflow/engine/transition/*.class`.
2. Change the directory to the parent of `classes/commonClasses` and run the following command (which will create a new custom jar file, ready to merge with the `ECM.ear` file).
`$jar -cvf <new jar file name> classes`

Follow these steps to merge the new `condition.jar` with `ECM.ear`:

3. Create the directory `$MQ_HOME/customEAR`.
4. Copy the `condition.jar` into the `customEAR` directory.
5. Go to `$MQ_HOME/build/custom`.
6. Execute `./customUtil.sh -updateEarFile`.
7. Enter "y" to merge. Continue and complete the script.
8. The updated `ECM.ear` will be available at `$MQ_HOME/customEAR`.
9. Deploy the new `ECM.ear`.
10. Restart the MDM Server. You can now use the new conditional transition in your workflow.

Result



Alternatively, on any MDM installation, you can copy the exported jar file to the application server library directory.

Validation

(Short description required.)

This chapter explains process Validation.

Validating

(Short description required.)

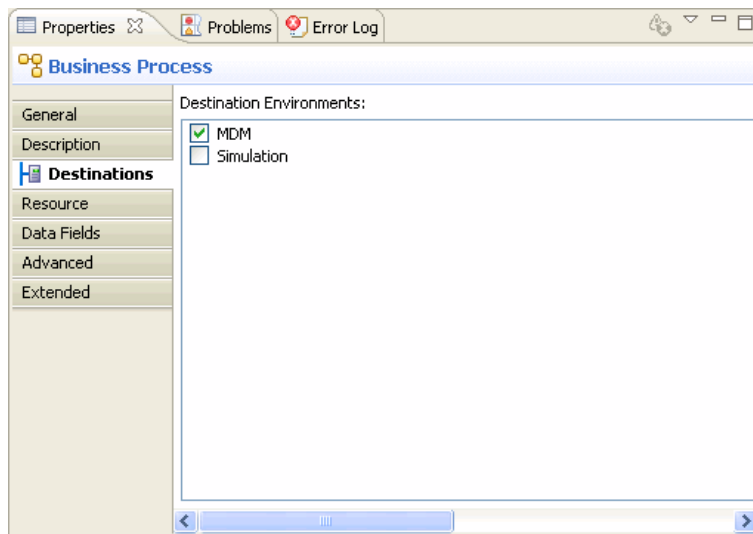
The MDM validation plug-in runs in the background as you modify or edit the process; it checks that changes are made in accordance with the validation rules.

Validation is triggered when the destination of the process is set to MDM. To do this:

Procedure

1. Select the Process in the Project Explorer.
2. In the **Properties** Window, click the **Destinations** tab.
3. Set the destination environment to **MDM** by selecting the checkbox.

Result



All errors will show up in the **Problems** tab of the **Properties** Window.



Validation will automatically occur if you import an existing MDM process into the MDM Process Designer for editing (since the process destination is set to MDM by default) or if you define a process from scratch using the MDM template.

However, if you choose to create a process from scratch without using a MDM template, you will need to set the process destination to MDM for validation to take place.

Problems Tab

(Short description required.)

The **Problems** tab displays errors related to specific elements in your process. Information is categorized into Errors and Warnings.

Problems Tab

Description	Resource	Path	Locat...	Type
2 errors, 15 warnings, 0 others				
Errors (2 items)				
BPMN 1.2 : Catch link event must have exactly one outgoing sequence flow. (MDM5Sub	Sample1.xpdl	Sample/Process P...	_Md1_...	BPMN Problems
MDM 8.x : Activities/Gateways must be reachable from the start activity. (MDM5SubPro	Sample1.xpdl	Sample/Process P...	_LQK17...	Validation Problems
Warnings (15 items)				
Business Object Model Generic : Cyclic Dependency has been detected for this resour	CatAction.bom	Sample/Business ...	_4RKD...	Validation Problems

- **Description:** This provides error descriptions along with the processname and elementname the error is applicable to.
- **Resource:** This displays the name of the xpdl file which contains the specific error.
- **Path:** This tells you the path of the file; this is typically in the <Project Name>/Process Packages format.
- **Location:** This signifies the internal process or event ID used in XPDML.

Validation Checks

(Short description required.)

The following validation checks are performed to ensure conformity to MDM standards:

Miscellaneous Validations

(Short description required.)

- **All Activity Types to Service Task**
All Activity types should be set to Service Task. Usage of any other activity type will result in an error.
- **Single starting point for process**
The process should have only a single Start Event.
- **Intermediate Event Restrictions**
Intermediate events are restricted to exception, timeout, cancel, and Link Event. Use of any other type will result in an error.

Activity Validations

(Short description required.)

- **Unique Activity Names**
Each activity should have a unique name, which is mapped to the MDM activity name. Activity names are case sensitive. Spaces and special characters are not allowed in activity name.
- **No Disconnected Activities**
Activities which cannot be executed from a start event are superfluous and will result in a validation error.

Transition Validations

(Short description required.)

- **No Multiple uncontrolled transitions**
Multiple uncontrolled transitions should not be present since they are not supported by the MDM workflow engine. If there is more than one outgoing transition from an activity, a gateway has to be used.
- **No inbound transitions for Boundary events**
Boundary events (cancel, error, timer) cannot have inbound transitions.

- Single outbound transition for Boundary events
Boundary events must have a single outbound transition.

Gateway Validations

(Short description required.)

- Only Conditional (XOR) or Parallel (AND) gateways
Gateway Type should be XOR Data Gateway (Exclusive Decision/Merge Data or Event based) or Parallel Gateway (Parallel Fork/Join). No other Gateway Types are supported.
- Parallel gateway placement in a process
A parallel gateway cannot be the last activity in a process.
- No multiple inputs for gateways with conditional outputs
Gateway with conditional outputs cannot have multiple inputs.
- No multiple conditional output flows for gateways
Gateways cannot have multiple conditional output flows.
- Gateway origins for conditional/default sequence flows
All conditional and default sequence flows must come from a gateway.

Parameter Validations

(Short description required.)

- Source values for rule/xpath/catalog/lookup evals
If the eval attribute value is rule or xpath or catalog or lookup, the source attribute has to be in the data field.

Parameter Mapping Validations

(Short description required.)

There should be no broken parameter mappings. Either remove the parameter mapping or add the parameter in the wsdl.

Data Field Restrictions

(Short description required.)

- Global data field restrictions (WARNING)
Data Fields must be declared in the real process or at the package level.



Business Studio elements that are not supported by MDM Process Designer, for example associations, annotations and so on, will not be validated. These associations can be freely used by the Designer.

Import Export and Deploying Processes

(Short description required.)

This chapter explains how to import processes into the MDM Process Designer, how to export processes created in MDM Process Designer to MDM and how to deploy processes.

Import Processes into MDM Process Designer

(Short description required.)

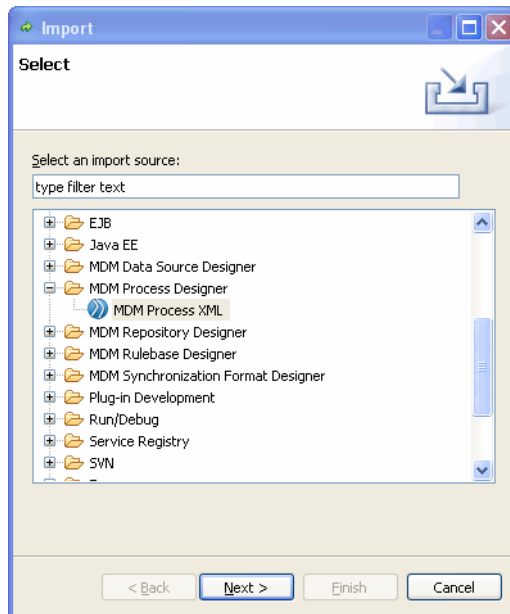
In addition to enabling you to graphically define processes, the MDM Process Designer also supports import of existing MDM processes for modification and subsequent re-export. This is particularly useful for customers with existing processes which need to be modified or tweaked.

Importing a Process

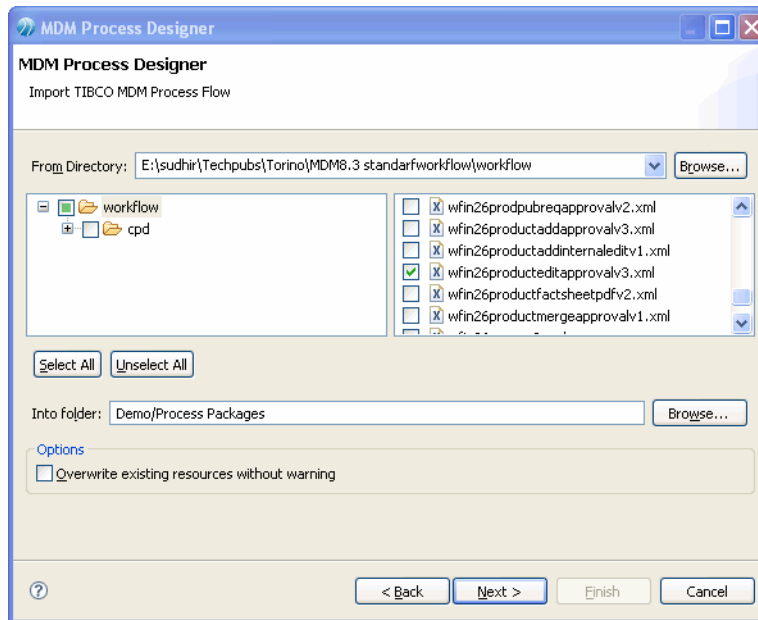
(Short description required.)

Procedure

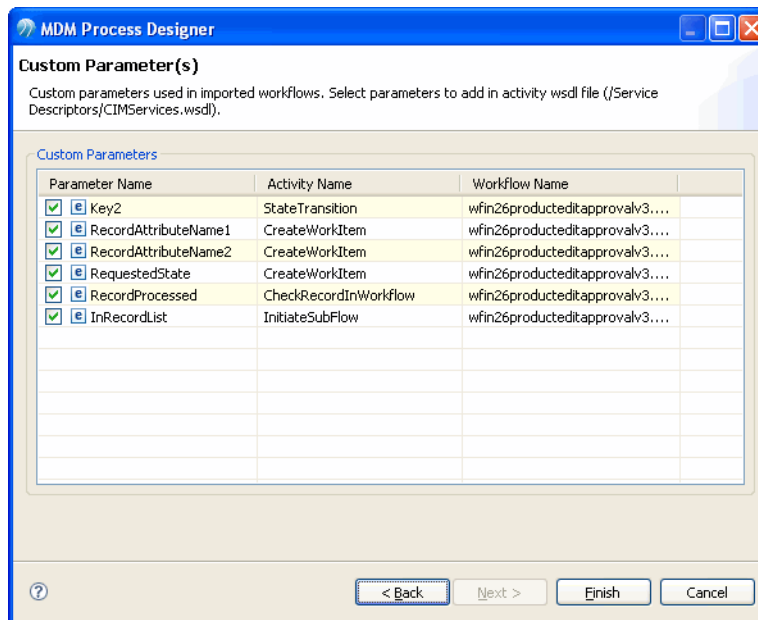
1. Create an empty project with a Process Package.
 - a) Go to **File -> New -> Project**.
 - b) Select **MDM Developer Project**.
 - c) Provide a name for the Project.
 - d) Select **Business Processes, Business Objects, and Services** in the Asset type Selection dialog.
 - e) Follow the rest of the wizard to complete the Process creation without using the MDM Template.
2. Use the **Import Wizard** to import an existing workflow:
 - a) Right-click the **Process Packages** folder in the Project and select **Import**.
 - b) Choose **MDM Process XML** under MDM Process Designer. Click **Next**.



- c) Browse and select the process xml file from the File System and Click **Next**.



d) The custom parameters used in the imported workflow is displayed.



e) Select the custom parameters which you want to add in the activity and click **Finish**.

Clean Up an Imported Process

(Short description required.)

When you import a process into MDM Process Designer, there may be some overlap in the flow elements, for instance, you may have one task over the other or overlapping transitions and annotations. You may need to manually move some of the elements to have the process look as intended. Here are some tips on how to clean up the process.

Transition Names

- Shorten names for transitions and activities.
- Move the position of transition names.
- Make transition names multi line where required.

Rearrange Transitions

- Rearrange transitions to avoid two on the same path.

Connect Isolated Activities

Connect any unconnected (isolated) activities or delete them altogether. In case of MDM workflows imported into Process Designer, if there are any activities unconnected by transitions (isolated activities), you need to either delete such activities or manually connect them via transitions. If you ignore an unconnected activity, you will get an error when you attempt to export the process.

Ensure all activities are connected by transitions and that any isolated activities are deleted before you attempt to export the process.



Unconnected activities will be overlapped after import. You need to clean up the layout and spread out the activities.

Connect/Remove broken parameter mappings

In case of MDM workflows imported or migrated into Process Designer, if there are any broken parameter mappings, you need to either remove such mappings or add the parameter(s) in the wsdl using the Quick Fix.

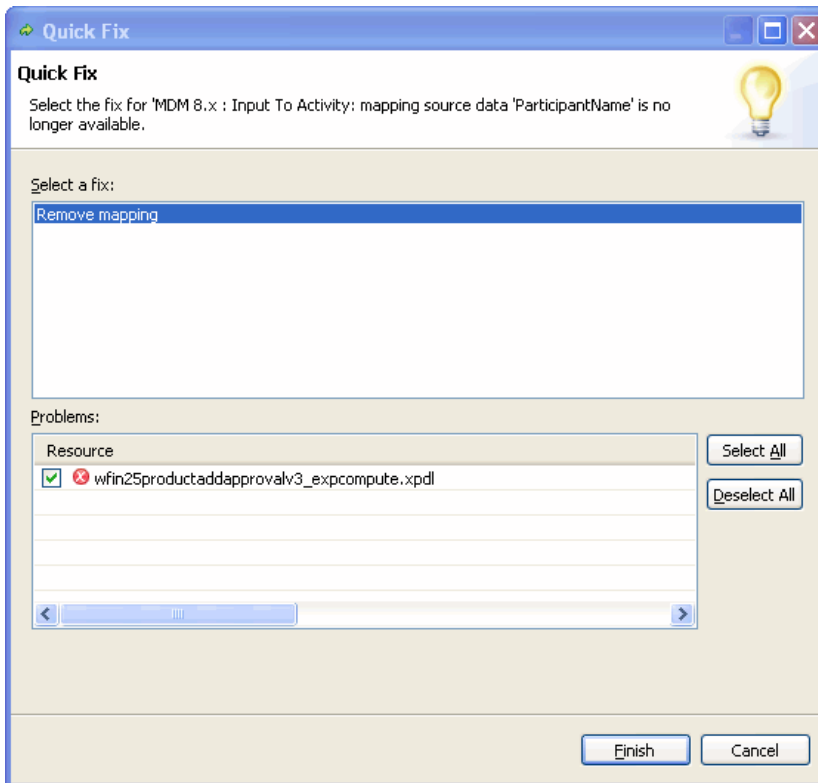
Click the Quick Fix option (available on right clicking a mapping validation error in the Problems tab.)

The Quick Fix dialog will display the problem and applicable resources. Click **Finish**.

The screenshot shows the 'Problems' tab in the TIBCO MDM Studio interface. It displays a table of 5 errors and 23 warnings. A context menu is open over one of the error rows, showing options like 'Go to', 'Copy', 'Delete', 'Select All', 'Show In', 'Quick Fix', and 'Properties'. The 'Quick Fix' option is highlighted.

Description	Resource	Path	Locat...	Type
MDM 8.x : Activities/Gateways must be reachable from the start activity. (Mainwfin23pr	wfin25prod...	Sample/Process P...	Main_...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'Particip	in25prod...	Sample/Process P...	//@pa...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'workD	in25prod...	Sample/Process P...	//@pa...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'workD	in25prod...	Sample/Process P...	//@pa...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'workD	in25prod...	Sample/Process P...	//@pa...	Validation Problems

ping source data 'ParticipantName' is no longer avail...dTimeOutEmailForApprovalWorkItem:wso:Send/part:parameters/group:sequence[1]/Approver)



Link Events

- Use Link Events for far stretching transitions.

Activity Placement and Names

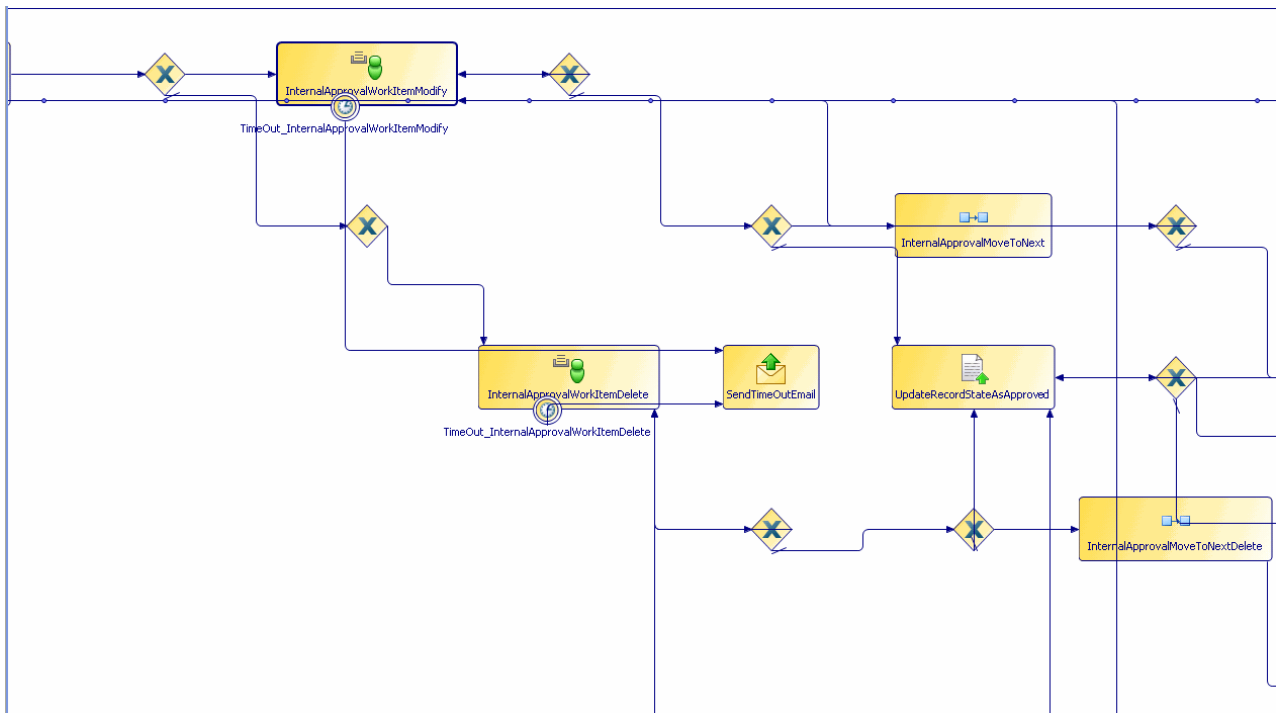
- Rearrange activities if required or overlapping.
- Expand activity rectangles to fit the name.

Any transition from the outer (main) process into the inner process will be eliminated since transitions between processes are not supported. You need to establish the same logic again.

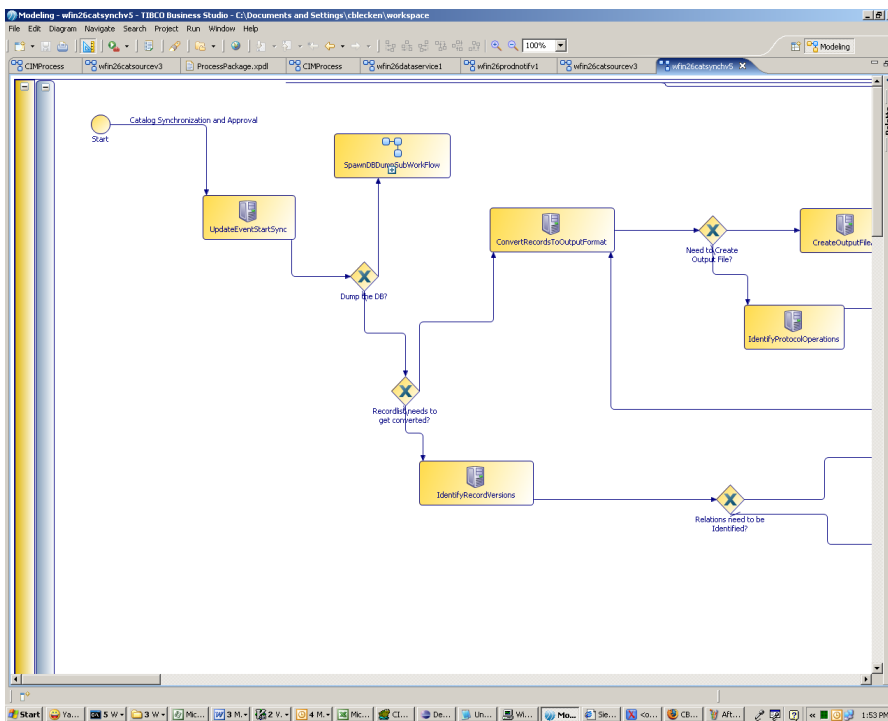


For instance, if your original workflow has duplicate exception handling activities, one of these will be ignored, and as a result the inner process will not have an exception handler. If you require this, you need to create it manually.

Sample of an imported process before cleanup



Sample of an imported process after cleanup



Limitations of the Process Import

(Short description required.)

- Existing MDM workflows do not have associated layout information that allows for placing activities and transitions per your preference. Import functionality, however, automatically generates layout information, and align activities and gateways rigidly on a grid. This generation may sometimes lead to an unintended layout which you may have to manually change to rearrange the activities and gateways to a more appropriate view.

- Import functionality also has practical limits on the size of the imported workflow. If it is too large, the workflow layout will be confusing and will require significant user intervention to be reasonable to use. While the import functionality works with an arbitrary number of activities, the practical limit is around 50 activities. It is recommended to break the workflow down into smaller subflows in case of large workflows.
- The automatic layout though useful, still required some manual rearrangement in the main process view.



The lack of layout information is the main reason why workflow export, its subsequent use and re-import is not useful, since the workflow will have changed the layout each time it is imported. Therefore, the import functionality should be viewed as a one time conversion from an existing workflow into a graphically editable process model. The user is encouraged to view the copy of the workflow definition inside the MDM Process Designer as the primary one (perhaps by sharing it through a source code repository with others) and not the version deployed in the MDM Engine.

Migrating Processes

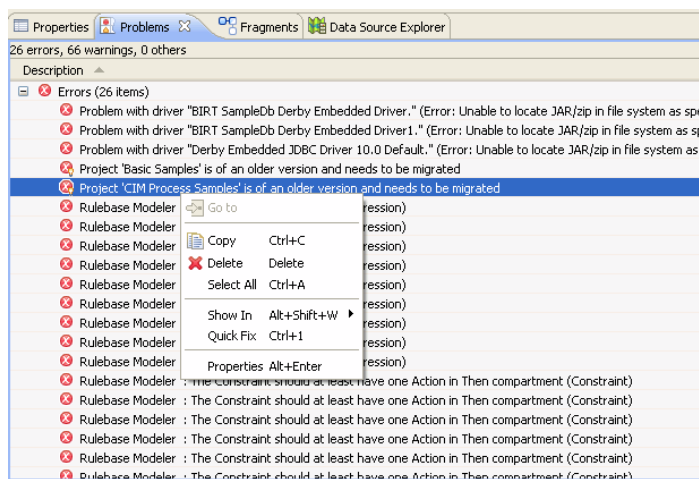
(Short description required.)

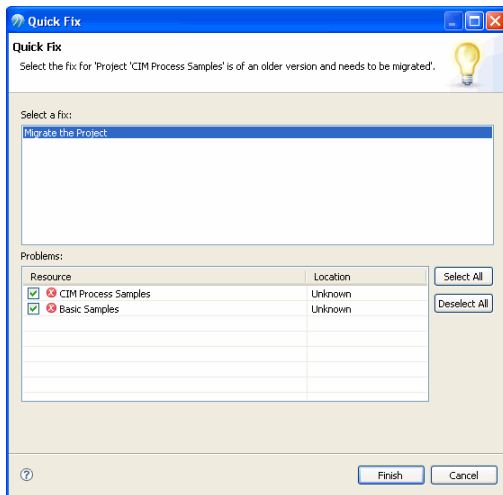
To migrate any older version projects into the current Process Designer, follow these steps:

Procedure

1. Import the older project into the Process Designer workspace. The processes won't be open and will run into validation problems.
2. Right-click the version problem in the Problems Tab and select **Quick Fix**; this action will migrate the selected processes to the latest version of the Process Designer.

Result





Directly Deploying Workflows

(Short description required.)

The Process Designer supports direct deployment of workflows. You do not need to manually export your process and import it into MDM. This network deployment is a web service, through which, you can list processes, deploy process definitions or undeploy process definitions.

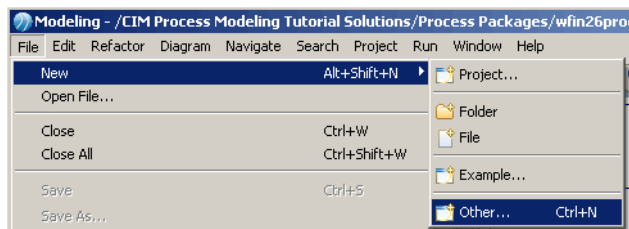
The XPDL file generated by the Process Designer is validated before it is translated into the native MDM workflow format.

Creating a MDM Deployment Server

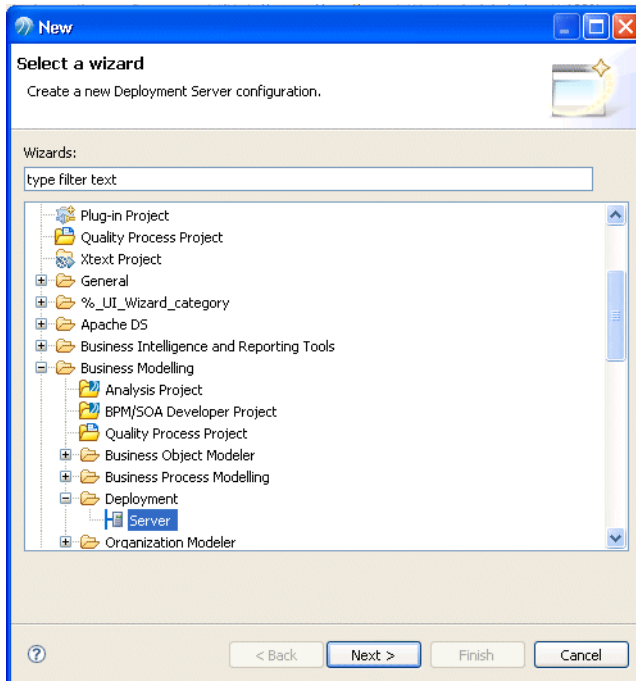
(Short description required.)

Procedure

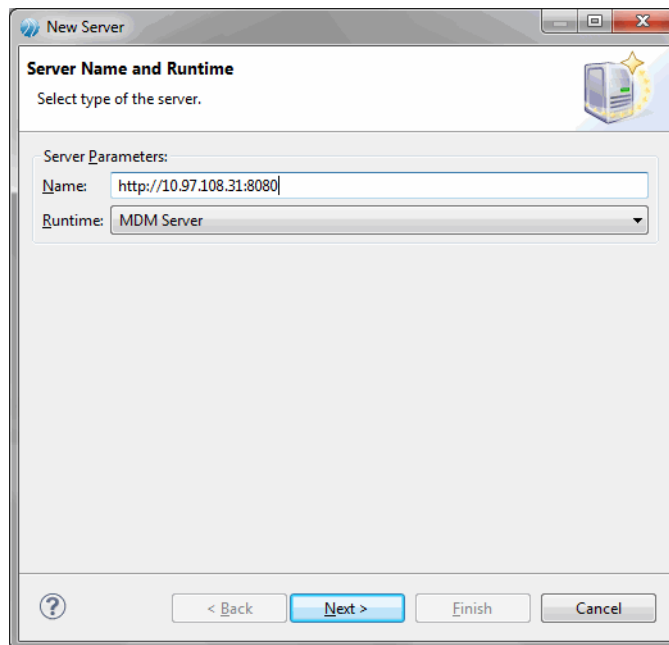
1. Select **File- > New- > Other.**



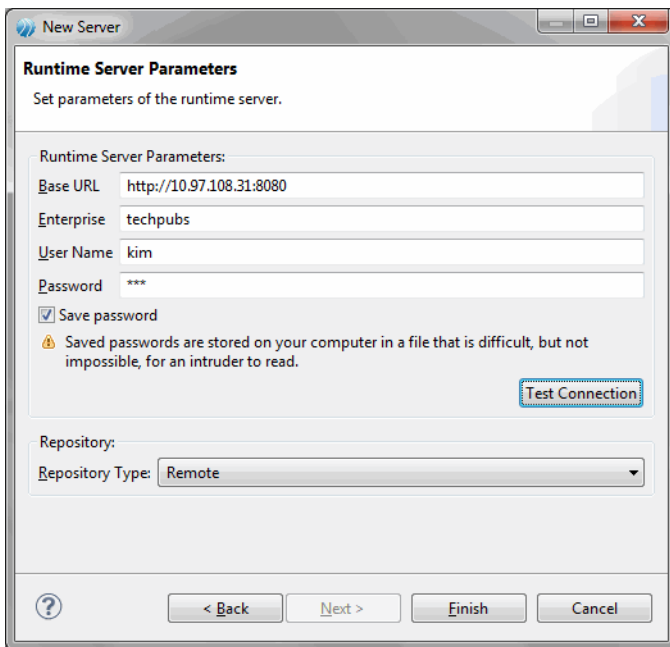
2. Select **Business Modeling- > Deployment- > Server .**



3. Provide the Server Name; select **MDM Server** as Runtime.



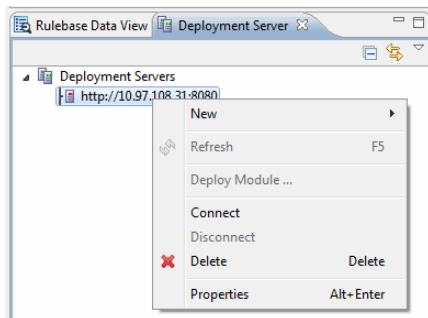
4. Enter the **Base URL**, **Enterprise**, **User Name**, **Password** and select **Remote** as the Repository Type. Click **Finish**.



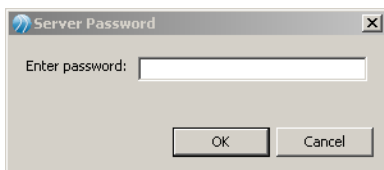
If you select the **Save password** option, you will not be prompted for a password in the following step.

5. Right click on the newly created **MDM Server** and select the **Connect** submenu.

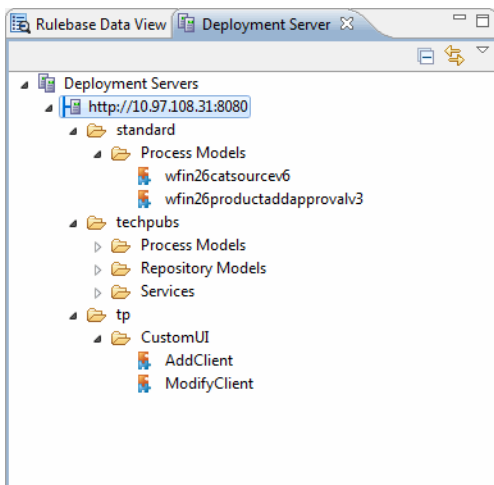
Result



If you did not choose to save the password, you will be prompted to enter the password.



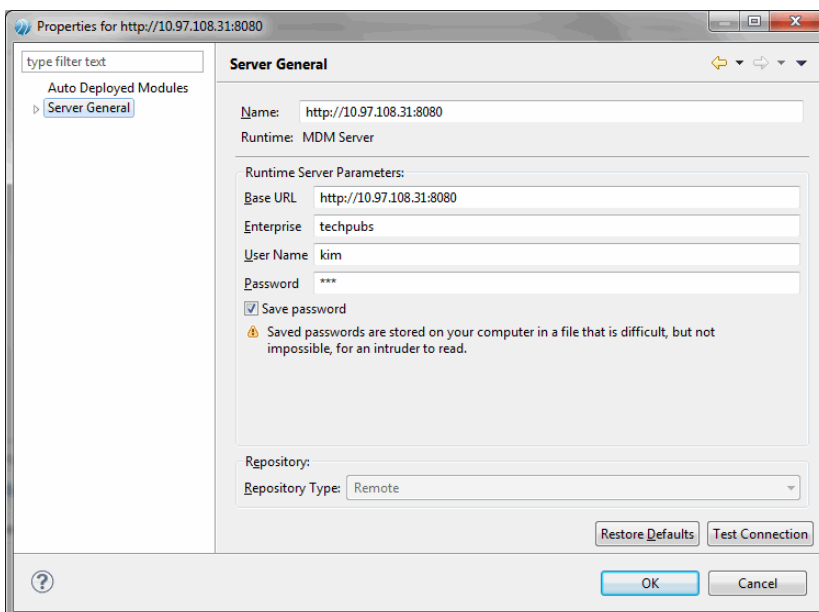
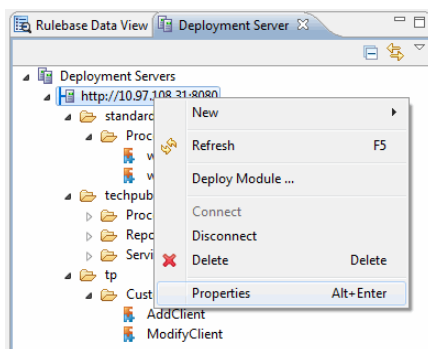
This will establish a connection between the MDM Server and the Process Designer client and will show all deployed workflows on the MDM server.



Editing Server Parameters

(Short description required.)

You can edit server parameters by right clicking the Server and going to **Properties**.



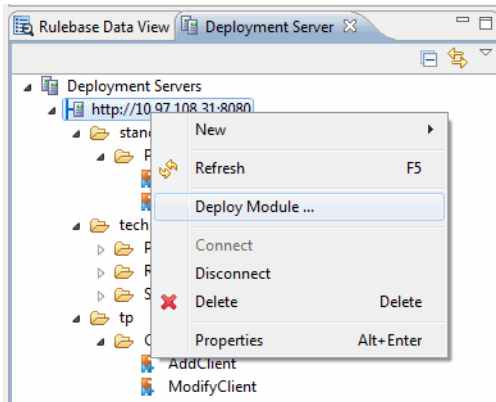
Deploying Workflows

(Short description required.)

You can deploy both new workflows and modified (existing) workflows.

Procedure

1. Right click <MDM Server and select **Deploy Module**.



In case you do not see the Deployment Server Pane, go to **Window > Show view > Other** and select **Deployment Server** under **Studio**. Click **OK**.

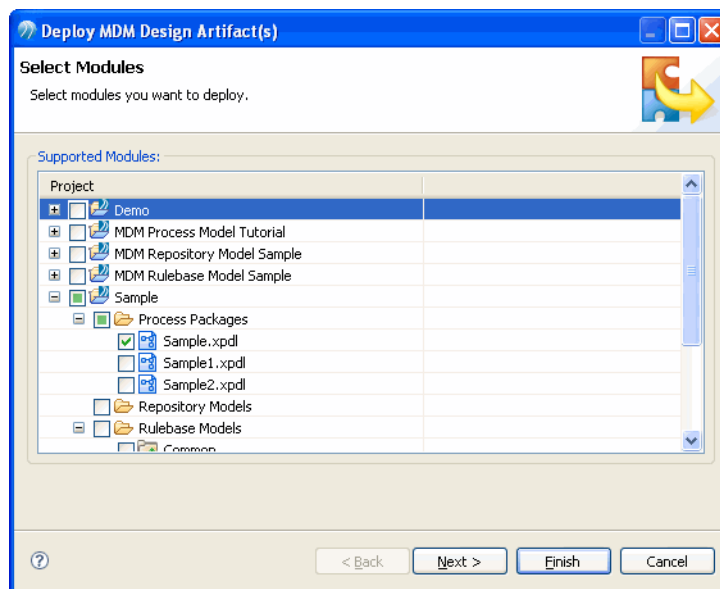


Click



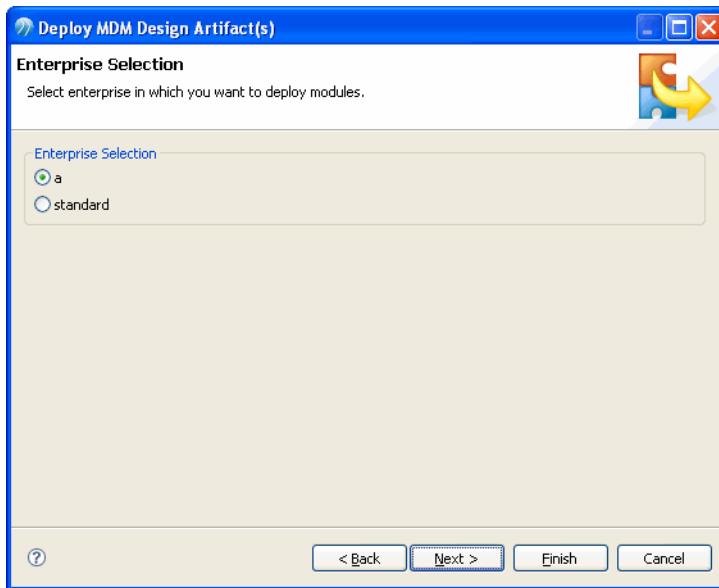
and select **Solution Design**.

2. Select the workflow to deploy and click **Next**.



3. Select the enterprise to deploy the workflow to (either the current enterprise or standard). Click **Finish**.

Result



This will either deploy the selected workflow and will show a successful deployment message or an error message if the workflow could not be deployed.

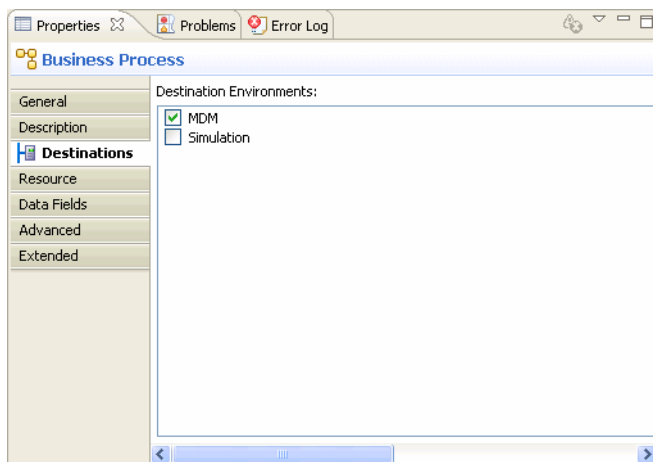
Deploying Workflows through Export

(Short description required.)

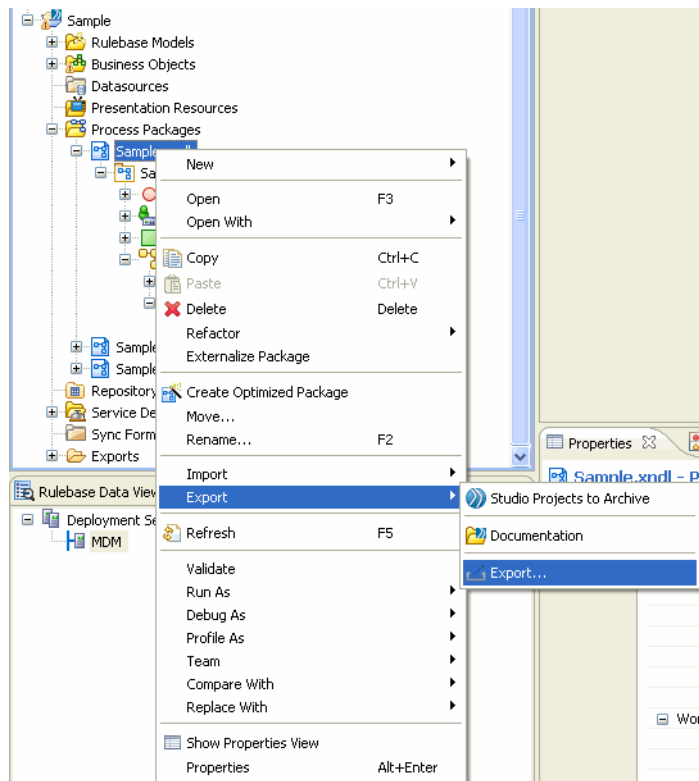
Once a MDM Workflow has been defined and edited, follow these steps to export it:

Procedure

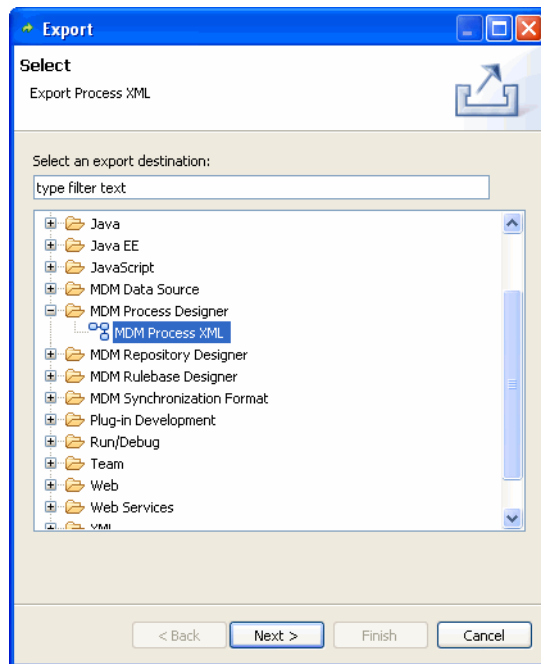
1. Ensure that the **Destination** for the Process is set to **MDM**. To do this:
 - In the **Properties** Window, click the **Destinations** Tab.
 - Set the Destination Environment to **MDM** by selecting the checkbox.



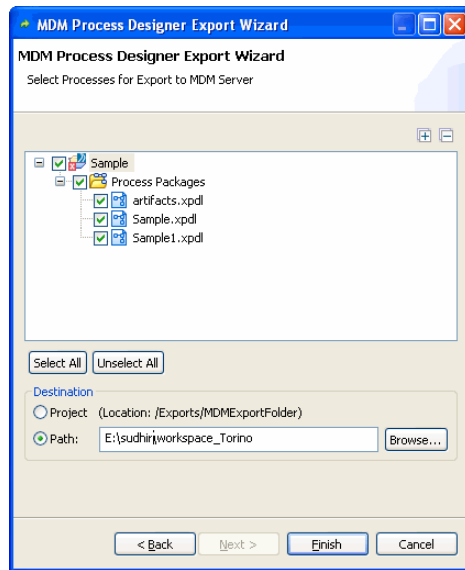
2. Right-click the Sample Package (xpdl file) in the Project Explorer and select the **Export** option.



3. Select **MDM Process XML** under **MDM Process Designer** and click **Next**.



4. Select the location to export the process (xpd) to. You can either export it to the default projects export folder, or to a file. Click **Finish**.



- The export is now complete. If you go to the location where you chose to export the file, you will see an xml output file. This is the xpdl process file which has been converted to the TIBCO MDM Engine xml format.

- | Name | Size | Type | Date Modified |
|----------------|------|--------------|-------------------|
| ProcessPackage | 4 KB | XML Document | 9/17/2007 1:51 AM |

Deployment to the Workflow Engine

(Short description required.)

Once you export your process to an XML file, you need to copy that XML file to the MDM defined location for workflow files.

Workflows in MDM are saved in either of the following locations:

- A common workflow location (\$MQ_COMMOM_DIR/standard/workflow) or
- An enterprise specific workflow location (\$MQ_COMMOM_DIR/<enterprise internal name>/workflow)

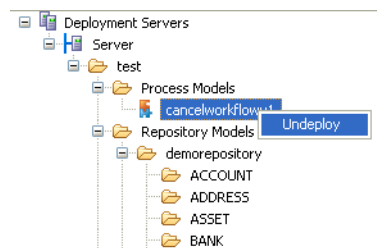


The Enterprise specific workflow location takes precedence over the common workflow location. If the required workflow is not found in the enterprise specific workflow location, the application will look in the common workflow location.

Undeploy workflows

(Short description required.)

To undeploy a workflow from the server, right click the deployed module and select **Undeploy**.



This undeploys the selected workflow, and a backup of it is internally renamed and stored.

Sharing Processes

(Short description required.)

This appendix describes how you can create and share processes with other users.

Overview

(Short description required.)

MDM Processes are often designed in a multi user environment, where multiple people need access to the same file and also a versioning of the process designs becomes important.

It is recommended that you use a standard version control system, such as CVS or Subversion/SVN. Both version control systems facilitate shared development of process designs and also allow versioning the projects.

SVN version control systems can be accessed from <http://subversion.tigris.org/>.

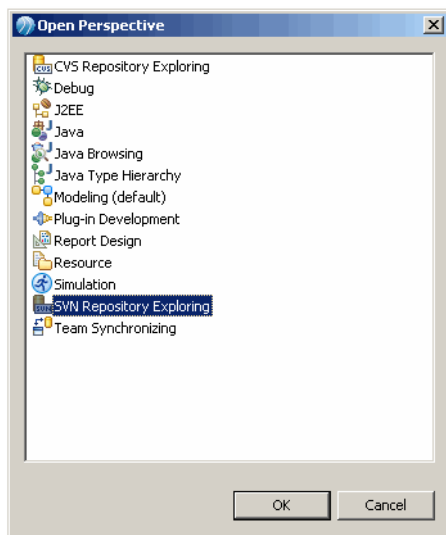
This Appendix takes you through sharing of MDM processes, assuming that a subversion version control system has been set up and is ready to use.

Setting up MDM Process Designer to use the Subversion Repository

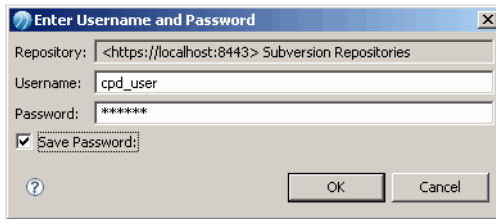
(Short description required.)

Procedure

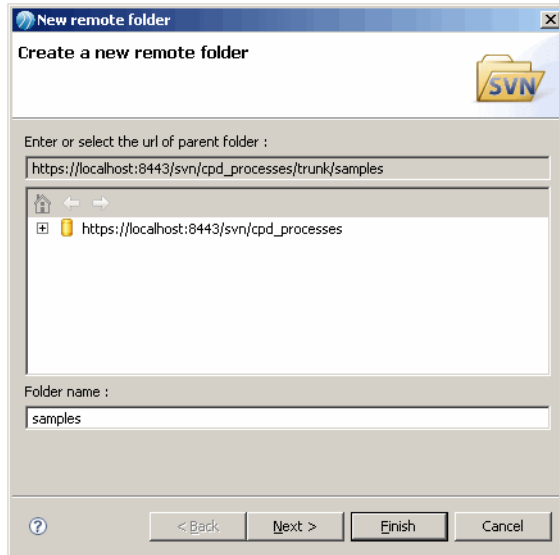
1. Start the MDM Process Designer. In the menu bar, select **Window-> Open Perspective -> Other** . In the Open Perspective dialog box that is displayed, select **SVN Repository Exploring**.



2. In the **SVN Repository** view displayed (initially empty) right click and select **New -> Repository Location** .
3. Enter the http base address, the username, and password for your SVN server in the dialog that is displayed. Select the **Save Password** checkbox. Click **OK**.



4. In the tree that is displayed, open the repository location and navigate to **trunk**. Right click it and select **New -> New Remote Folder** . Name this folder appropriately (here we use the name 'samples') and optionally add a commit comment. This is the folder you will share your process projects in.

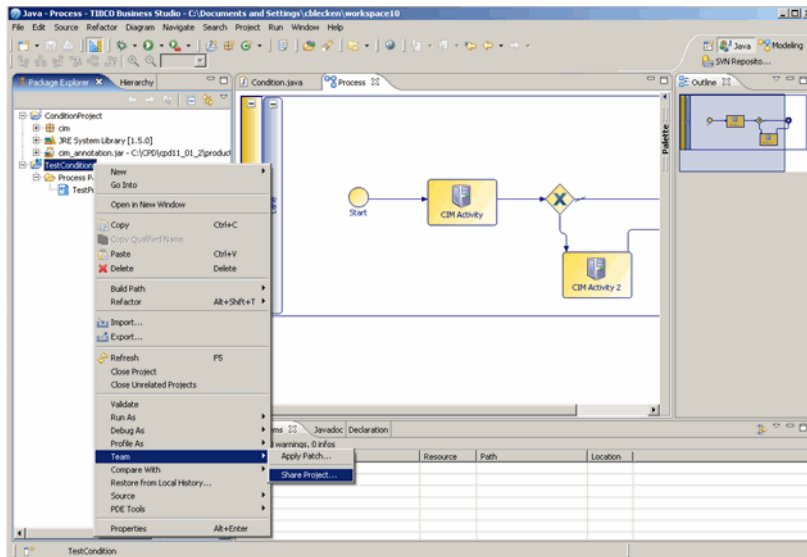


Sharing a Process Design Project

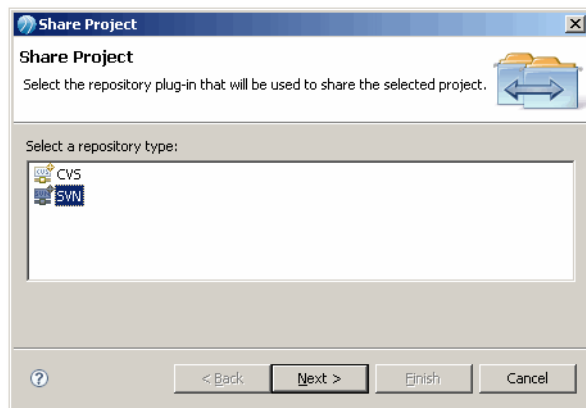
(Short description required.)

Procedure

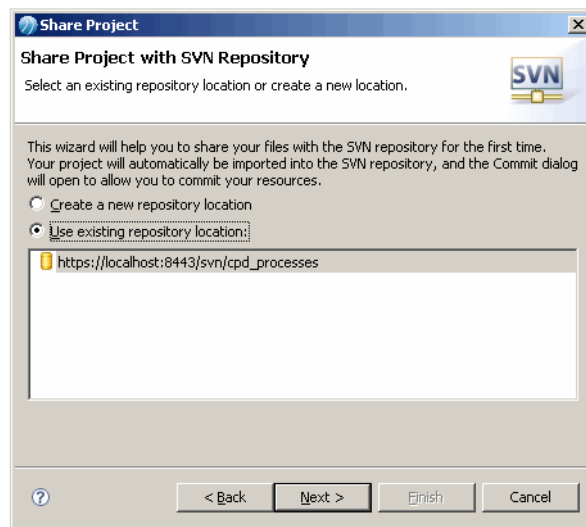
1. Switch to the modeling perspective and select the project you want to share. Right click the Project and select **Team -> Share Project** .



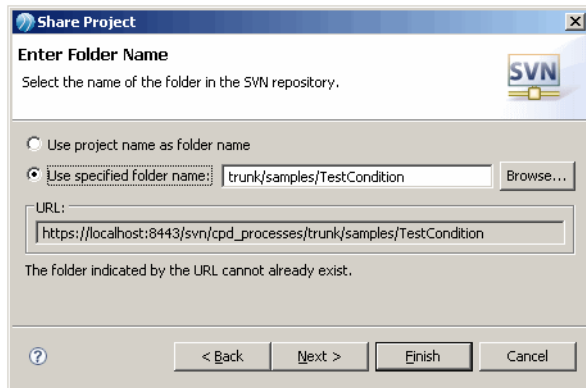
2. In the Share Project dialog, you will be prompted to select the repository plug-in to use the share the selected project. Select **SVN** and click **Next**.



3. You are then prompted to select an existing repository or create a new location. Select as appropriate and if using an existing repository, provide the repository name.

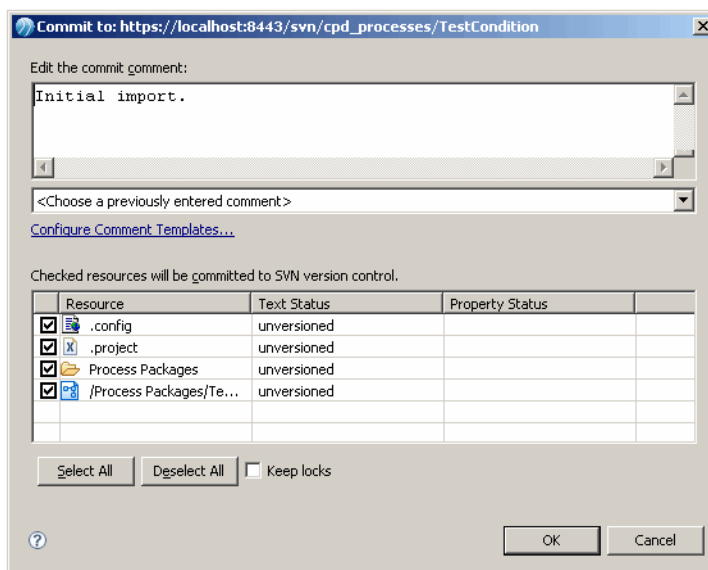


4. Add folder information for the remote repository. Click **Finish**.



5. Provide a commit comment and click **OK**. The project is now committed.

Result



Your project is now shared and the following section explains how other users can access it.

Accessing a shared project

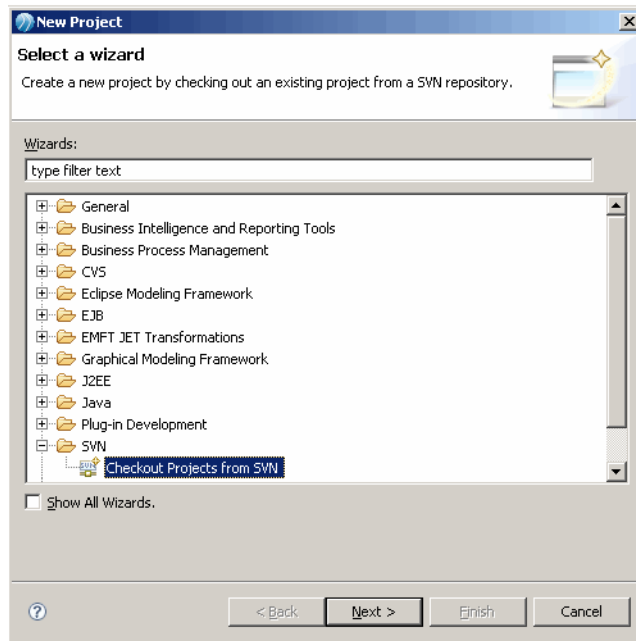
(Short description required.)



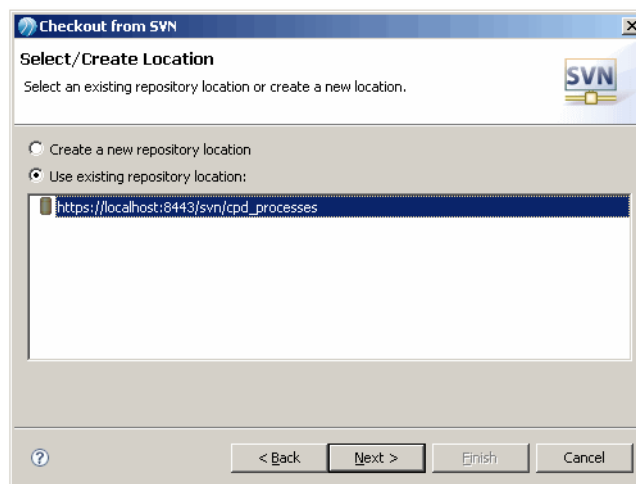
In order to access a previously committed project with a different CPD installation or with a different user/workspace, ensure you have completed the SVN repository setup as detailed in [Setting up MDM Process Designer to use the Subversion Repository](#).

Procedure

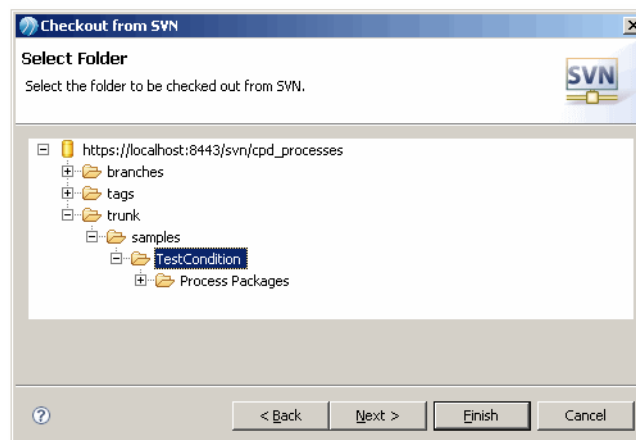
1. Select **File -> New -> Project**. Select **SVN-> Checkout Projects from SVN**. Click **Next**.



2. Select the repository to access by selecting the **Use existing repository location** option. Click **Next**.



3. Select the node in the tree control which indicates the project name. Click **Next** to map this project to your workspace.



4. Optionally, change the project name and click **Finish**. The project will now be available in the Modeling perspective.

