TIBCO® MDM Studio

Process Designer User Guide

Version 6.1.1 | June 2024

Document Updated | October 2024

Contents

Getting Started

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

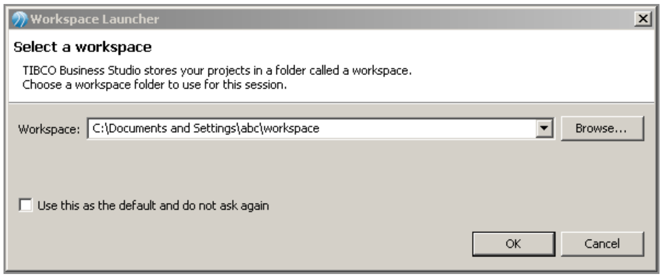
Start Process Designer

You can start Process Designer on Windows or Linux operating system.

For Windows

After the installation completes, start MDM Process Designer by selecting **Start** > **Program Files** > **TIBCO** > **Studio Designer**.

Provide a workspace location (folder where projects are saved).



Workspace Launcher

Select a workspace

TIBCO Business Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Documents and Settings\abc\workspace

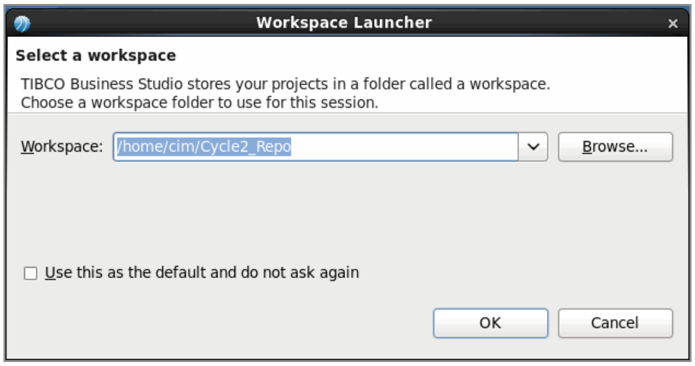
□ Use this as the default and do not ask again

TIBCO MDM Studio opens up and you are ready to start using it. For details on how to use the Process Designer, see Using the Process Designer.

For Linux

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

Provide a workspace location (folder where projects are saved).



Workspace Launcher

Select a workspace

TIBCO Business Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

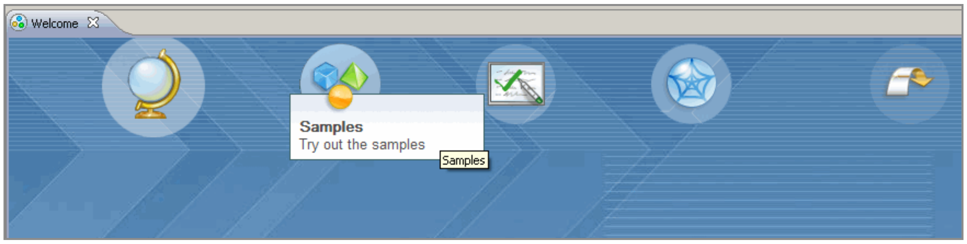
Workspace: /home/cim/Cycle2_Repo

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

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.



> **Note:** 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

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 TIBCO 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\<version>\samples folder.
5. Click **Finish**. The select project opens in the workspace.

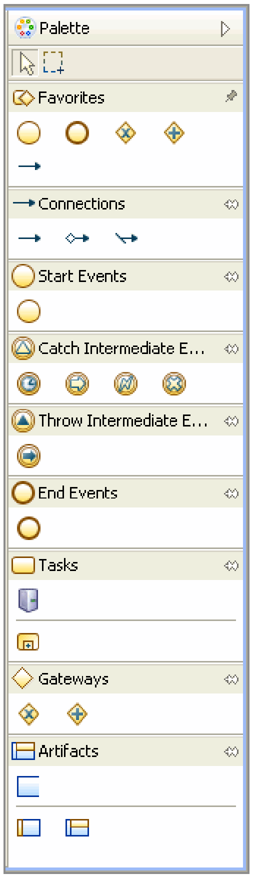
Result

Note: You need to import `AllECMClasses.jar` into java project to refer other mdm classes.

Palette

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.
ϟ	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.

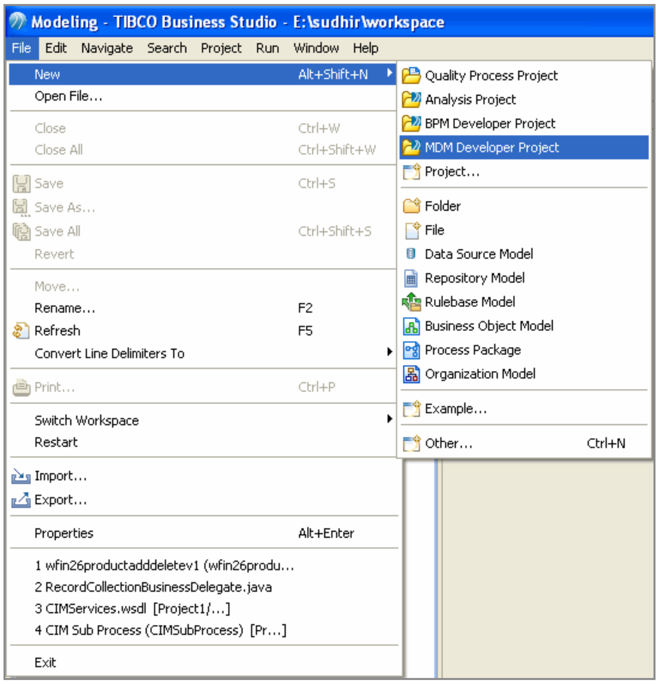
Process Designer

You can use Process Designer to graphically design processes.

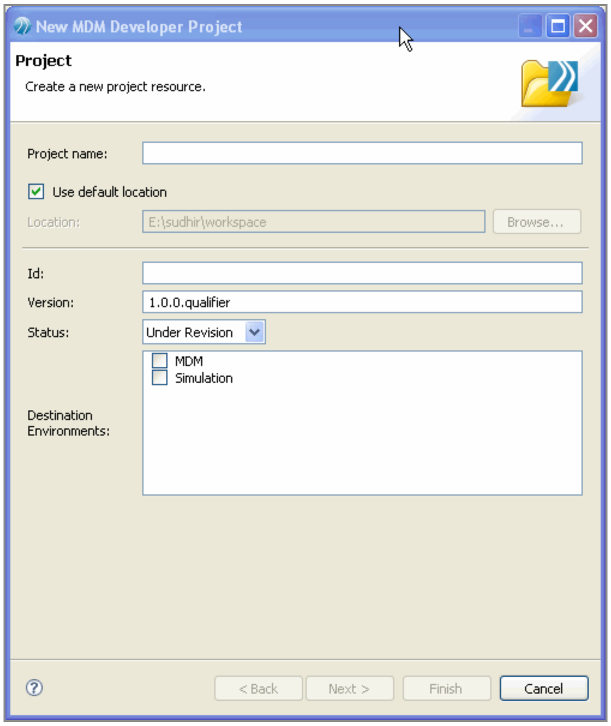
Creating a Project

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**.


New MDM Developer Project

Project

Create a new project resource.

Project name:

☑ Use default location

Location: E:\sudhir\workspace

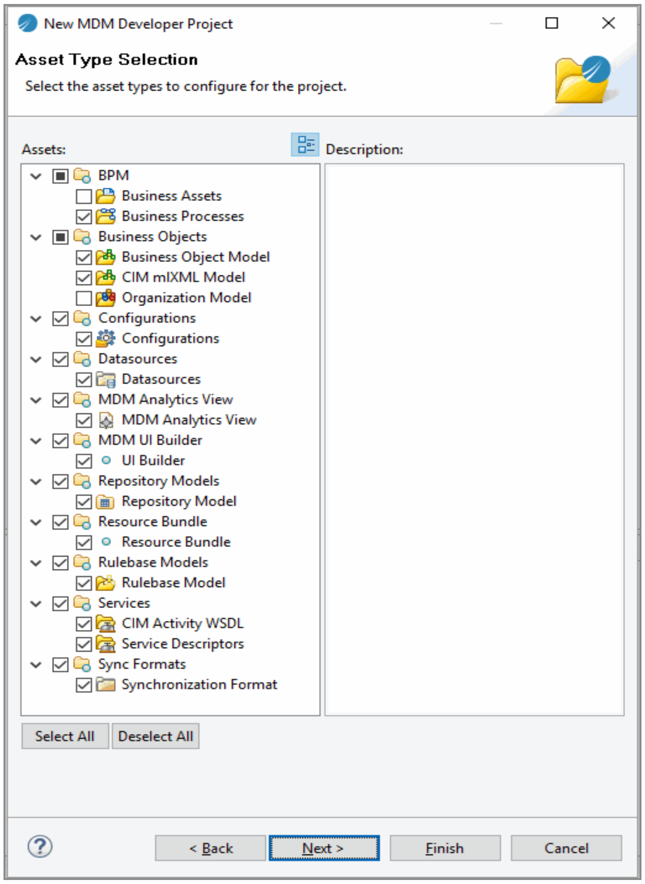
Id:

Version: 1.0.0.qualifier

Status: Under Revision

Destination Environments:
□ MDM
□ Simulation

3. The Asset Type Selection dialog is displayed. Ensure that the following is selected:
   - **Business Processes** (under BPM)
   - **Business Object Model** and **CIM mlXML Model** (under Business Objects)
   - **CIM Activity WSDL** and **Service Descriptors** (under Services)
4. Click **Next**.



New MDM Developer Project

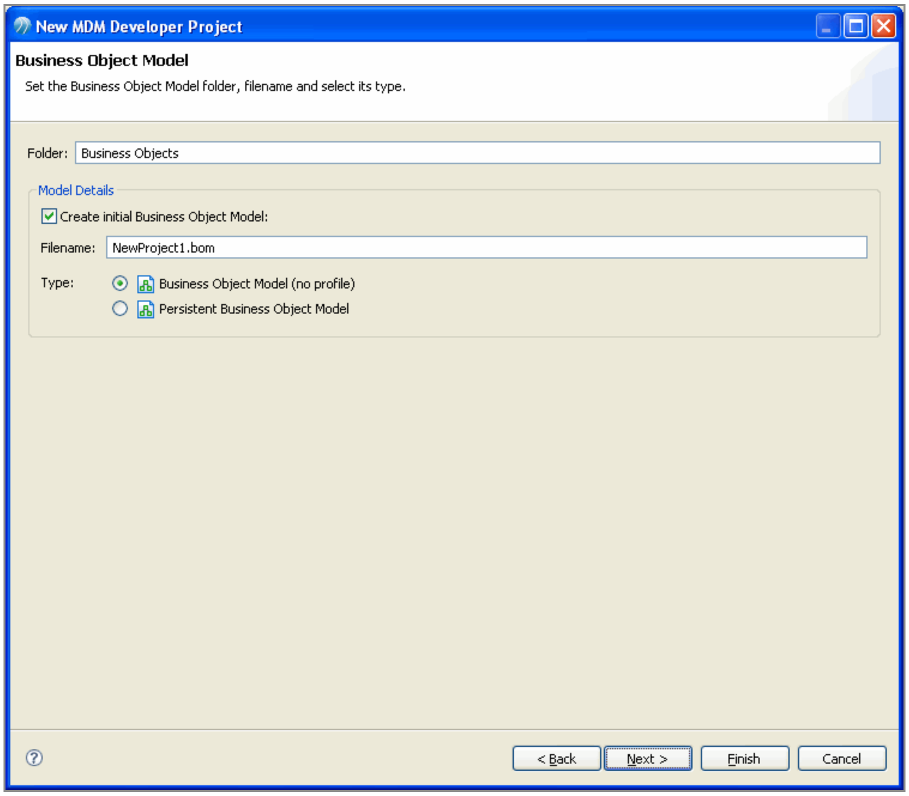
Asset Type Selection

Select the asset types to configure for the project.

Assets:

- ▣ BPM
  - ☐ Business Assets
  - ☑ Business Processes
- ▣ Business Objects
  - ☑ Business Object Model
  - ☑ CIM mIXML Model
  - ☐ Organization Model
- ☑ Configurations
  - ☑ Configurations
- ☑ Datasources
  - ☑ Datasources
- ☑ MDM Analytics View
  - ☑ MDM Analytics View
- ☑ MDM UI Builder
  - ☑ UI Builder
- ☑ Repository Models
  - ☑ Repository Model
- ☑ Resource Bundle
  - ☑ Resource Bundle
- ☑ Rulebase Models
  - ☑ Rulebase Model
- ☑ Services
  - ☑ CIM Activity WSDL
  - ☑ Service Descriptors
- ☑ Sync Formats
  - ☑ Synchronization Format

5. 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**.



Business Object Model

Set the Business Object Model folder, filename and select its type.

Folder: Business Objects

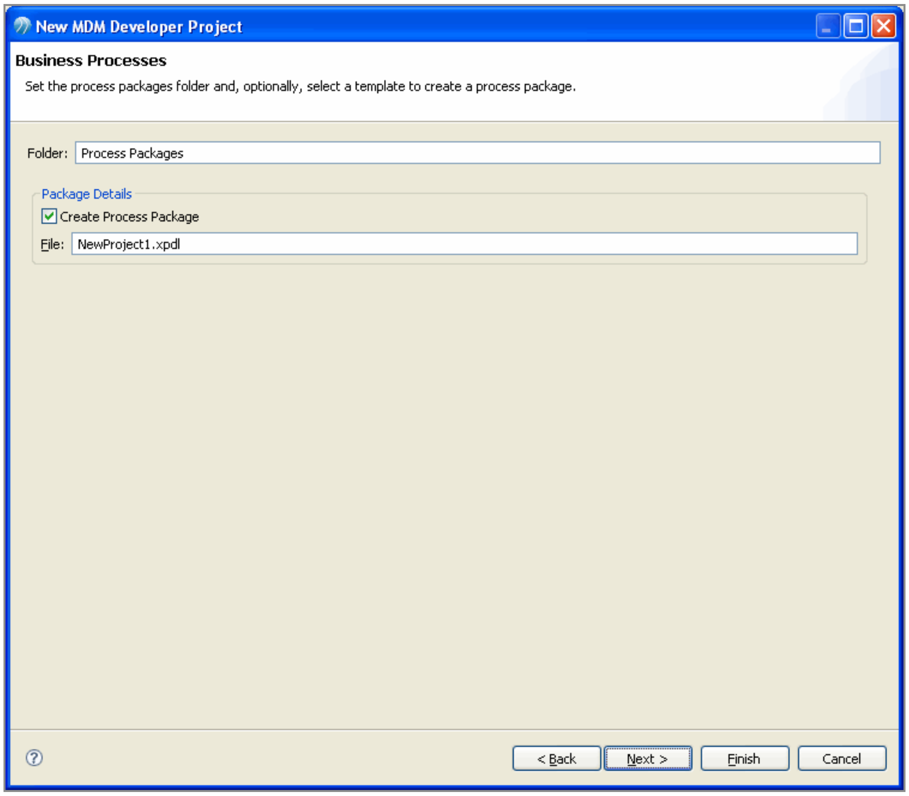
Model Details

☑ Create initial Business Object Model:

Filename: NewProject1.bom

Type: ● Business Object Model (no profile)
○ Persistent Business Object Model

6. 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 xpdl file. Click **Next**.



New MDM Developer Project

Business Processes

Set the process packages folder and, optionally, select a template to create a process package.

Folder: Process Packages

Package Details

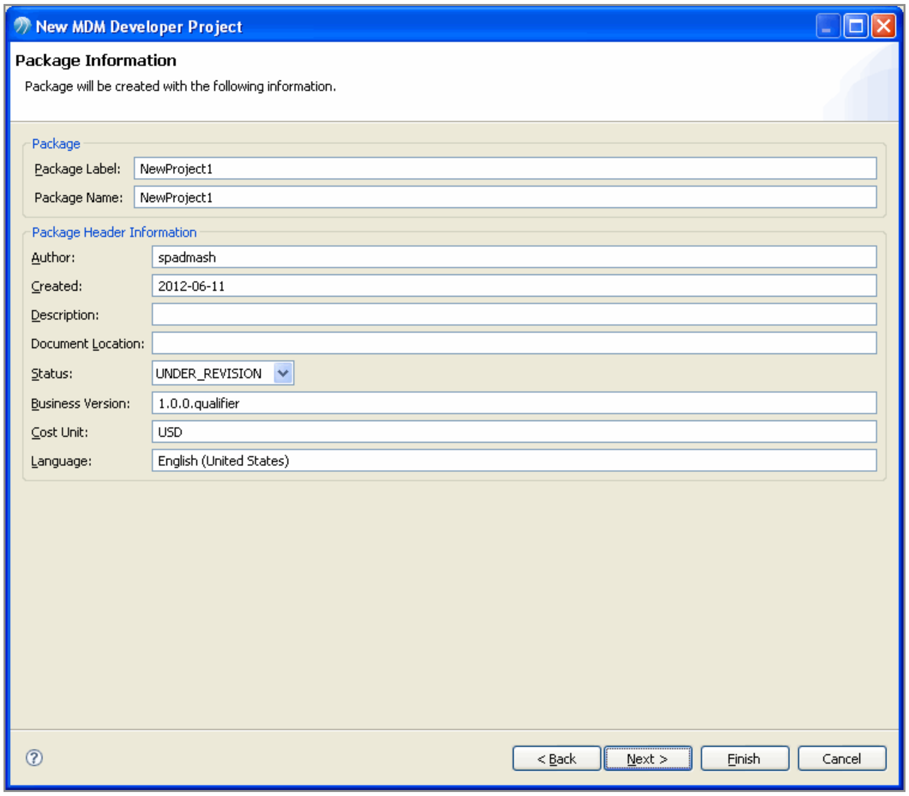
☑ Create Process Package

File: NewProject1.xpdl

7. 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.



Package Information

Package will be created with the following information.

Package

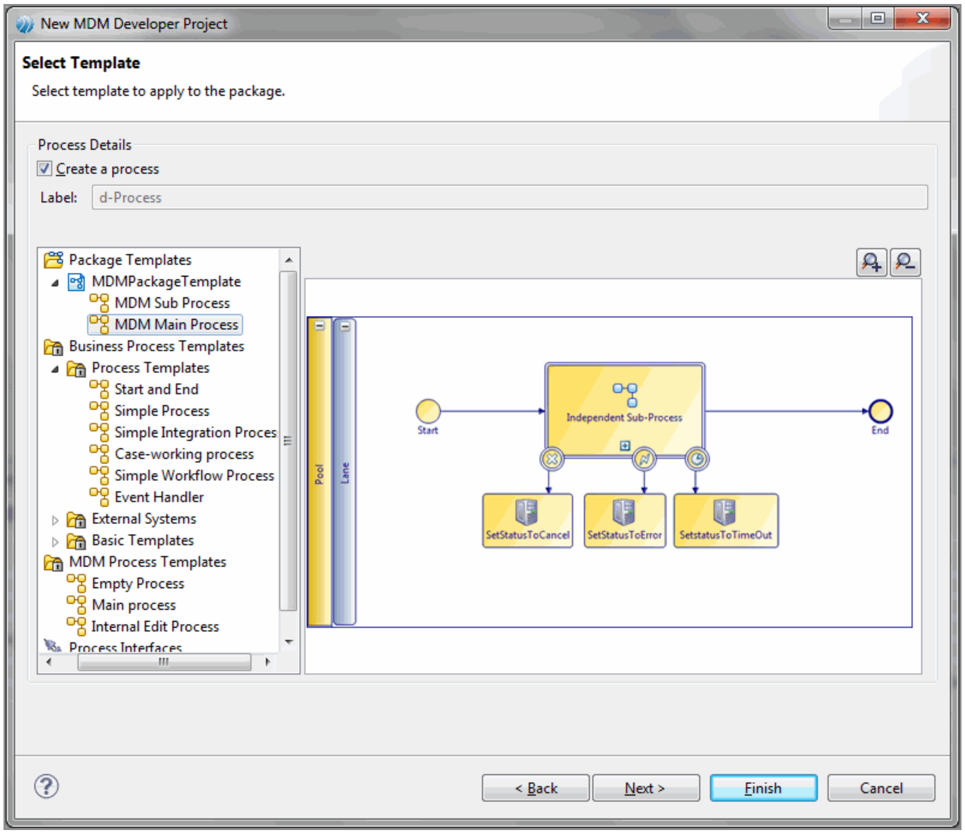
Package Label:	NewProject1
Package Name:	NewProject1

Package Header Information

Author:	spadmash
Created:	2012-06-11
Description:	
Document Location:	
Status:	UNDER_REVISION
Business Version:	1.0.0.qualifier
Cost Unit:	USD
Language:	English (United States)

8. 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: d-Process

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

A brief description of projects and their components.

A Project follows the following hierarchy: **Project > Process Packages > Processes**

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

Project

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

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

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

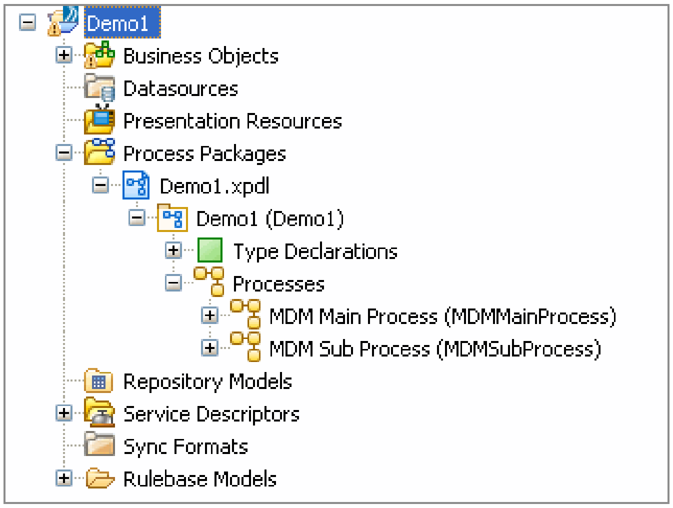
Project skeleton using the MDM Template

You can use MDM template 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.

Figure 1: Process Designer Sample Main (Shell) Process and sub process



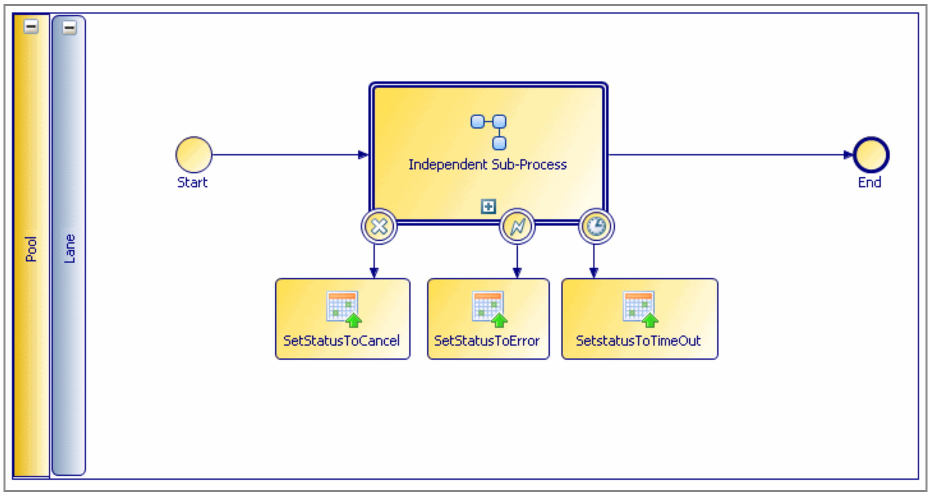
If you double-click the Main MDM process in the Project Explorer, the flow details are displayed in an editor that comes up on the right; this is the primary editor that allows you to graphically layout your process.

Main MDM Process

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.

Figure 2: Main Process

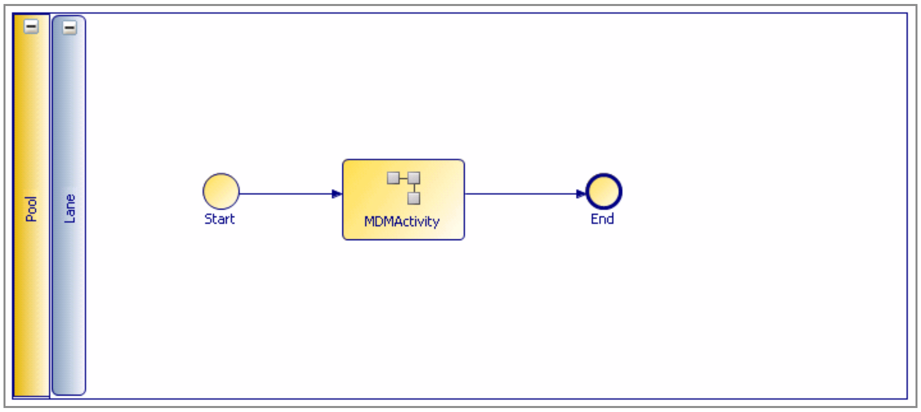


Sub Process

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.

Figure 3: Sub process



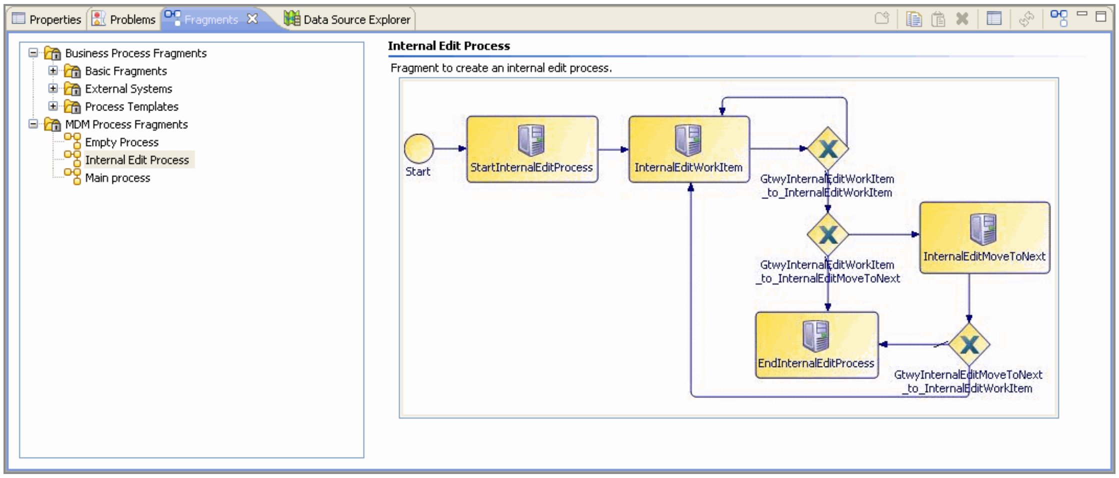
MDM Process Templates

You can use the template to quickly create new processes rather than reusing an entire process.

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

Figure 4: Fragment View


Properties
Problems
Fragments
Data Source Explorer

- Business Process Fragments
  - Basic Fragments
  - External Systems
  - Process Templates
- MDM Process Fragments
  - Empty Process
  - Internal Edit Process
  - Main process

Internal Edit Process

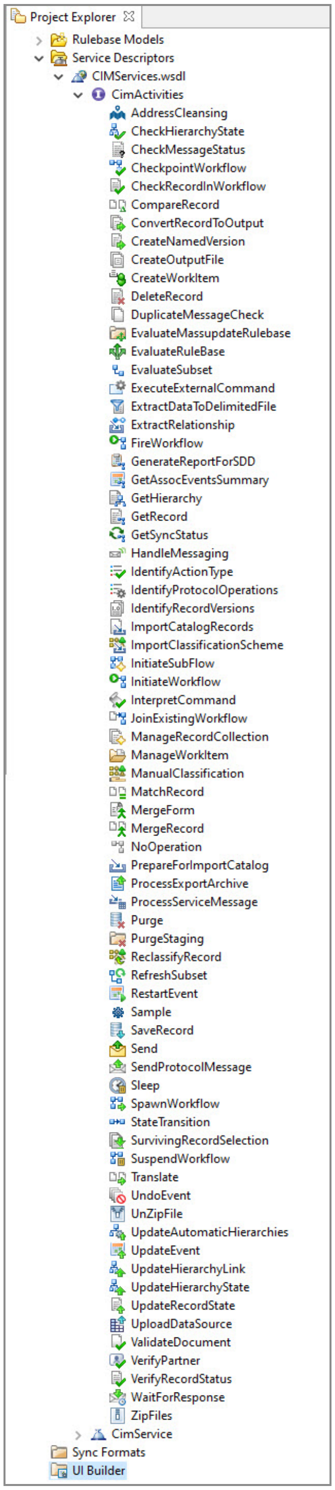
Fragment to create an internal edit process.

Start
StartInternalEditProcess
InternalEditWorkItem
GtwyInternalEditWorkItem _to_InternalEditWorkItem
GtwyInternalEditWorkItem _to_InternalEditMoveToNext
InternalEditMoveToNext
EndInternalEditProcess
GtwyInternalEditMoveToNext _to_InternalEditWorkItem

CIM Services WSDL

The CIMServices.wsdl contains all the TIBCO 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.

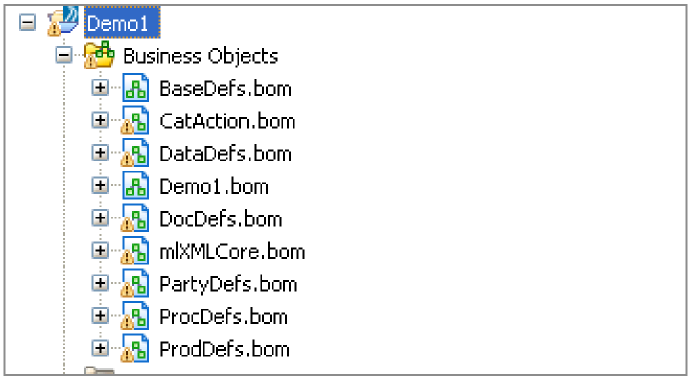


Project Explorer

- Rulebase Models
- Service Descriptors
  - CIMServices.wsdl
    - CimActivities
      - AddressCleansing
      - CheckHierarchyState
      - CheckMessageStatus
      - CheckpointWorkflow
      - CheckRecordInWorkflow
      - CompareRecord
      - ConvertRecordToOutput
      - CreateNamedVersion
      - CreateOutputFile
      - CreateWorkItem
      - DeleteRecord
      - DuplicateMessageCheck
      - EvaluateMassupdateRulebase
      - EvaluateRuleBase
      - EvaluateSubset
      - ExecuteExternalCommand
      - ExtractDataToDelimitedFile
      - ExtractRelationship
      - FireWorkflow
      - GenerateReportForSDD
      - GetAssocEventsSummary
      - GetHierarchy
      - GetRecord
      - GetSyncStatus
      - HandleMessaging
      - IdentifyActionType
      - IdentifyProtocolOperations
      - IdentifyRecordVersions
      - ImportCatalogRecords
      - ImportClassificationScheme
      - InitiateSubFlow
      - InitiateWorkflow
      - InterpretCommand
      - JoinExistingWorkflow
      - ManageRecordCollection
      - ManageWorkItem
      - ManualClassification
      - MatchRecord
      - MergeForm
      - MergeRecord
      - NoOperation
      - PrepareForImportCatalog
      - ProcessExportArchive
      - ProcessServiceMessage
      - Purge
      - PurgeStaging
      - ReclassifyRecord
      - RefreshSubset
      - RestartEvent
      - Sample
      - SaveRecord
      - Send
      - SendProtocolMessage
      - Sleep
      - SpawnWorkflow
      - StateTransition
      - SurvivingRecordSelection
      - SuspendWorkflow
      - Translate
      - UndoEvent
      - UnZipFile
      - UpdateAutomaticHierarchies
      - UpdateEvent
      - UpdateHierarchyLink
      - UpdateHierarchyState
      - UpdateRecordState
      - UploadDataSource
      - ValidateDocument
      - VerifyPartner
      - VerifyRecordStatus
      - WaitForResponse
      - ZipFiles
    - CimService
- Sync Formats
- UI Builder

For more details, see Using the Process Designer and Parameters.

BOM Files



- Demo1
  - Business Objects
    - BaseDefs.bom
    - CatAction.bom
    - DataDefs.bom
    - Demo1.bom
    - DocDefs.bom
    - mlXMLCore.bom
    - PartyDefs.bom
    - ProcDefs.bom
    - ProdDefs.bom

Note: 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 an TIBCO MDM process

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

Each MDM activity is expressed as a web service operation.

The TIBCO MDM Process Designer uses Web Service Description Language (WSDL) for definition of MDM activities.

For more details, see Activity Details.

Adding Activities

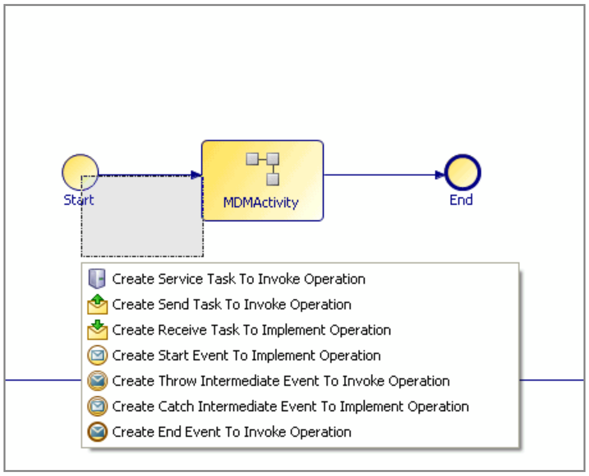
There are two ways you can insert activities:

Adding a New Activity - Method 1

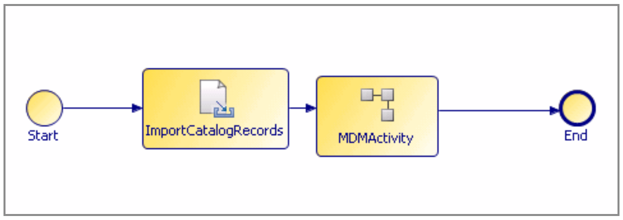
You can insert an activity using this method.

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.


Start
MDMActivity
End
Create Service Task To Invoke Operation
Create Send Task To Invoke Operation
Create Receive Task To Implement Operation
Create Start Event To Implement Operation
Create Throw Intermediate Event To Invoke Operation
Create Catch Intermediate Event To Implement Operation
Create End Event To Invoke Operation

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

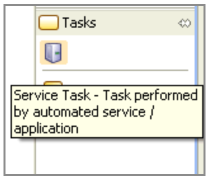


Adding a New Activity - Method 2

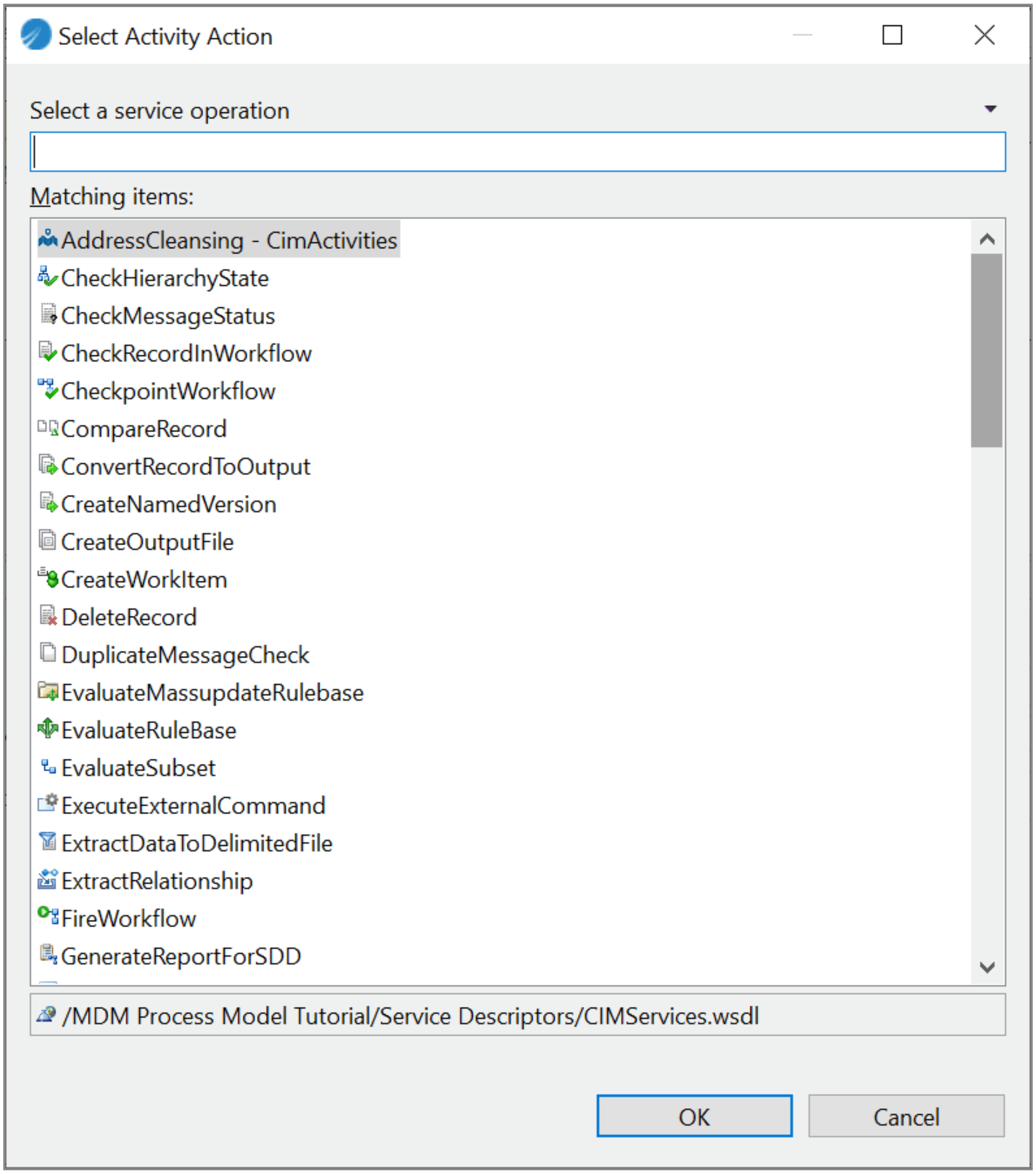
You can insert an activity using this method.

Procedure

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


Tasks
Service Task - Task performed by automated service / application

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

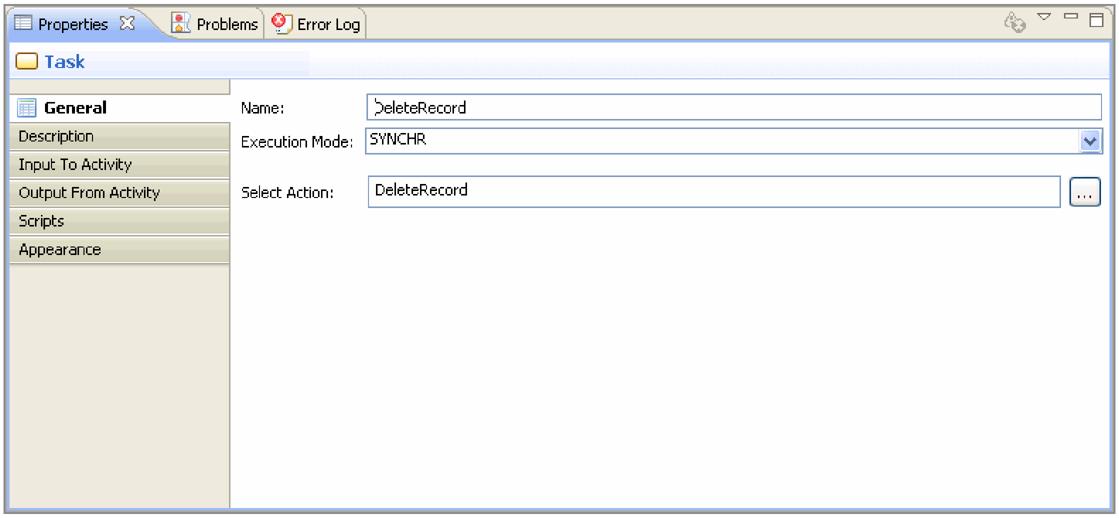
You can modify an activity by editing the parameters from the property section.

Activity Details

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.

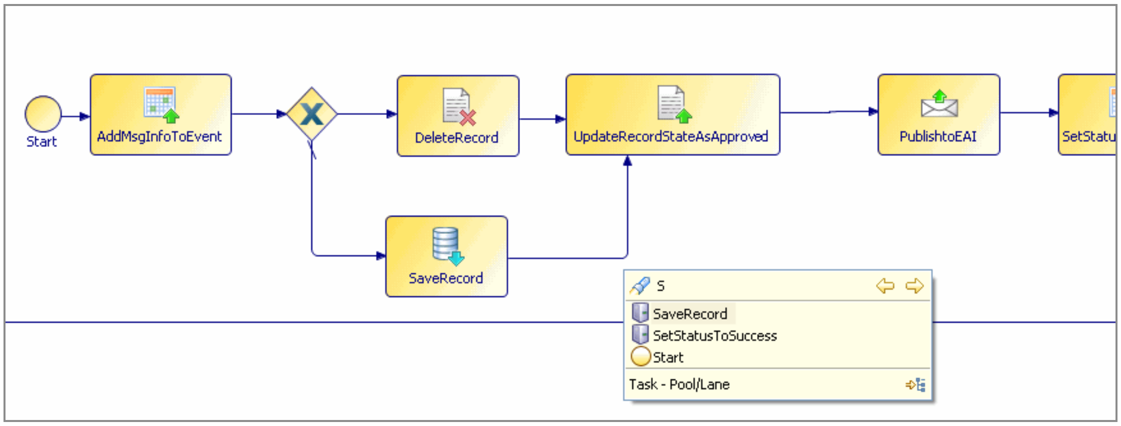


Name: DeleteRecord
Execution Mode: SYNCHR
Select Action: DeleteRecord

Search for activity in a workflow

Use Ctrl+F to search for a specific activity in a workflow.

Ensure that the drawing pane is selected.

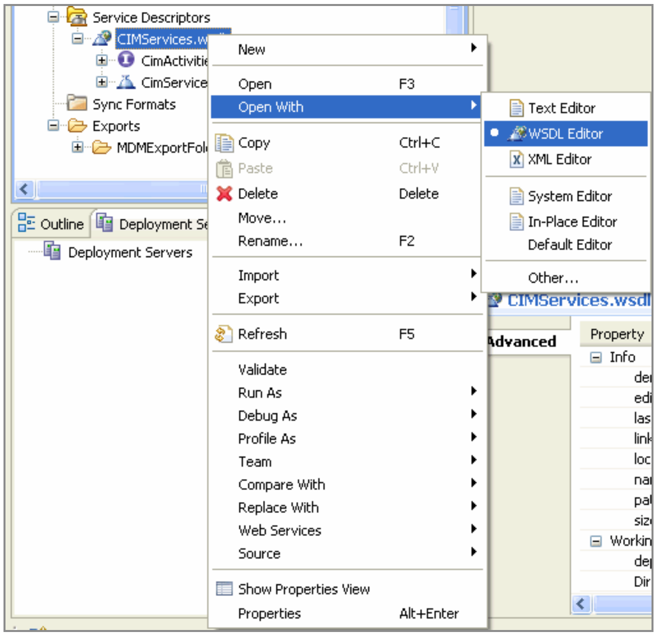


Defining Custom Activities

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

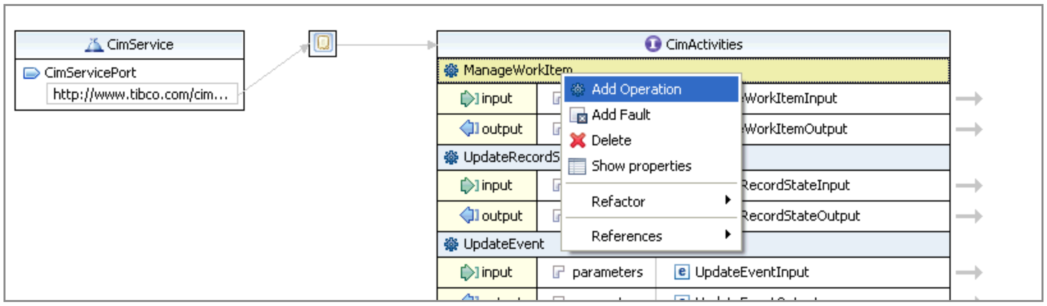
Procedure

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

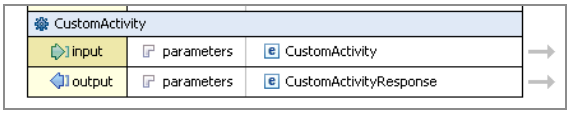


2. The list of MDM activities is displayed - right click any activity and select **Add**

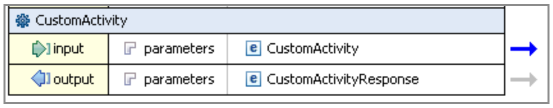
Operation.



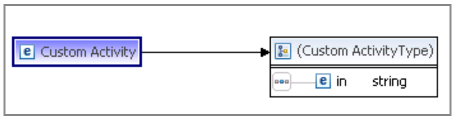
3. The new activity (called **NewOperation** by default) gets added at the bottom; you can rename the activity, for example **CustomActivity** as below.



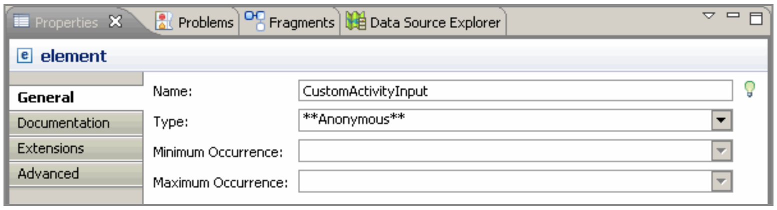
4. Click the arrow to the right of **input** in the newly added custom activity.



5. The inline schema will be displayed.

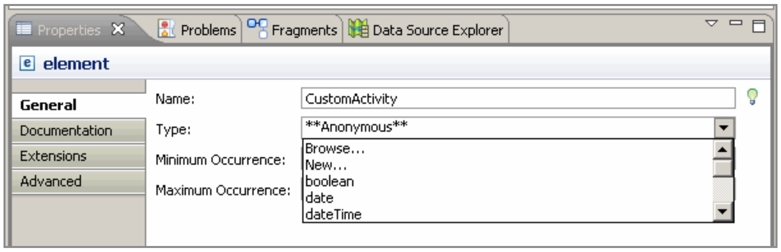


6. In the **Properties** window, **General** tab, change the **Name** to <ActivityName>Input. For example, **CustomActivityInput**.



Name: CustomActivityInput
Type: **Anonymous**
Minimum Occurrence:
Maximum Occurrence:

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

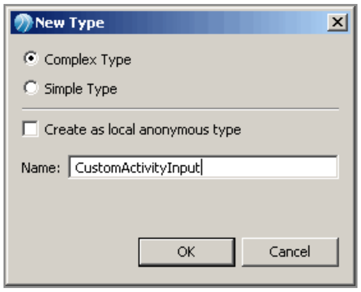


Name: CustomActivity

Type: **Anonymous**

Browse...
New...
boolean
date
dateTime

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



● Complex Type
○ Simple Type

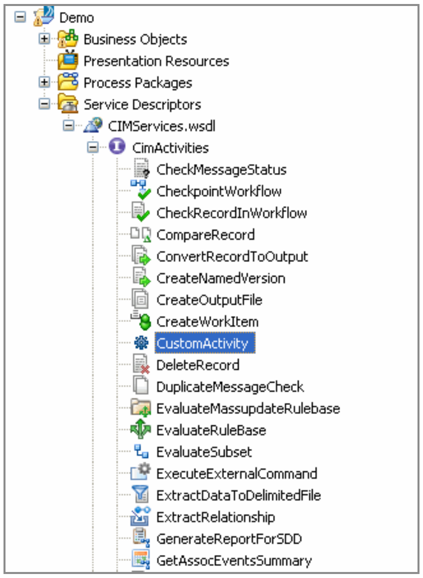
□ Create as local anonymous type

Name: CustomActivityInput

9. 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.

10. You can now add input and output parameters to the custom activity. For more information on adding parameters, see Adding and Modifying Custom Parameters.

11. Once you save the wsdl, the newly defined activities will be displayed in the list of activities in **CimActivities** under **Service Descriptors**.



- Demo
  - Business Objects
  - Presentation Resources
  - Process Packages
  - Service Descriptors
    - CIMServices.wsdl
      - CimActivities
        - CheckMessageStatus
        - CheckpointWorkflow
        - CheckRecordInWorkflow
        - CompareRecord
        - ConvertRecordToOutput
        - CreateNamedVersion
        - CreateOutputFile
        - CreateWorkItem
        - CustomActivity
        - DeleteRecord
        - DuplicateMessageCheck
        - EvaluateMassupdateRulebase
        - EvaluateRuleBase
        - EvaluateSubset
        - ExecuteExternalCommand
        - ExtractDataToDelimitedFile
        - ExtractRelationship
        - GenerateReportForSDD
        - GetAssocEventsSummary

InterpretCommand Activity

You can execute a script and retrieve results using InterpretCommand activity.

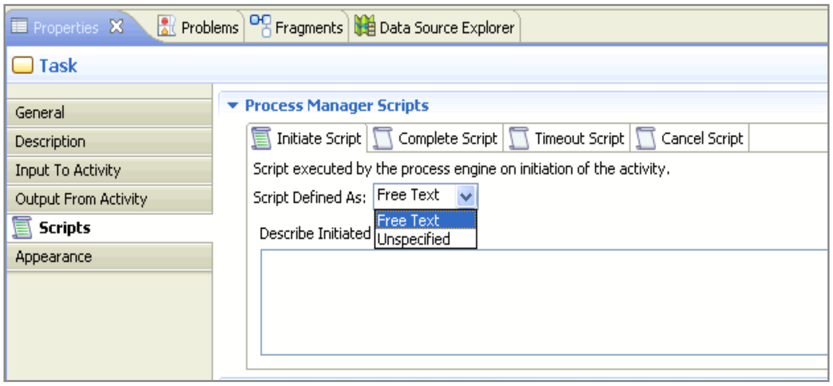
Input parameters are passed to the script, and output parameters are assigned values equal to the variables inside the script.

Creating InterpretCommand activity

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** drop down. You can then enter your script.



Script executed by the process engine on initiation of the activity.

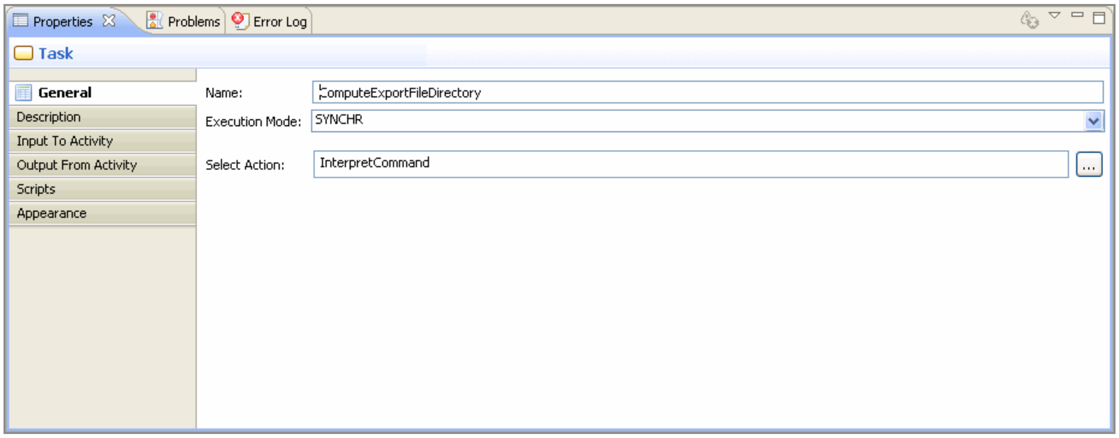
Script Defined As: Free Text

Describe Initiated

Example - InterpretCommand

Procedure

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

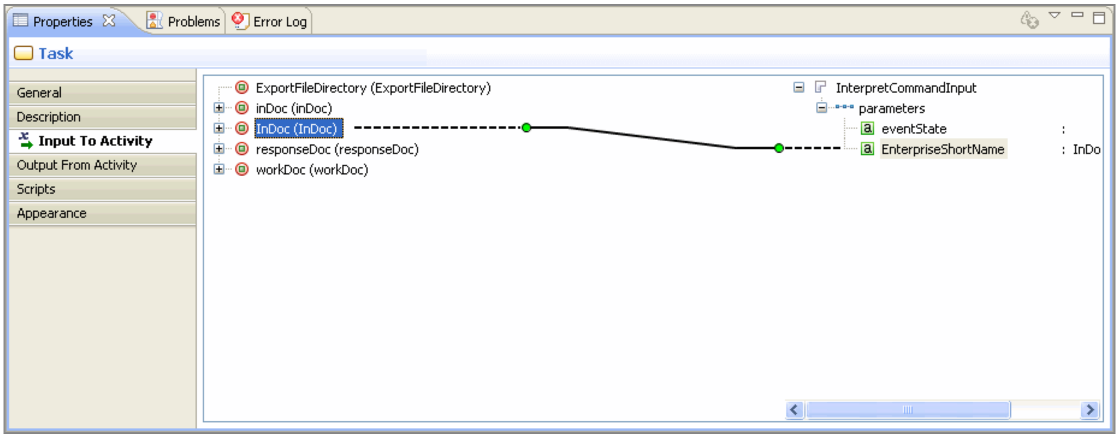


Name: computeExportFileDirectory

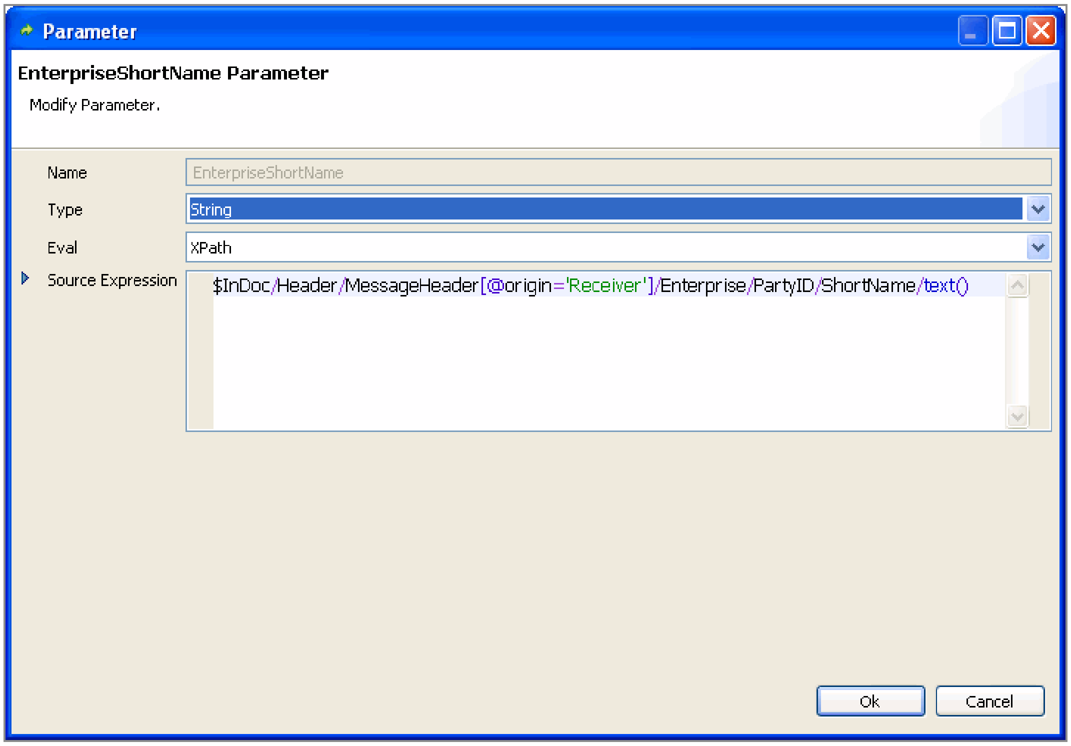
Execution Mode: SYNCHR

Select Action: InterpretCommand

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**.



Name: EnterpriseShortName

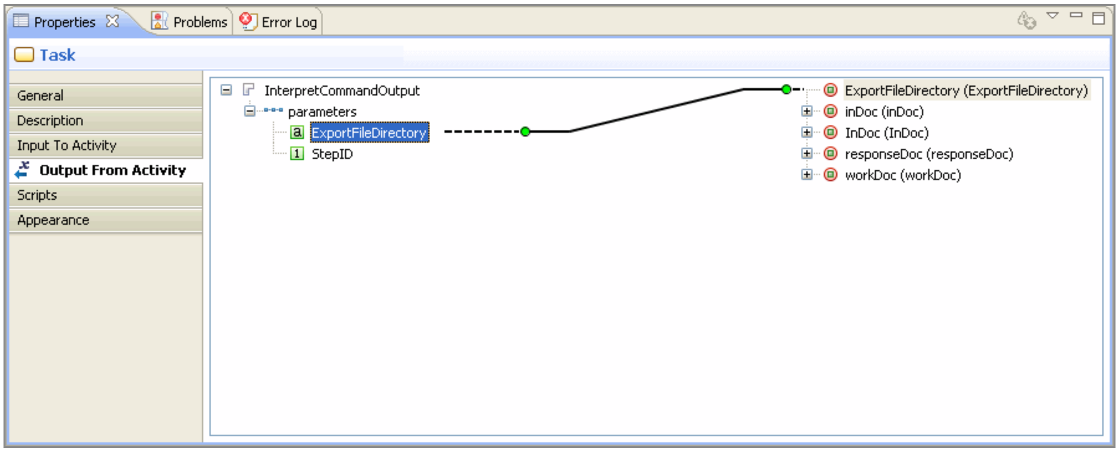
Type: String

Eval: XPath

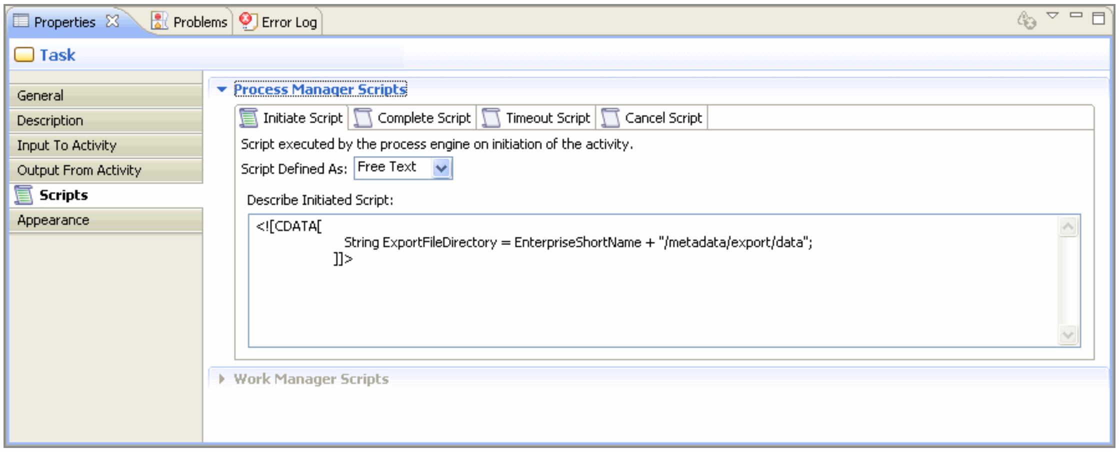
Source Expression:

```
$InDoc/Header/MessageHeader[@origin='Receiver']/Enterprise/PartyID/ShortName/text()
```

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** drop-down**.** You can then enter your script.



Script executed by the process engine on initiation of the activity.

Script Defined As: Free Text

Describe Initiated Script:

```
<![CDATA[
            String ExportFileDirectory = EnterpriseShortName + "/metadata/export/data";
        ]]>
```

Subflows

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.

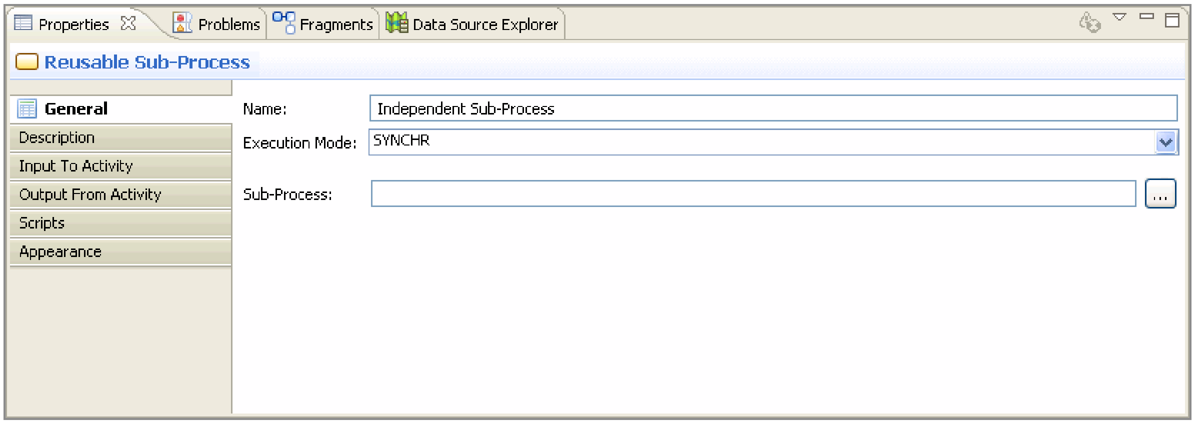
There are two ways you can define subflows in the Process Designer:

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

You can have multiple Subflow activities defined for a workflow.

Using Reusable Sub Process

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



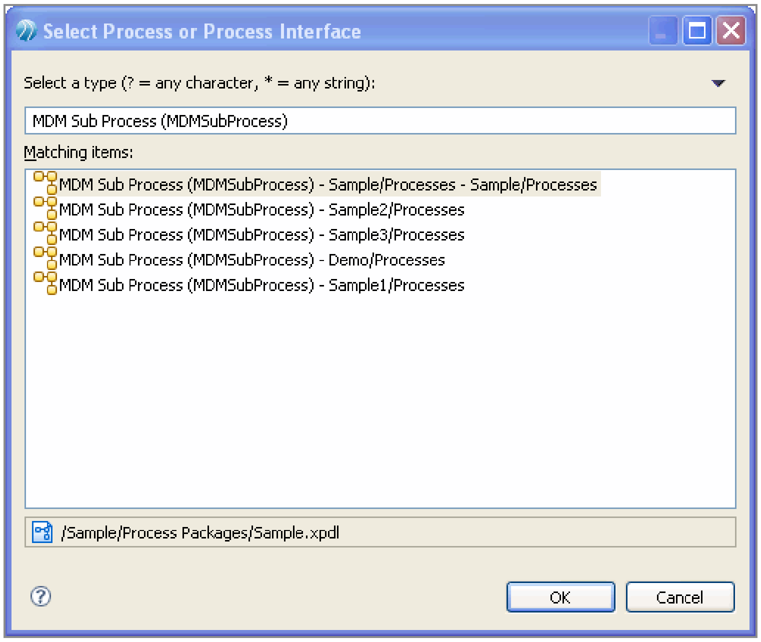
Name: Independent Sub-Process

Execution Mode: SYNCHR

Sub-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**.



Select a type (? = any character, * = any string):

MDM Sub Process (MDMSubProcess)

Matching items:

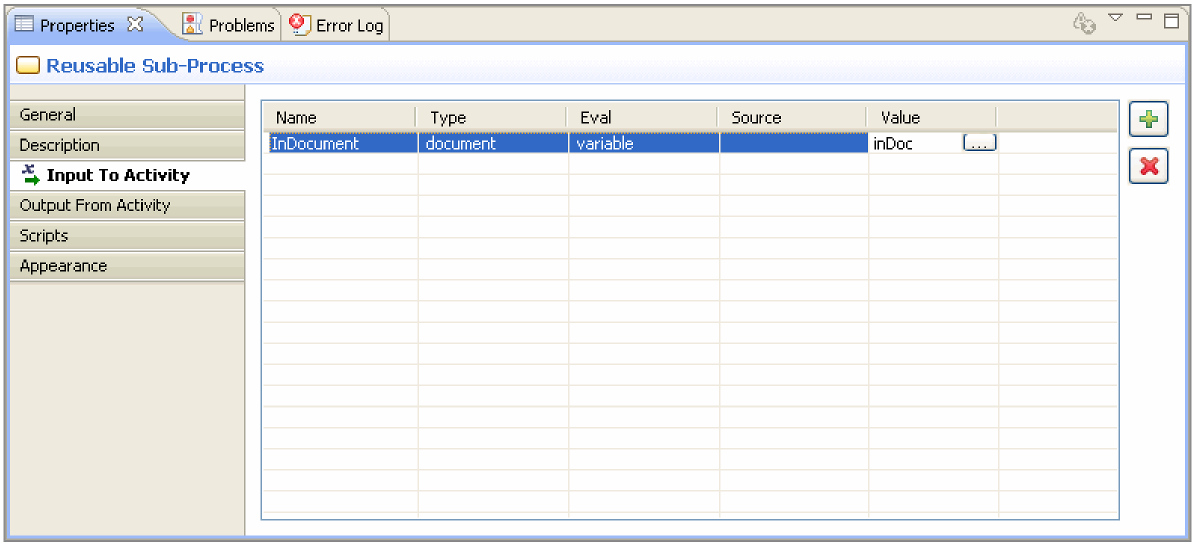
MDM Sub Process (MDMSubProcess) - Sample/Processes - Sample/Processes
MDM Sub Process (MDMSubProcess) - Sample2/Processes
MDM Sub Process (MDMSubProcess) - Sample3/Processes
MDM Sub Process (MDMSubProcess) - Demo/Processes
MDM Sub Process (MDMSubProcess) - Sample1/Processes

/Sample/Process Packages/Sample.xpdl

4. Click **OK**.

Defining Input Parameters

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.

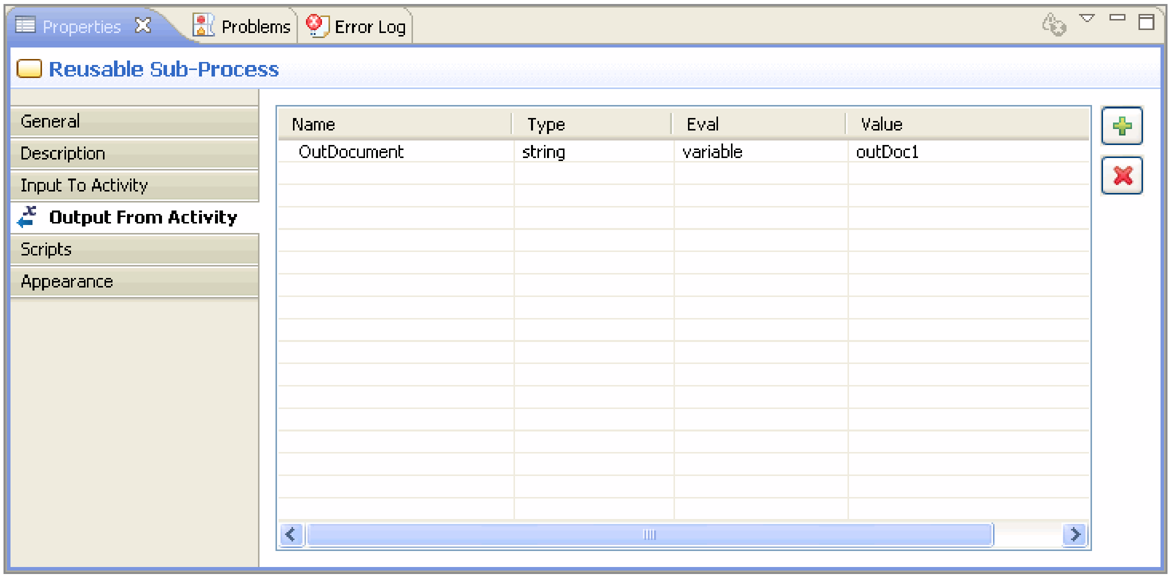


Name	Type	Eval	Source	Value
InDocument	document	variable		inDoc

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

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.



Name	Type	Eval	Value
OutDocument	string	variable	outDoc1

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

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

Activity parameters available in the Process Designer.

Parameters and Variables

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

Local Activity Parameters

Each TIBCO 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

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.

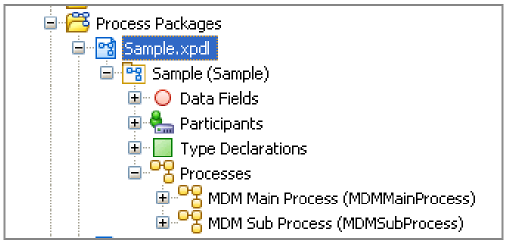
Global Parameters

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.

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.

Defining Global Variables

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



Procedure

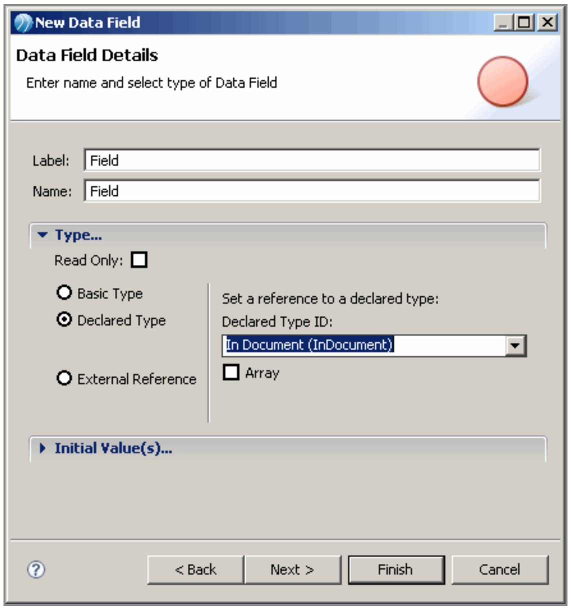
1. To add a global variable (new data field), right-click on Package (For example, Sample) and select **New > Data field**. This starts up the new data field wizard.

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.



Label: Field

Name: Field

Read Only: □

○ Basic Type

● Declared Type

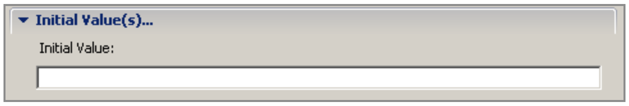
○ External Reference

Set a reference to a declared type:

Declared Type ID: In Document (InDocument)

□ Array

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.



Initial Value:

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

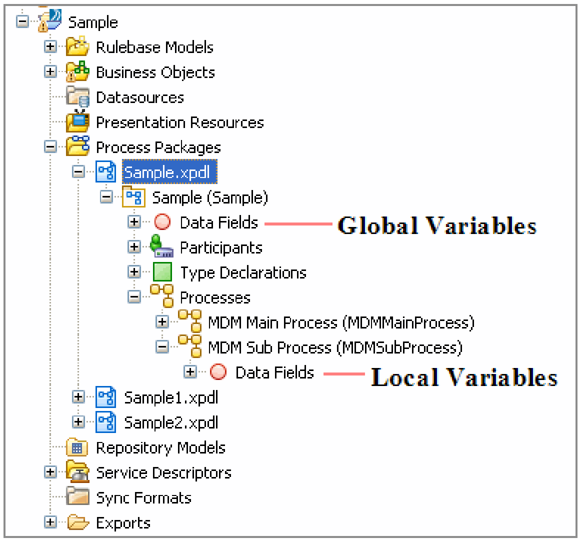
Result

You can map parameters to global variables as required.

Defining Local Variables

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

Input parameters define the data that is put input to the process.

Defining Values for Input Parameters

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.
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 *TIBCO MDM Workflow guide*.
Compute	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 = Launch Date(output from rulebase) -

Eval Mode	Description
	RemindBeforeNumberOfDays 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 revivified, 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.
Event	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. 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 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.
UserProfile	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,

Eval Mode	Description
	Enterprise Name, Enterprise Vertical, Formatted User Name, Organization ID, Organization Name, Organization Type, User ID, and User Name. 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. 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.
System	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. 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 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.

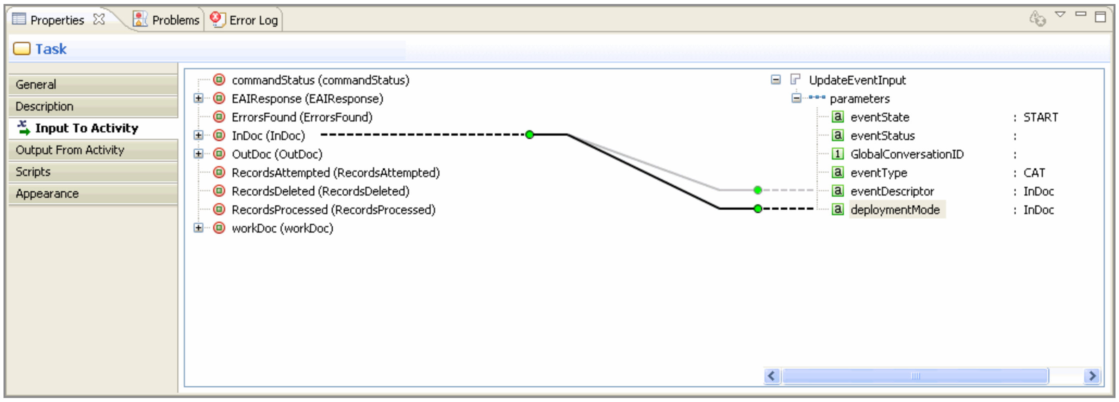
Example 1 - Setting the eventState

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.



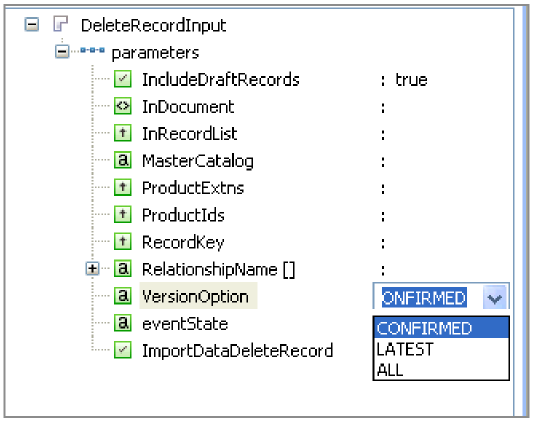
```
commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

UpdateEventInput
  parameters
    eventState             : START
    eventStatus            :
    GlobalConversationID   :
    eventType              : CAT
    eventDescriptor        : InDoc
    deploymentMode         : InDoc
```

Example 2 - Setting the VersionOption

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

Select the VersionOption parameter under **parameters**.

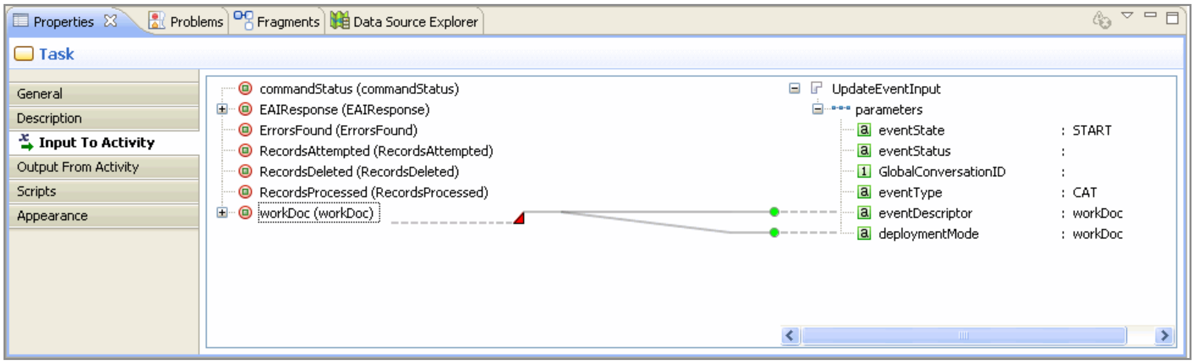
Click on the drop-down list corresponding to the VerisonOption parameter and select **LATEST**.



```
DeleteRecordInput
  parameters
    IncludeDraftRecords      : true
    InDocument               :
    InRecordList             :
    MasterCatalog            :
    ProductExtns             :
    ProductIds               :
    RecordKey                :
    RelationshipName []      :
    VersionOption            ONFIRMED
    eventState               CONFIRMED
    ImportDataDeleteRecord   LATEST
                             ALL
```

Viewing Input Parameters

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.



commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

UpdateEventInput
parameters

eventState	: START
eventStatus	:
GlobalConversationID	:
eventType	: CAT
eventDescriptor	: workDoc
deploymentMode	: workDoc

A complete list of parameters for the selected activity is 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

Output parameters define the data that is returned as output by the process.

Defining Values for Output Parameters

Procedure

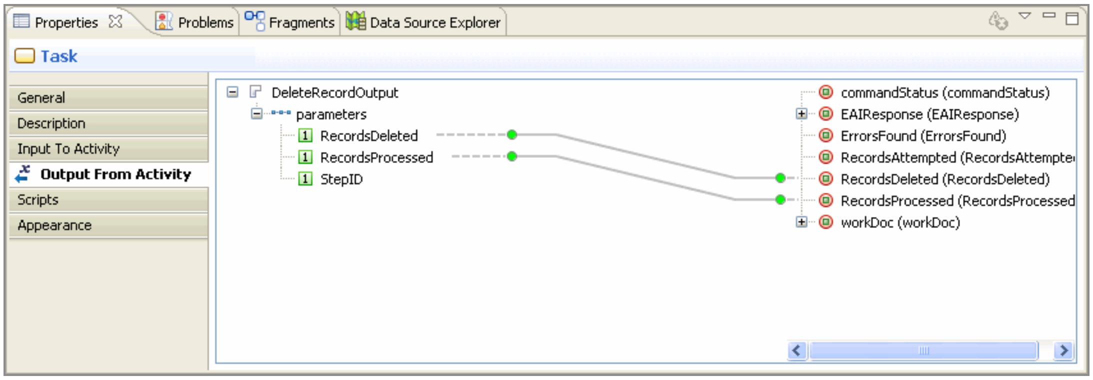
1. Select the activity in the process flow.
2. In the **Properties** Window, go to the **Output From Activity** tab.
3. Expand the *activityname* output parameters on the left.
4. Select the parameter that you want to provide a value to.
5. Drag it onto a data field on the right to map it.

Result

Note: Output parameters can only be mapped to Data Fields.

Viewing Output Parameters

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

You can provide a new value or change the default type of a parameter.

Adding Parameter for eval mode=Variable

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.

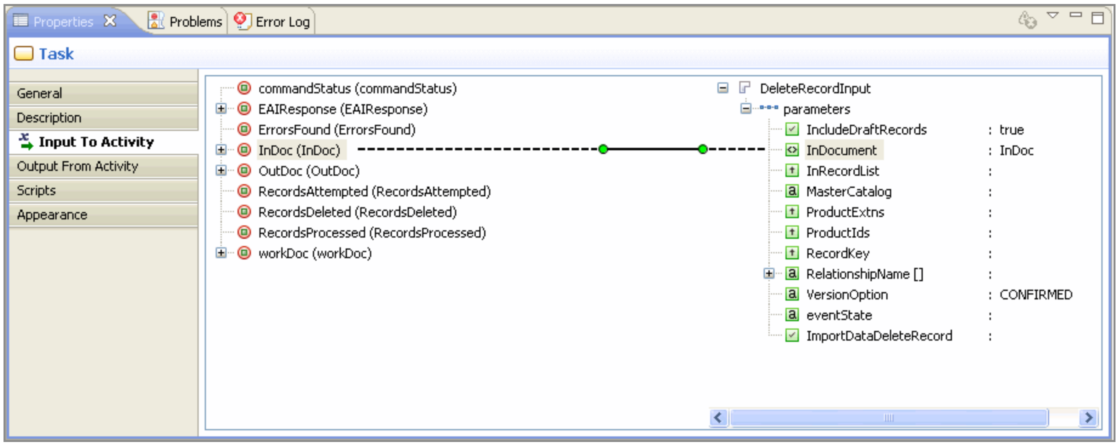
Procedure

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.

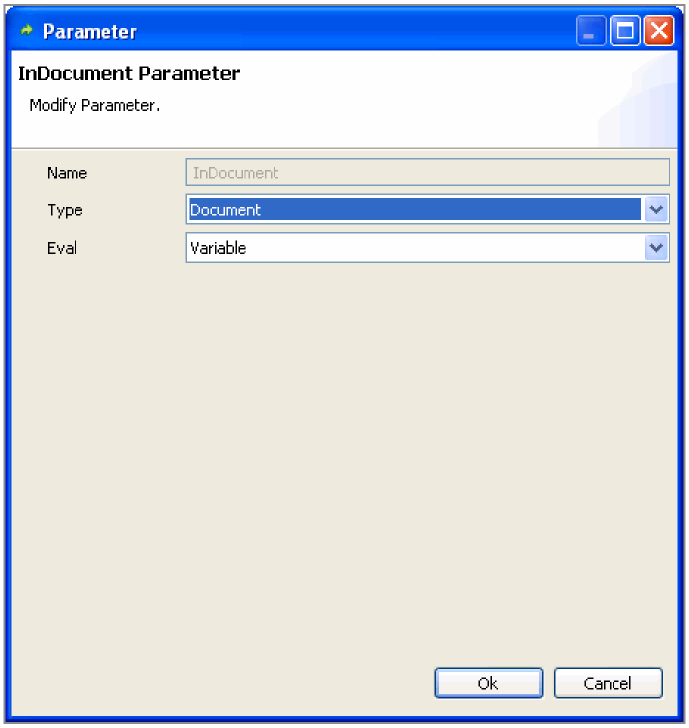
4. Select the parameter that you want to provide a value to and drag it onto DataField (Workflow state variable) on the left.



commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

DeleteRecordInput
parameters
IncludeDraftRecords : true
InDocument : InDoc
InRecordList :
MasterCatalog :
ProductExtns :
ProductIds :
RecordKey :
RelationshipName [] :
VersionOption : CONFIRMED
eventState :
ImportDataDeleteRecord :

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



InDocument Parameter

Modify Parameter.

Name: InDocument
Type: Document
Eval: Variable

6. Click **OK**.

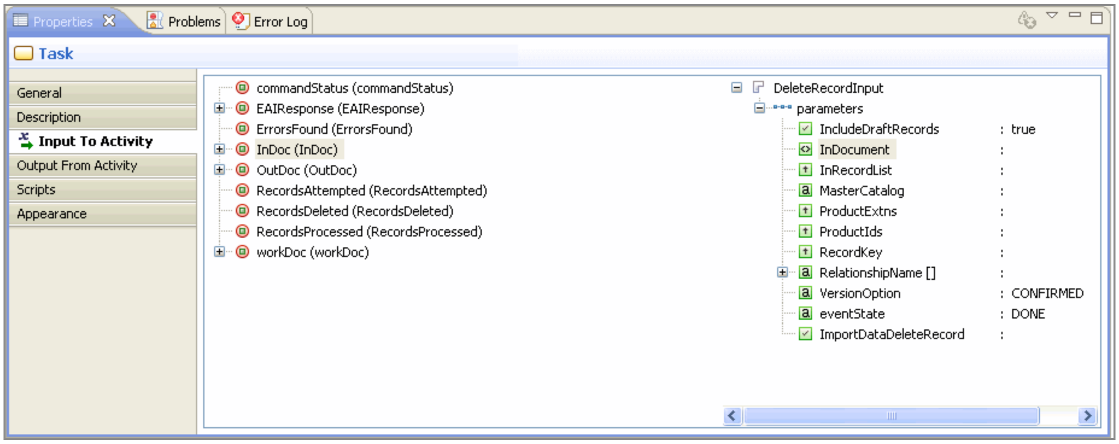
Adding Parameter for eval mode=Constant

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

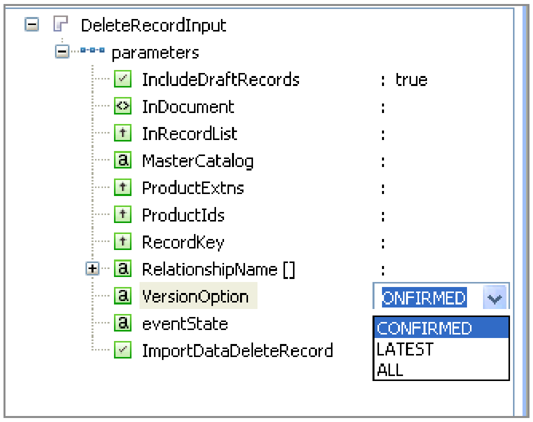
1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties** Window, go to the **Input to Activity** tab.
3. Expand the **activityname** input parameters on the right.
4. Select the parameter that you want to provide a value to and type in constant value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



```
commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)
```

```
DeleteRecordInput
  parameters
    IncludeDraftRecords      : true
    InDocument               :
    InRecordList             :
    MasterCatalog            :
    ProductExtns             :
    ProductIds               :
    RecordKey                :
    RelationshipName []      :
    VersionOption            : CONFIRMED
    eventState               : DONE
    ImportDataDeleteRecord   :
```

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

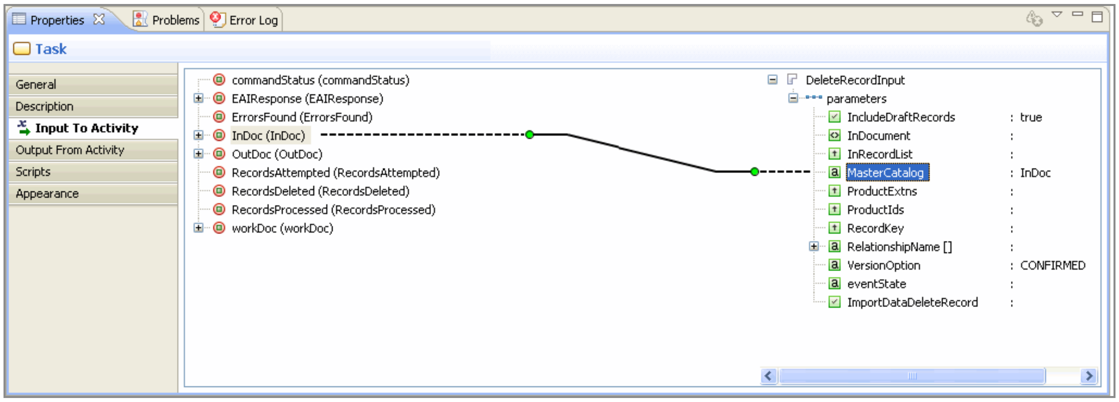


```
DeleteRecordInput
  parameters
    IncludeDraftRecords      : true
    InDocument               :
    InRecordList             :
    MasterCatalog            :
    ProductExtns             :
    ProductIds               :
    RecordKey                :
    RelationshipName []      :
    VersionOption              ONFIRMED
    eventState                 CONFIRMED
    ImportDataDeleteRecord     LATEST
                               ALL
```

Adding Parameter for eval mode=XPath

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 (mlXMLDocument type) on the left.

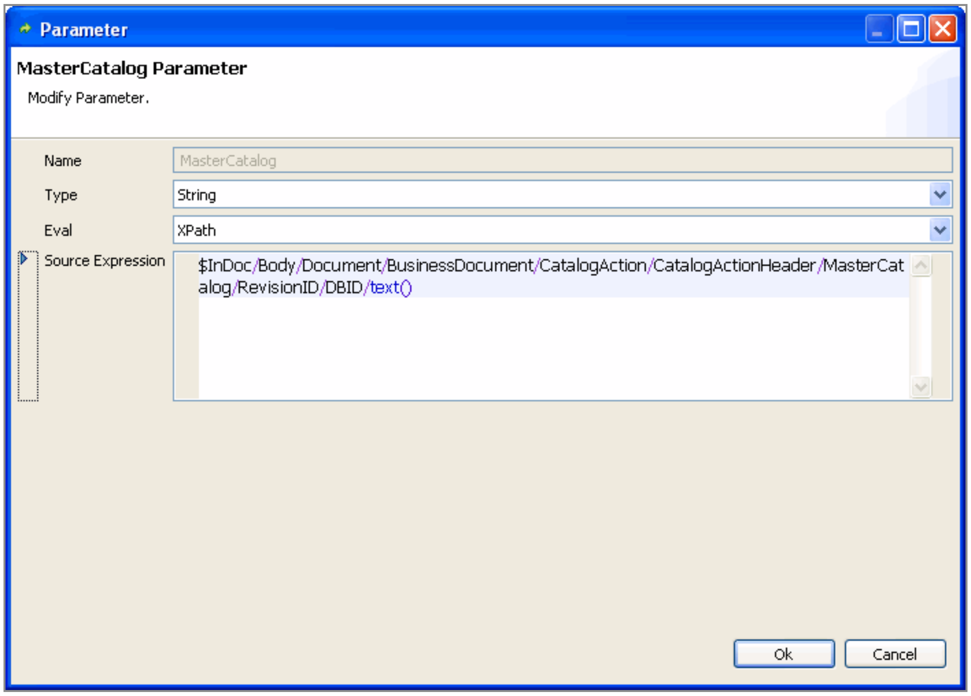


commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

DeleteRecordInput
parameters

IncludeDraftRecords	: true
InDocument	:
InRecordList	:
MasterCatalog	: InDoc
ProductExtns	:
ProductIds	:
RecordKey	:
RelationshipName []	:
VersionOption	: CONFIRMED
eventState	:
ImportDataDeleteRecord	:

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



MasterCatalog Parameter

Modify Parameter.

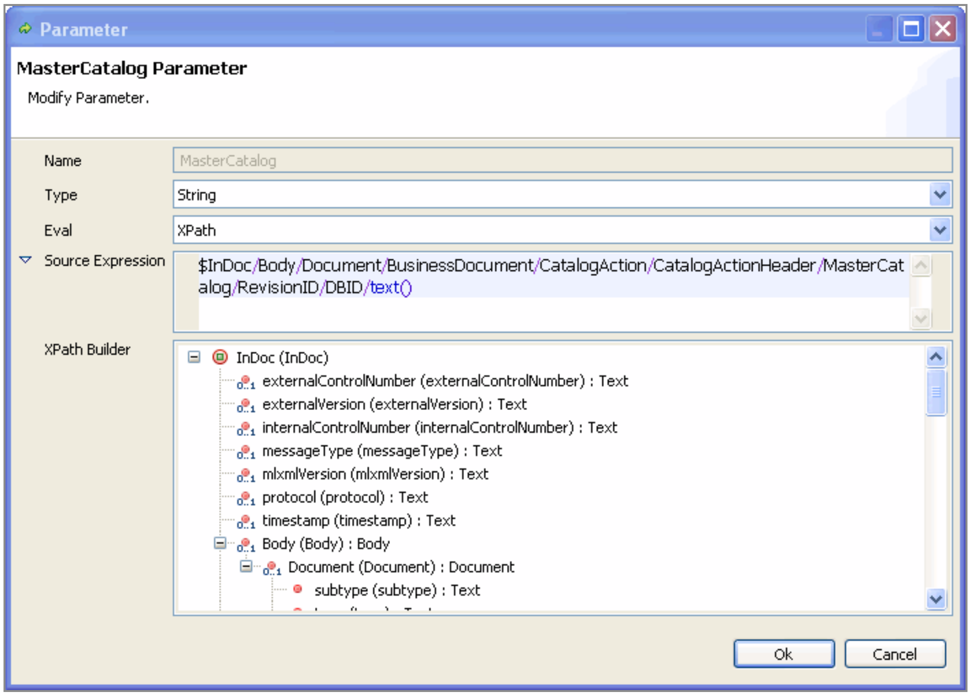
Name: MasterCatalog

Type: String

Eval: XPath

Source Expression: $InDoc/Body/Document/BusinessDocument/CatalogAction/CatalogActionHeader/MasterCatalog/RevisionID/DBID/text()

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



MasterCatalog Parameter

Modify Parameter.

Name: MasterCatalog

Type: String

Eval: XPath

Source Expression: $InDoc/Body/Document/BusinessDocument/CatalogAction/CatalogActionHeader/MasterCatalog/RevisionID/DBID/text()

XPath Builder:

```
InDoc (InDoc)
    externalControlNumber (externalControlNumber) : Text
    externalVersion (externalVersion) : Text
    internalControlNumber (internalControlNumber) : Text
    messageType (messageType) : Text
    mlxmlVersion (mlxmlVersion) : Text
    protocol (protocol) : Text
    timestamp (timestamp) : Text
    Body (Body) : Body
        Document (Document) : Document
            subtype (subtype) : Text
```

7. Click **OK**.

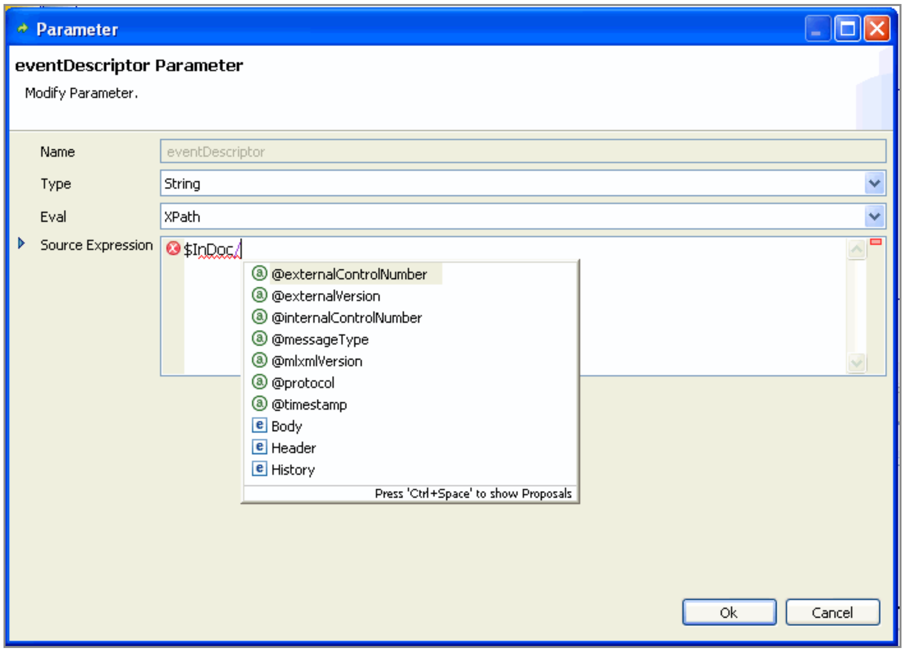
Result

Note: Use the ’$' character followed by the variable name (mapped data field) to access the mlXmlDocument.

Using Content Assist

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.



eventDescriptor Parameter

Modify Parameter.

Name: eventDescriptor

Type: String

Eval: XPath

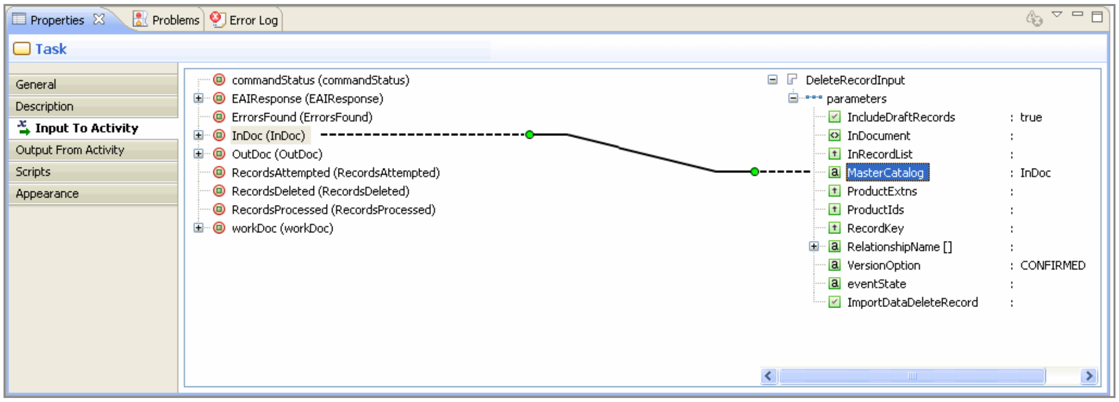
Source Expression: $InDoc/

Adding Parameter for eval mode=Rule

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.

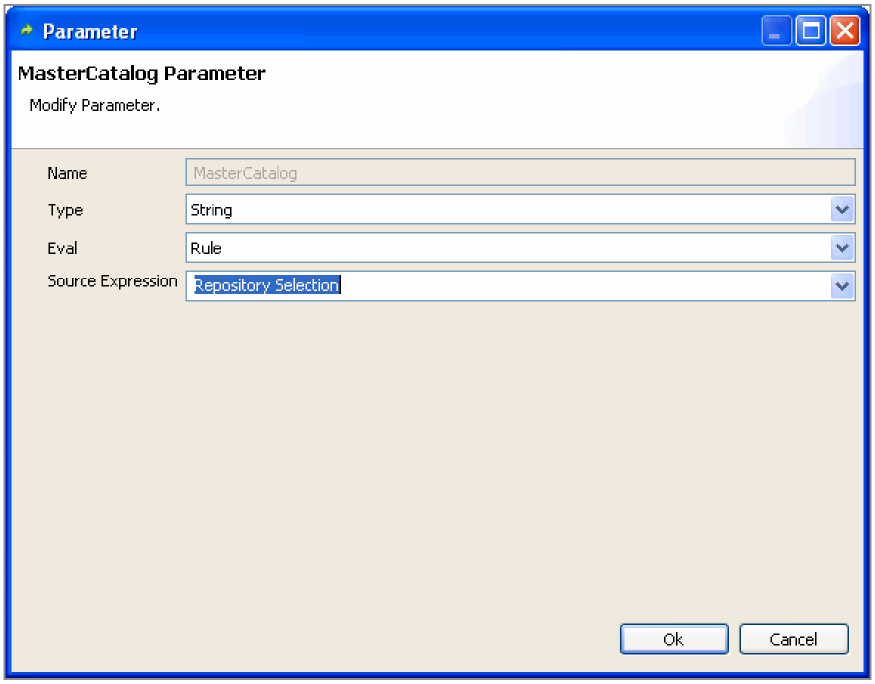
Procedure

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.


Properties
Problems
Error Log
Task
General
Description
Input To Activity
Output From Activity
Scripts
Appearance
commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)
DeleteRecordInput
parameters

IncludeDraftRecords : true
InDocument :
InRecordList :
MasterCatalog : InDoc
ProductExtns :
ProductIds :
RecordKey :
RelationshipName [] :
VersionOption : CONFIRMED
eventState :
ImportDataDeleteRecord :

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.



MasterCatalog Parameter

Modify Parameter.

Name: MasterCatalog
Type: String
Eval: Rule
Source Expression: Repository Selection

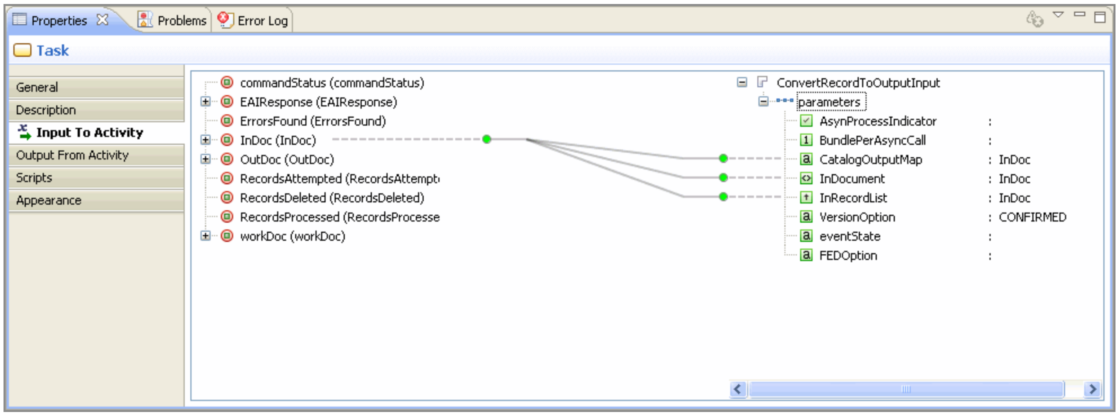
6. Click **OK**.

Adding Parameter for eval mode=Lookup

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

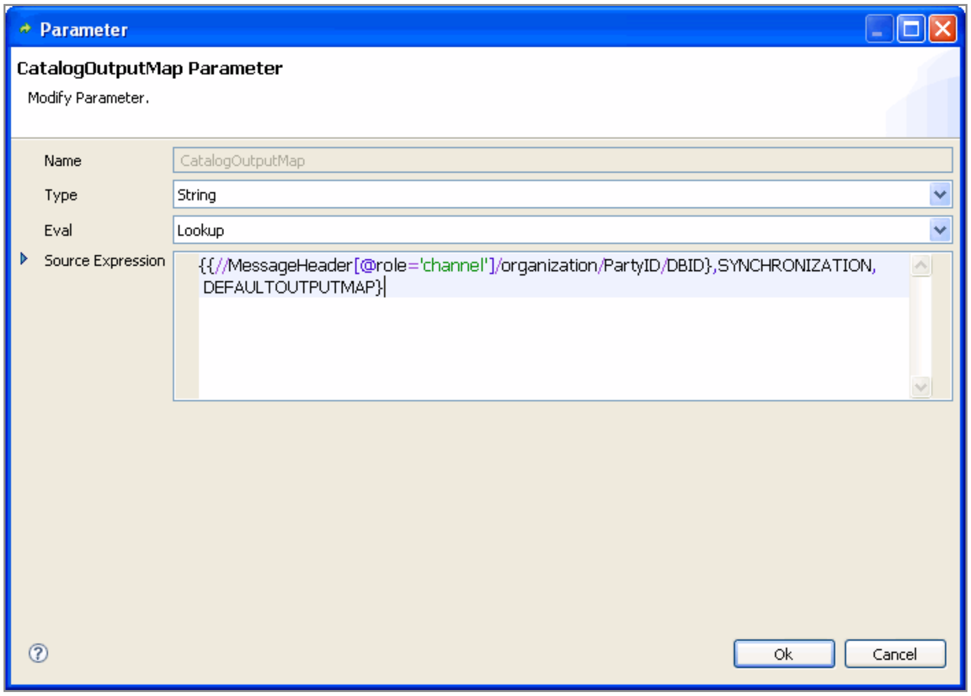
1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties** Window, go to the **Input to Activity** tab.
3. Expand the **activityname** input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField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



commandStatus (commandStatus)
EAIResponse (EAIResponse)
ErrorsFound (ErrorsFound)
InDoc (InDoc)
OutDoc (OutDoc)
RecordsAttempted (RecordsAttempt
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcesse
workDoc (workDoc)

ConvertRecordToOutputInput
parameters
AsynProcessIndicator :
BundlePerAsyncCall :
CatalogOutputMap : InDoc
InDocument : InDoc
InRecordList : InDoc
VersionOption : CONFIRMED
eventState :
FEDOption :

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



Name: CatalogOutputMap

Type: String

Eval: Lookup

Source Expression:

```
{{//MessageHeader[@role='channel']/organization/PartyID/DBID},SYNCHRONIZATION,
DEFAULTOUTPUTMAP}
```

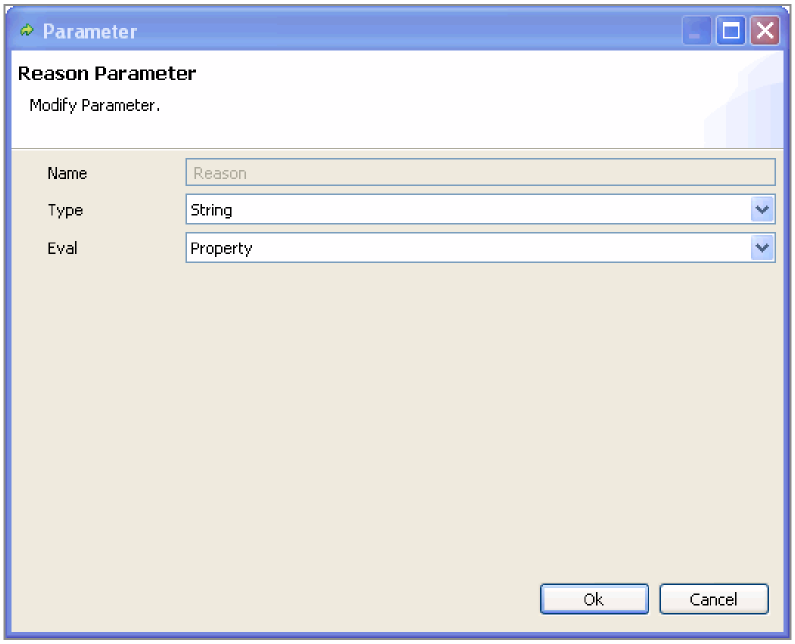
6. Click **OK**.

Adding Parameter for eval mode=Property

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

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 and select **Property** from the **Eval** drop-down list.

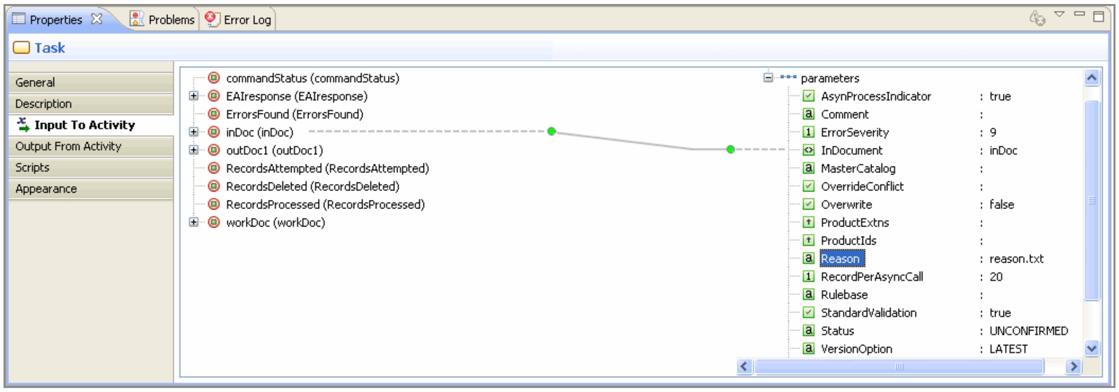


Reason Parameter

Modify Parameter.

Name	Reason
Type	String
Eval	Property

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



commandStatus (commandStatus)
EAIresponse (EAIresponse)
ErrorsFound (ErrorsFound)
inDoc (inDoc)
outDoc1 (outDoc1)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

parameters

AsynProcessIndicator	: true
Comment	:
ErrorSeverity	: 9
InDocument	: inDoc
MasterCatalog	:
OverrideConflict	:
Overwrite	: false
ProductExtns	:
ProductIds	:
Reason	: reason.txt
RecordPerAsyncCall	: 20
Rulebase	:
StandardValidation	: true
Status	: UNCONFIRMED
VersionOption	: LATEST

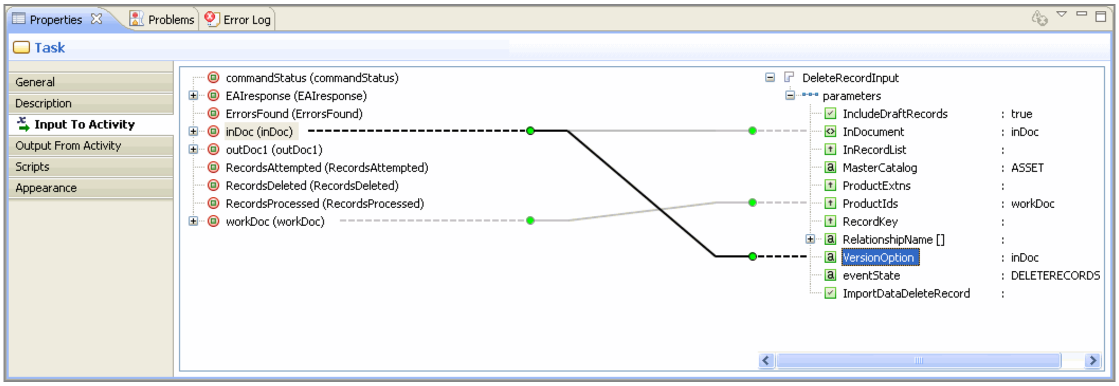
6. Click **OK**.

Adding Parameter for eval mode=Catalog

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.

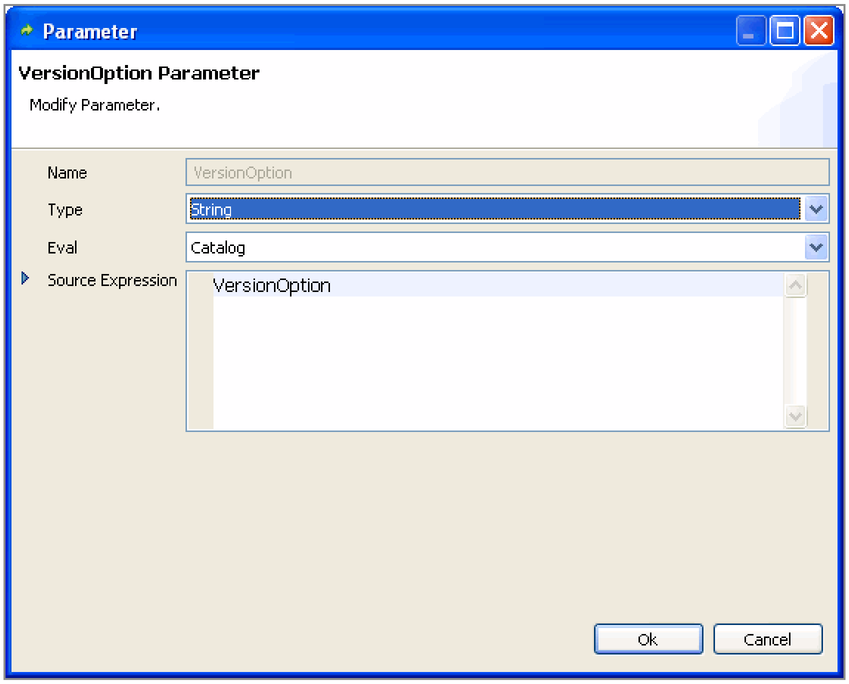


commandStatus (commandStatus)
EAIresponse (EAIresponse)
ErrorsFound (ErrorsFound)
inDoc (inDoc)
outDoc1 (outDoc1)
RecordsAttempted (RecordsAttempted)
RecordsDeleted (RecordsDeleted)
RecordsProcessed (RecordsProcessed)
workDoc (workDoc)

DeleteRecordInput
parameters

IncludeDraftRecords	: true
InDocument	: inDoc
InRecordList	:
MasterCatalog	: ASSET
ProductExtns	:
ProductIds	: workDoc
RecordKey	:
RelationshipName []	:
VersionOption	: inDoc
eventState	: DELETERECORDS
ImportDataDeleteRecord	:

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



VersionOption Parameter

Modify Parameter.

Name	VersionOption
Type	String
Eval	Catalog
Source Expression	VersionOption

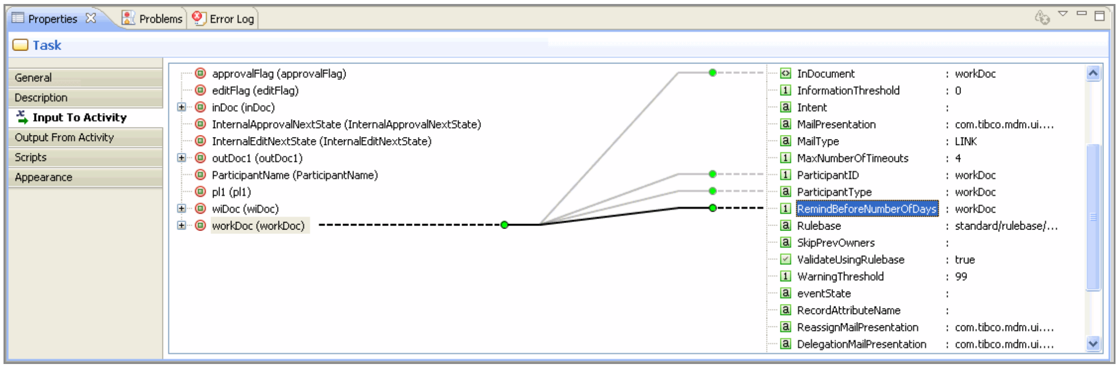
6. Enter the source expression in the source expression editor and click **OK**.

Adding Parameter for eval mode=Compute

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

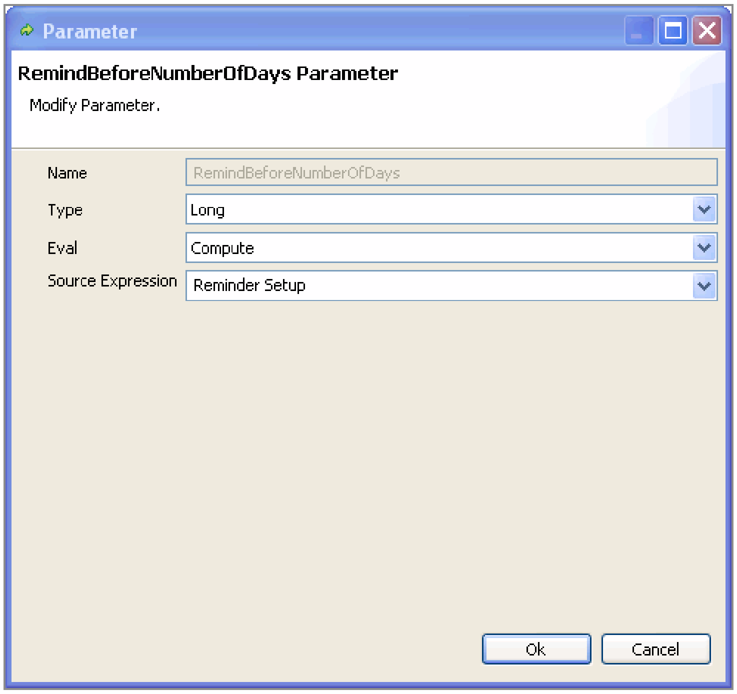
1. Select the activity (that you want to define parameters for) in the process flow.
2. In the **Properties** Window, go to the **Input to Activity** tab.
3. Expand the **activityname** input parameters on the right.
4. Select the parameter that you want to provide a value to and drag it onto DataField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



approvalFlag (approvalFlag)
editFlag (editFlag)
inDoc (inDoc)
InternalApprovalNextState (InternalApprovalNextState)
InternalEditNextState (InternalEditNextState)
outDoc1 (outDoc1)
ParticipantName (ParticipantName)
pl1 (pl1)
wiDoc (wiDoc)
workDoc (workDoc)

InDocument	: workDoc
InformationThreshold	: 0
Intent	:
MailPresentation	: com.tibco.mdm.ui....
MailType	: LINK
MaxNumberOfTimeouts	: 4
ParticipantID	: workDoc
ParticipantType	: workDoc
RemindBeforeNumberOfDays	: workDoc
Rulebase	: standard/rulebase/...
SkipPrevOwners	:
ValidateUsingRulebase	: true
WarningThreshold	: 99
eventState	:
RecordAttributeName	:
ReassignMailPresentation	: com.tibco.mdm.ui....
DelegationMailPresentation	: com.tibco.mdm.ui....

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



Parameter

RemindBeforeNumberOfDays Parameter

Modify Parameter.

Name	RemindBeforeNumberOfDays
Type	Long
Eval	Compute
Source Expression	Reminder Setup

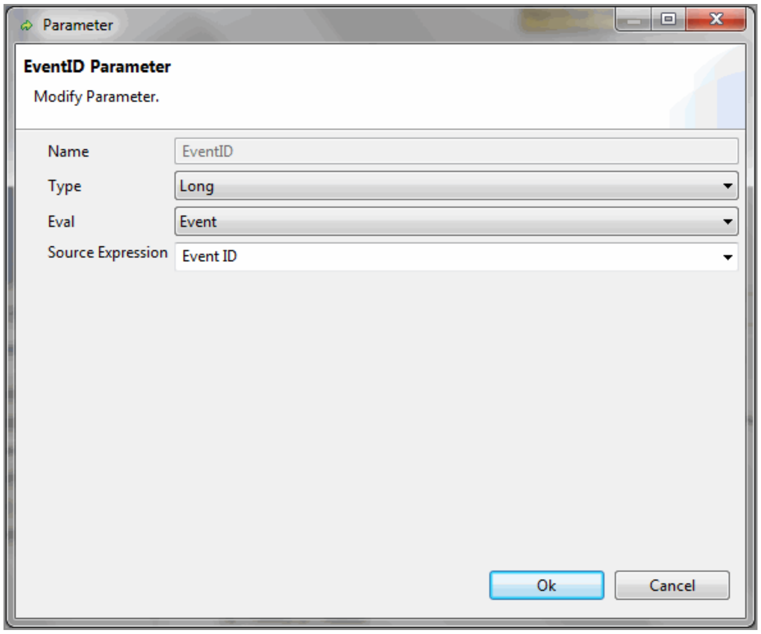
6. Click **OK**.

Adding Parameter for eval mode=Event

An 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.



Parameter

EventID Parameter

Modify Parameter.

Name: EventID

Type: Long

Eval: Event

Source Expression: Event ID

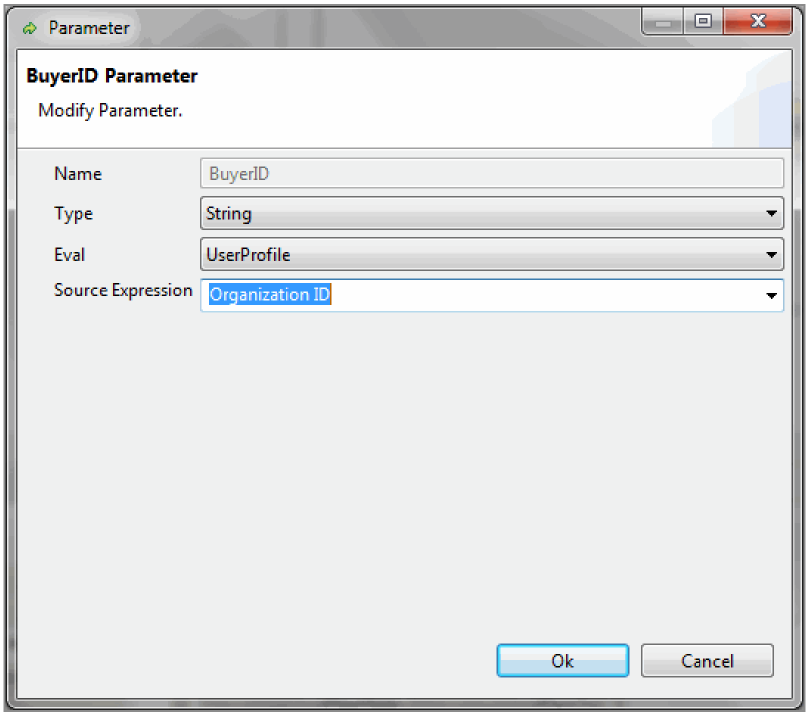
6. Click **OK**.

Adding Parameter for eval mode=UserProfile

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.



BuyerID Parameter

Modify Parameter.

Name: BuyerID

Type: String

Eval: UserProfile

Source Expression: Organization ID

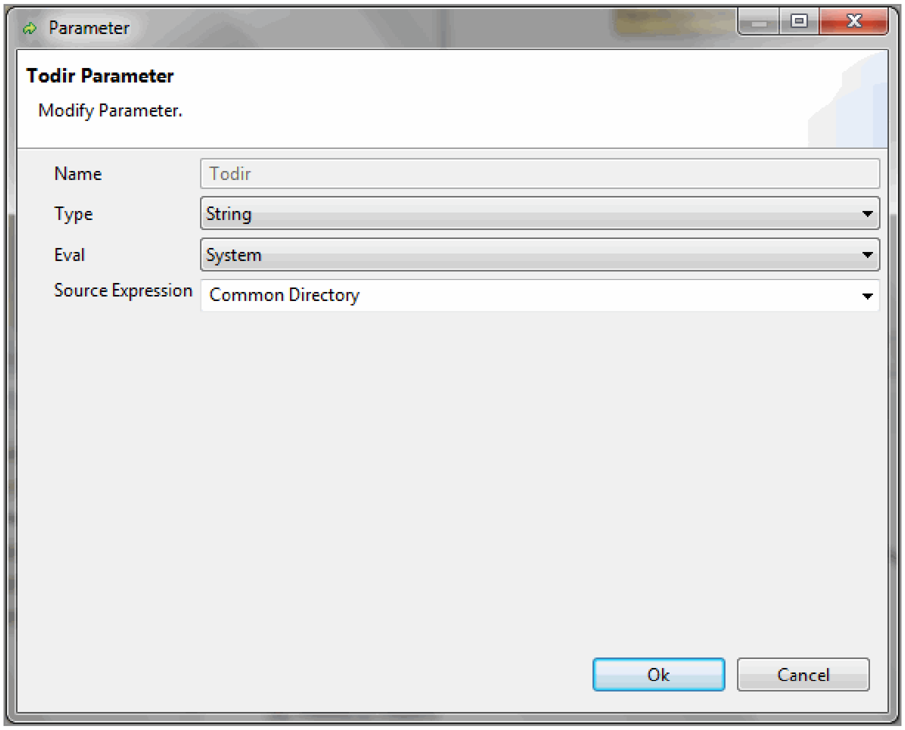
6. Click **OK**.

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.



Todir Parameter

Modify Parameter.

Name: Todir

Type: String

Eval: System

Source Expression: Common Directory

6. Click **OK**.

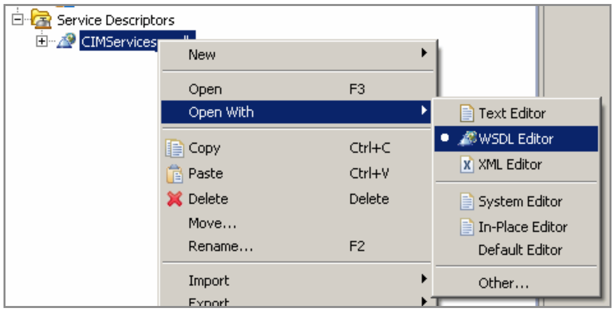
Adding and Modifying Custom Parameters

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).

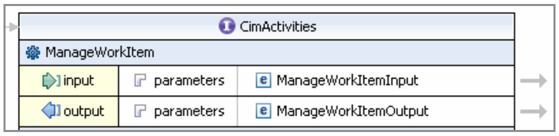
To add or modify a parameter, use the WSDL editor supplied with the TIBCO MDM Process Designer.

Procedure

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

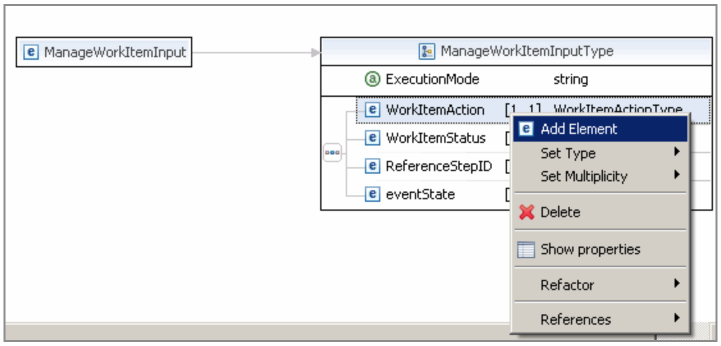


2. Go to a MDM Activity and click the arrow symbol next to <ActivityName>Input.



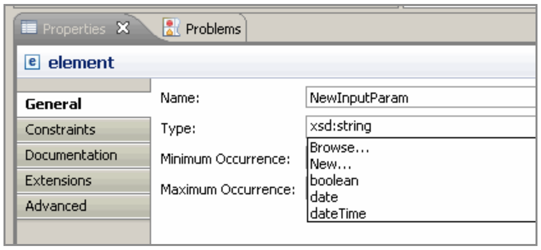
CimActivities		
ManageWorkItem		
input	parameters	ManageWorkItemInput
output	parameters	ManageWorkItemOutput

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.

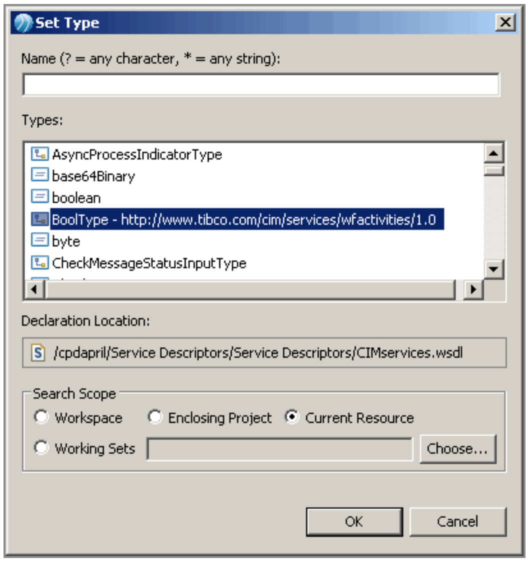


Name: NewInputParam

Type: xsd:string

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



Set Type

Name (? = any character, * = any string):

Types:

- AsyncProcessIndicatorType
- base64Binary
- boolean
- BoolType - http://www.tibco.com/cim/services/wfactivities/1.0
- byte
- CheckMessageStatusInputType

Declaration Location:

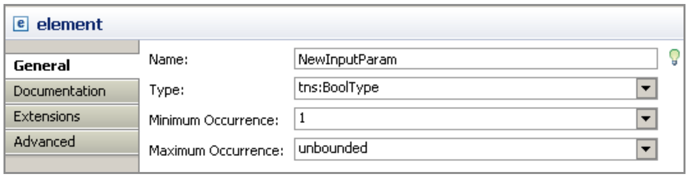
/cpdapril/Service Descriptors/Service Descriptors/CIMservices.wsdl

Search Scope

○ Workspace ○ Enclosing Project ● Current Resource

○ Working Sets

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.



element

Name: NewInputParam

Type: tns:BoolType

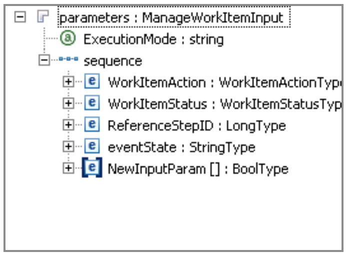
Minimum Occurrence: 1

Maximum Occurrence: unbounded

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.



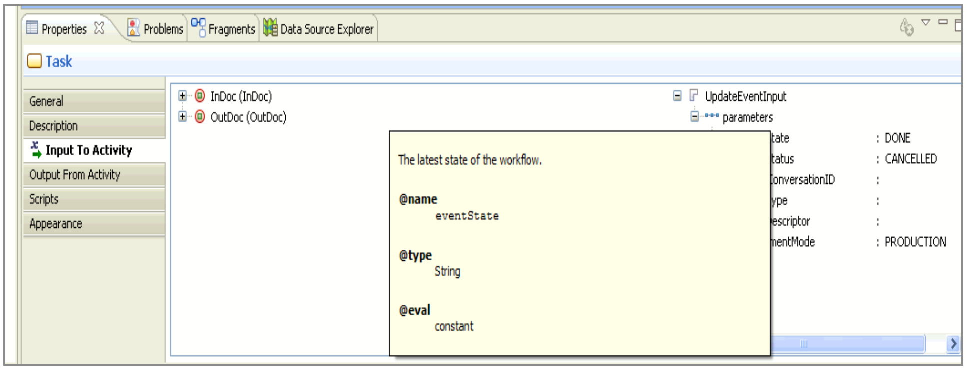
parameters : ManageWorkItemInput
ExecutionMode : string
sequence
WorkItemAction : WorkItemActionTyp
WorkItemStatus : WorkItemStatusTyp
ReferenceStepID : LongType
eventState : StringType
NewInputParam [] : BoolType

Tooltip Parameter Descriptions

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

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_<version>.jar\com\tibco\xpd\cim\properties\tooltip\parameterdescriptions.properties file`.



The latest state of the workflow.

@name
eventState

@type
String

@eval
constant

tate : DONE
tatus : CANCELLED
onversationID :
ype :
escriptor :
nentMode : PRODUCTION

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.

> **Note:** 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.

> **Note:** 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 parameter description

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_<version>.jar` with a zip file tool such as WinZip.

3. Take backup of `com.tibco.xpd.cim.properties_<version>.jar` file.

4. Extract the `com.tibco.xpd.cim.properties_<version>`

`.jar\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_<version>.jar`. This overwrites the earlier properties jar file.
7. Restart TIBCO MDM Studio and view the newly added parameter description.

Transitions

Explanation of transitions in TIBCO MDM Process Designer.

Transition Types

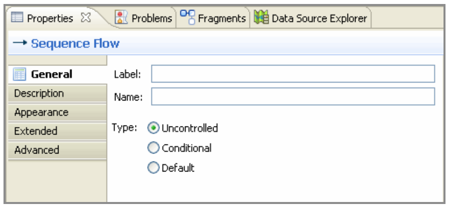
Local transitions can be either simple or conditional.

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.

Simple Transitions

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**.



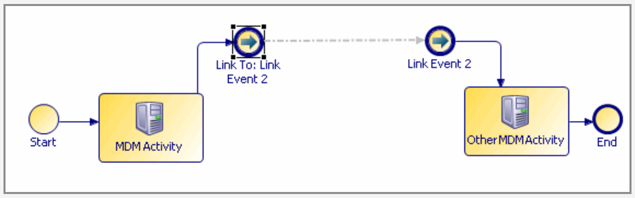
Label:

Name:

Type: ● Uncontrolled
○ Conditional
○ Default

✓ **Tip:** 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

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

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.

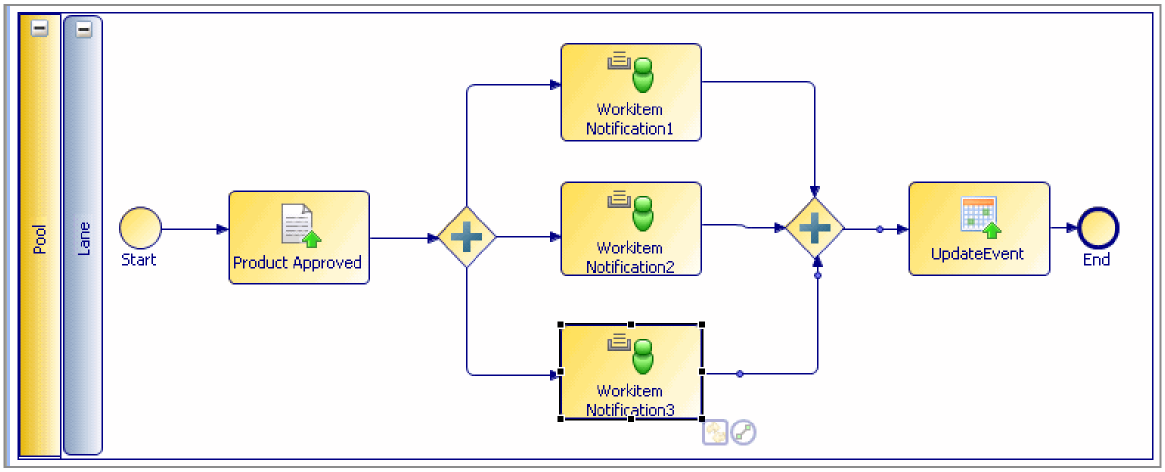
For example, consider the case of a workflow which functions as follows:

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.

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

Parallel Transitions



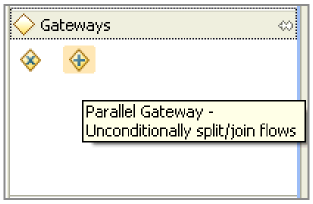
Creating a Parallel Transition

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:

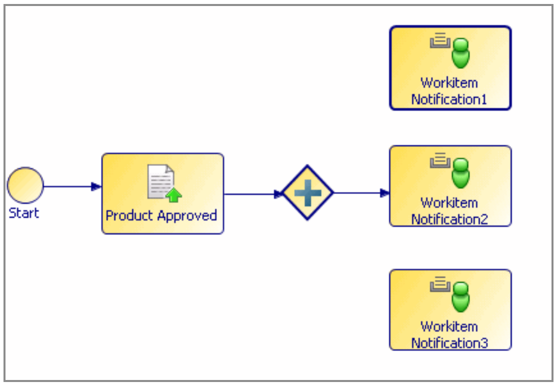
Procedure

1. Create an activity **ProductApproved** and create three workitem notification activities.

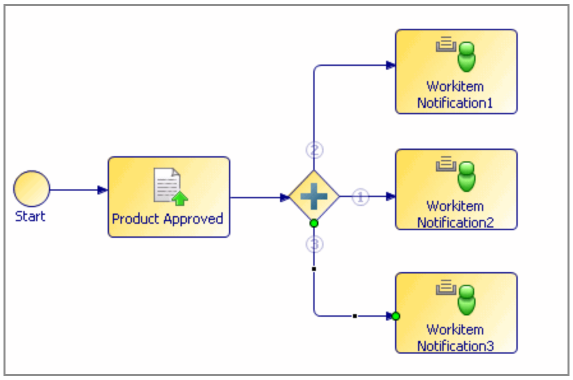
2. Next, select **Parallel Gateway: Unconditionally split/join flows** in the Palette.



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



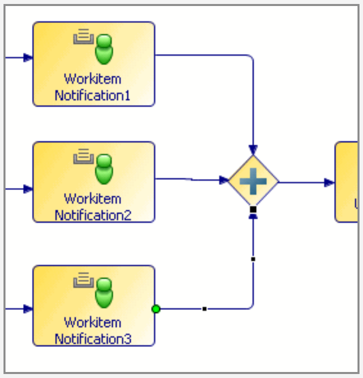
3. Use the sequence flow to link the gateway with the other **Workflow Notification** tasks.



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

Note: 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

Procedure

1. TIBCO 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, TIBCO MDM Process Designer does not currently support group and group limit attributes for parallel transition.

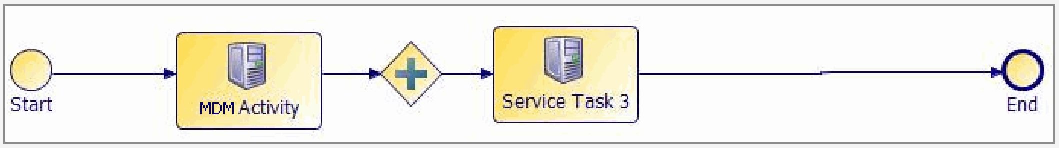
2. Every parallel context has to join via a parallel gateway. A process cannot end while the process is split into multiple contexts. Before the process ends, all contexts have to join together.

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.

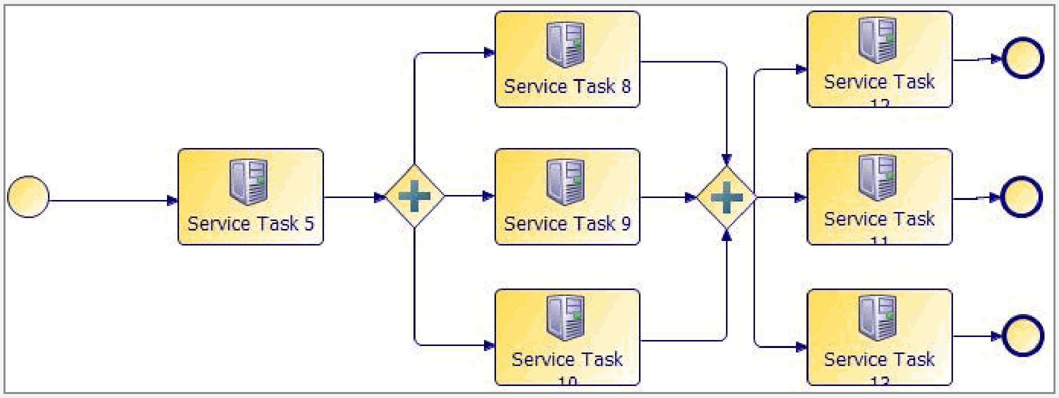
Result

Note: 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.

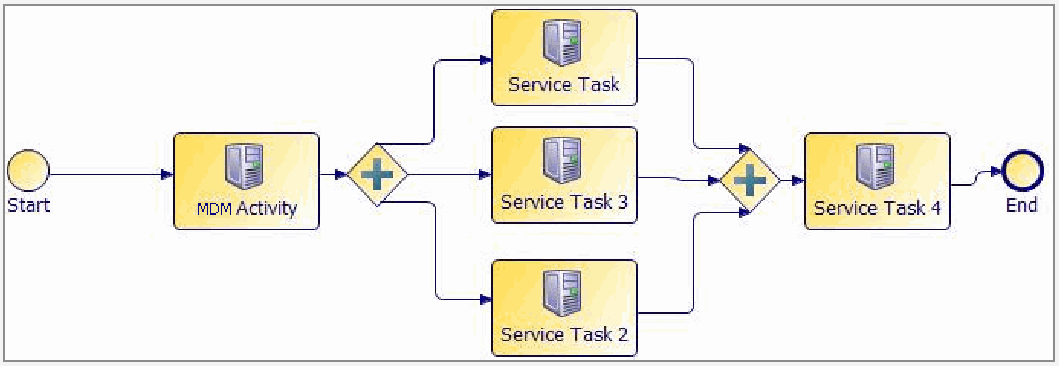
Not Supported: A Split connecting to a single activity.



Not Supported: A Join connecting to a multiple activities.



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



Transition Conditions

You can create conditions for simple transitions.

Conditional Transitions

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.

Note: Text area (free text) based conditions get embedded into the workflow file and can be quickly turned around since the TIBCO 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.

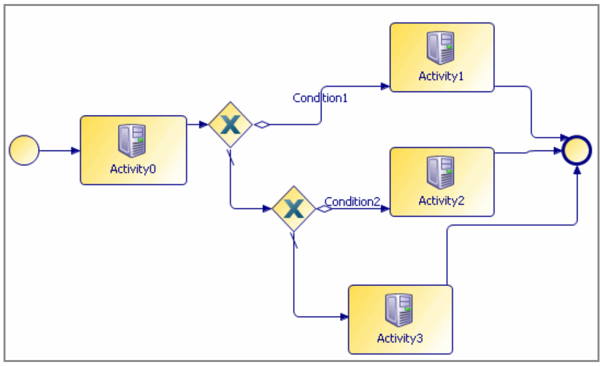
Interpreted Transitions

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

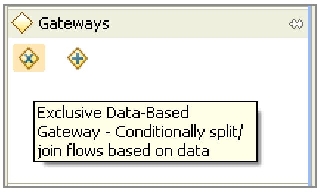
To create a conditional transition, you need to use **Gateway: Conditionally split/join flows based on data** to link one activity to two or more activities. (You can also define multiple conditions for the same activity by using multiple Gateways).

To define a simple conditional transition:

Procedure

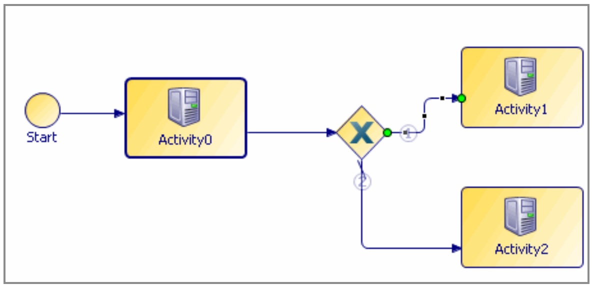
1. **Create the activities**: Create two activities Activity1 and Activity2, which Activity0 can optionally move to depending on the condition fulfilled.

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.

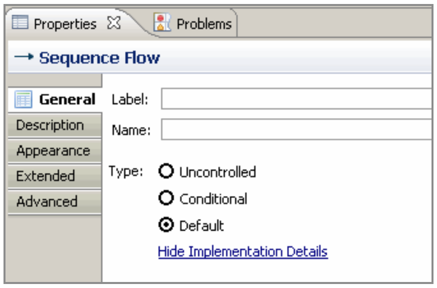
   

   Exclusive Data-Based Gateway - Conditionally split/join flows based on data

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.



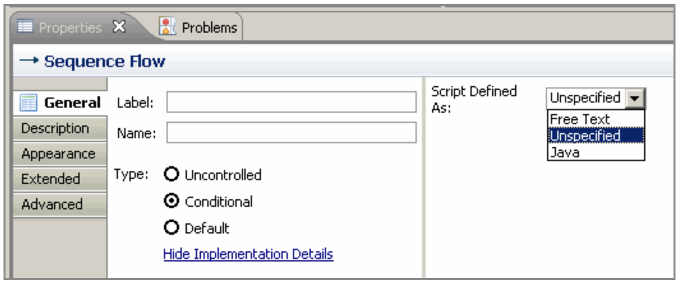
Sequence Flow

Label:

Name:

Type: ○ Uncontrolled ○ Conditional ● Default

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.



Sequence Flow

Label:

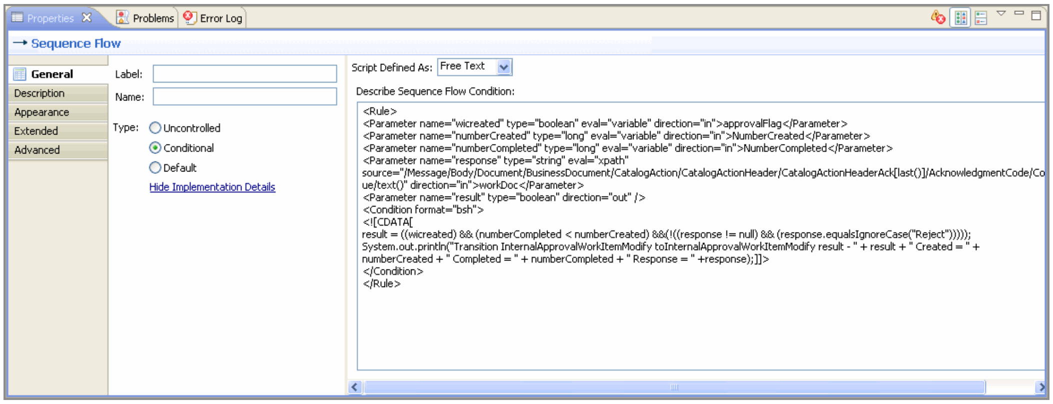
Name:

Type: ○ Uncontrolled ● Conditional ○ Default

Script Defined As: Unspecified

> **Note:** 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.



Label:

Name:

Type: ○ Uncontrolled ● Conditional ○ Default

Script Defined As: Free Text

Describe Sequence Flow Condition:

```
<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/CatalogActionHeader/CatalogActionHeaderAck[last()]/AcknowledgmentCode/Co
ue/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 toInternalApprovalWorkItemModify result - " + result + " Created = " +
numberCreated + " Completed = " + numberCompleted + " Response = " +response);]]>
</Condition>
</Rule>
```

Result

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

> **Note:** 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

There are two examples of a conditional transition.

Example 1

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"

      <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/Cata
logA      ctionHeader/CatalogActionHeaderAck[last
()]/AcknowledgmentCode/Code/Val      ue/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

This conditional transition example does the following:

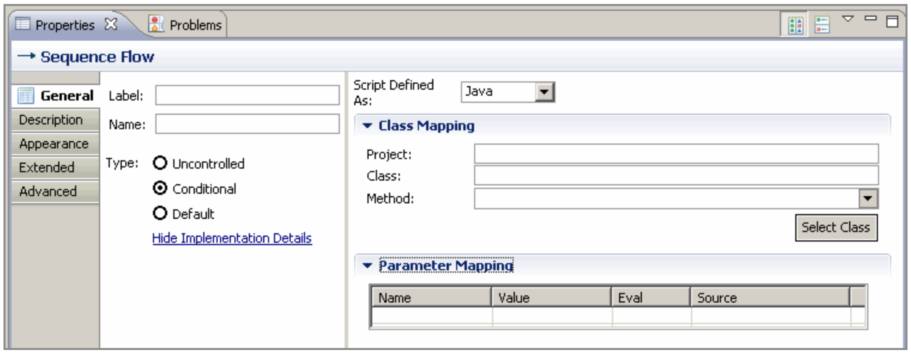
- If a ChangeProduct action is taken on a record, it is sent for edit approval (to the users authorized to approve record modifications).
- If a DeleteRecord action is taken on a record, it is sent for delete approval.

```
<Rule>
      <Parameter name="action" type="string" eval="xpath"
      source="/Message/Body/Document/BusinessDocument/CatalogAction
/CatalogA      ctionDetails/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
/CatalogA      ctionDetails/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

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.



Type: ○ Uncontrolled ● Conditional ○ Default

Script Defined As: Java

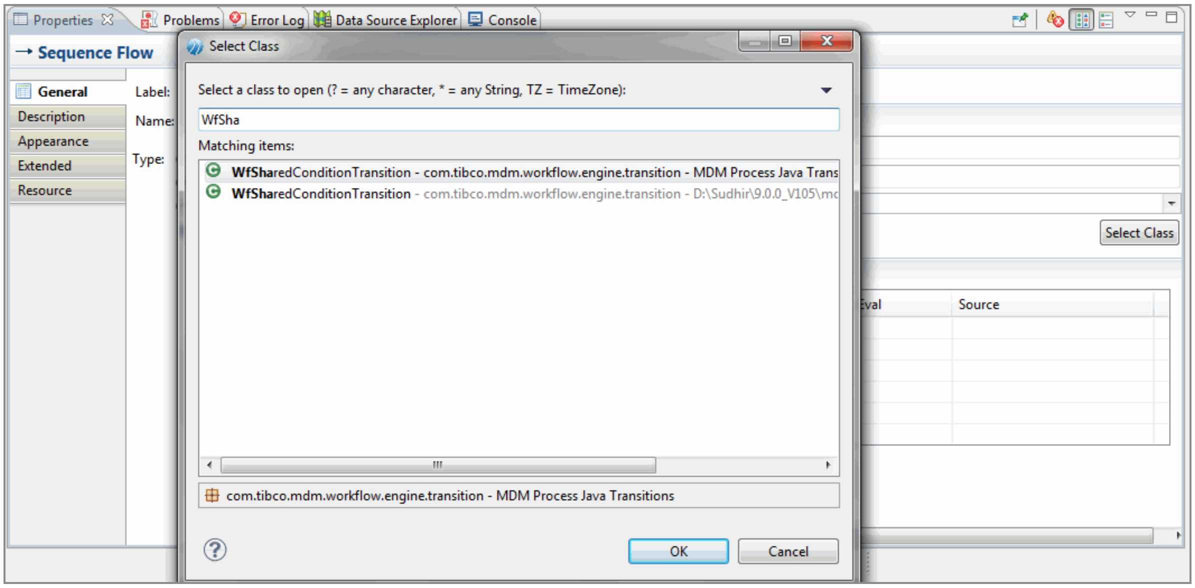
Name	Value	Eval	Source

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.

Note: 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.



Select Class

Select a class to open (? = any character, * = any String, TZ = TimeZone):

WfSha

Matching items:

WfSharedConditionTransition - com.tibco.mdm.workflow.engine.transition - MDM Process Java Trans

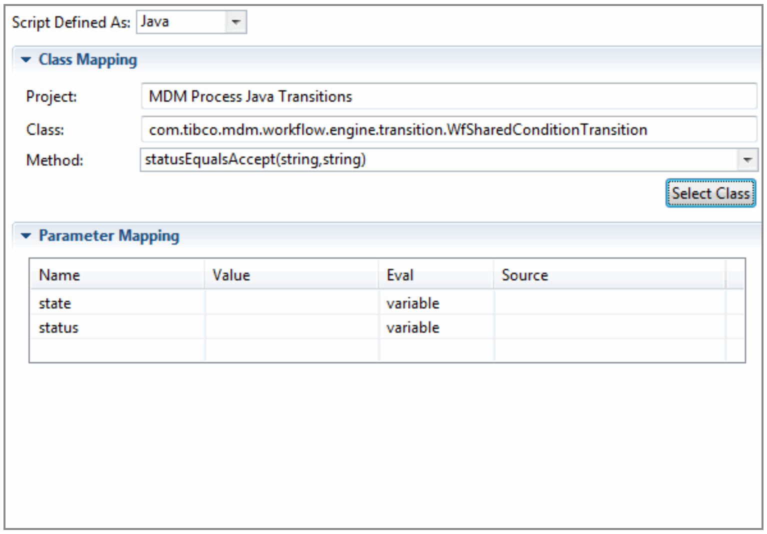
WfSharedConditionTransition - com.tibco.mdm.workflow.engine.transition - D:\Sudhir\9.0.0_V105\mc

com.tibco.mdm.workflow.engine.transition - MDM Process Java Transitions

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

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



Script Defined As: Java

Class Mapping

Project: MDM Process Java Transitions

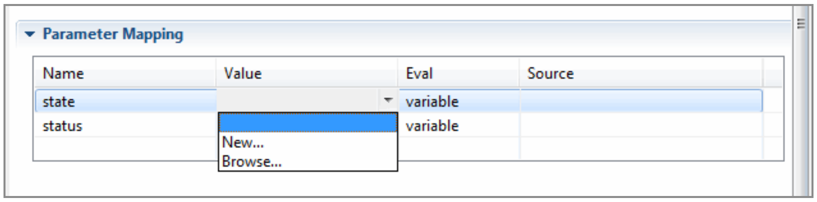
Class: com.tibco.mdm.workflow.engine.transition.WfSharedConditionTransition

Method: statusEqualsAccept(string,string)

Parameter Mapping

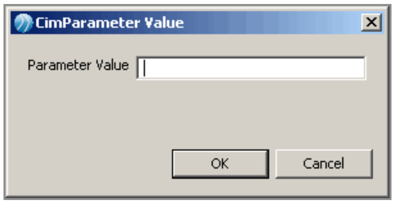
Name	Value	Eval	Source
state		variable	
status		variable	

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

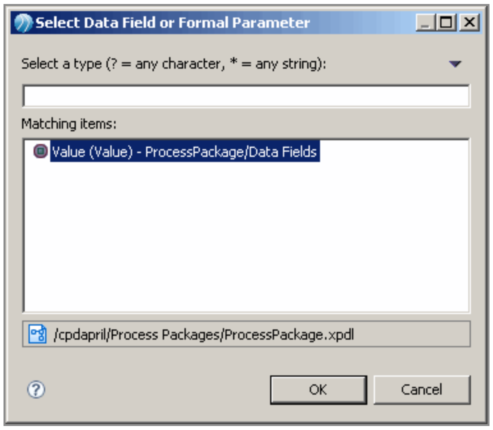


Name	Value	Eval	Source
state		variable	
status		variable	

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.


Select Data Field or Formal Parameter
Select a type (? = any character, * = any string):
Matching items:
OK
Cancel

Value (Value) - ProcessPackage/Data Fields

/cpdapril/Process Packages/ProcessPackage.xpdl

Creating a Compiled Transition

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

Procedure

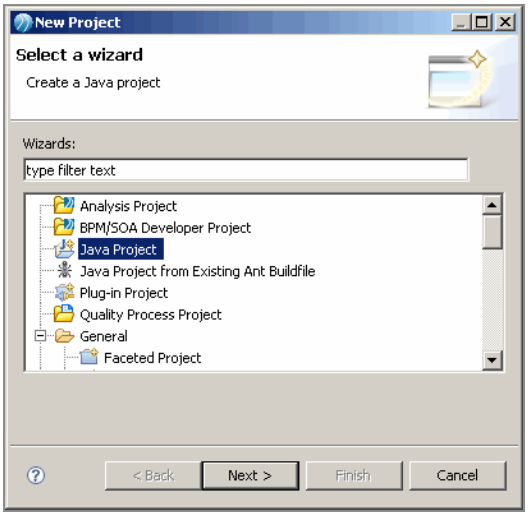
1. Creating a Java Transition Project
2. Using the Condition in the process
3. Compiling and packaging the Java Project
4. Merging Jar With eml.war

Creating a Java Transition Project

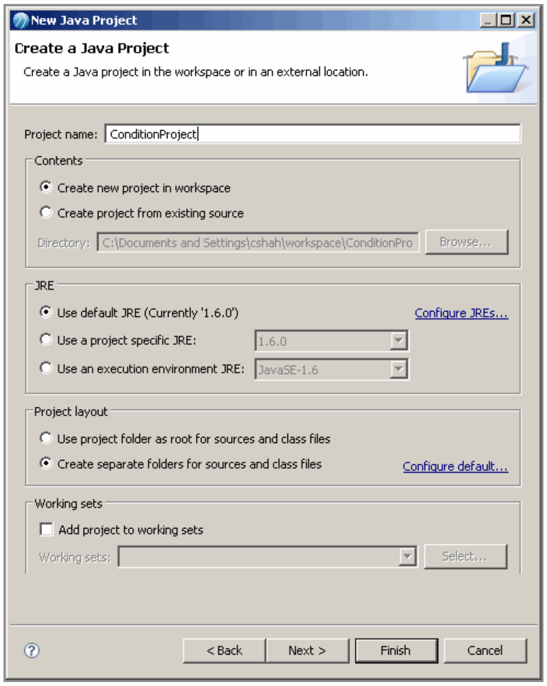
Create a new Java Project for the transition condition.

Procedure

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



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



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: ConditionProject

Contents

● Create new project in workspace

○ Create project from existing source

Directory: C:\Documents and Settings\cshah\workspace\ConditionPro Browse...

JRE

● Use default JRE (Currently '1.6.0') Configure JREs...

○ Use a project specific JRE: 1.6.0

○ Use an execution environment JRE: JavaSE-1.6

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files Configure default...

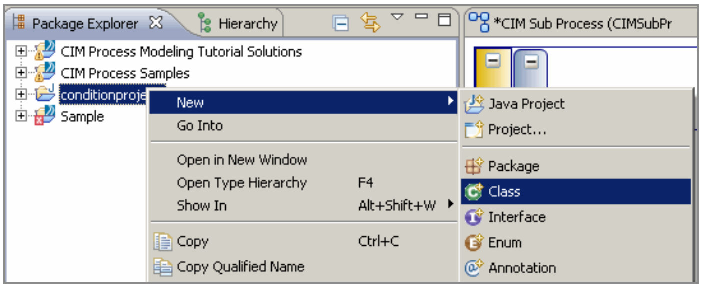
Working sets

□ Add project to working sets

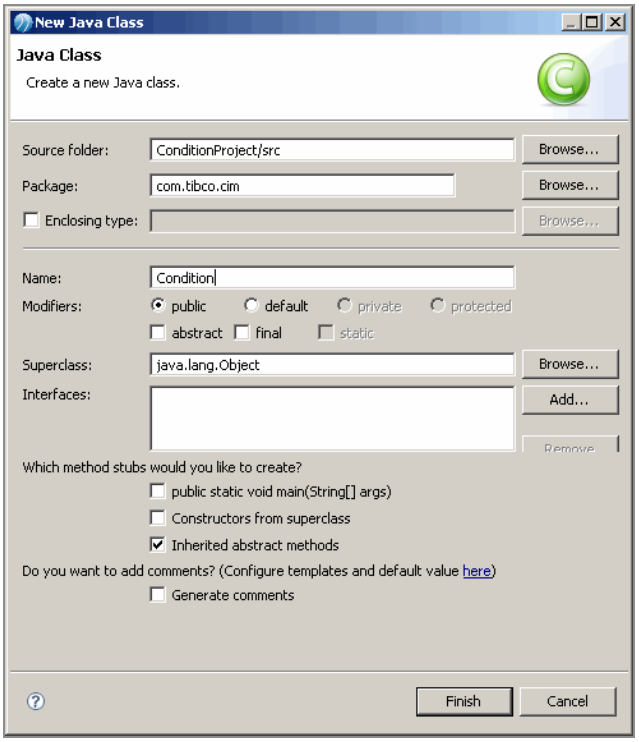
Working sets: Select...

< Back Next > Finish Cancel

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**).



New Java Class

Java Class

Create a new Java class.

Source folder: ConditionProject/src

Package: com.tibco.cim

□ Enclosing type:

Name: Condition

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

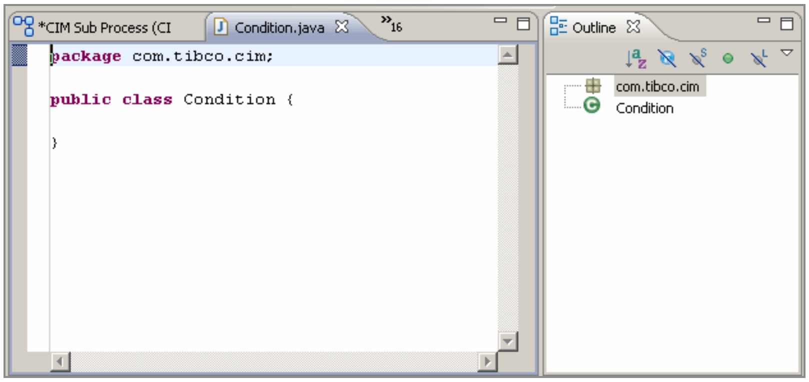
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

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



```
package com.tibco.cim;

public class Condition {

}
```

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

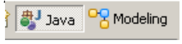
```
package com.tibco.cim;
public class Condition {
 static public boolean evaluateCondition(String a, String b){
      return true;
```

```
  }
static public boolean evalRule(String subdoctype,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;
  }
}
```

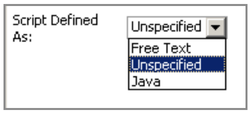
Using the Condition in the process

Procedure

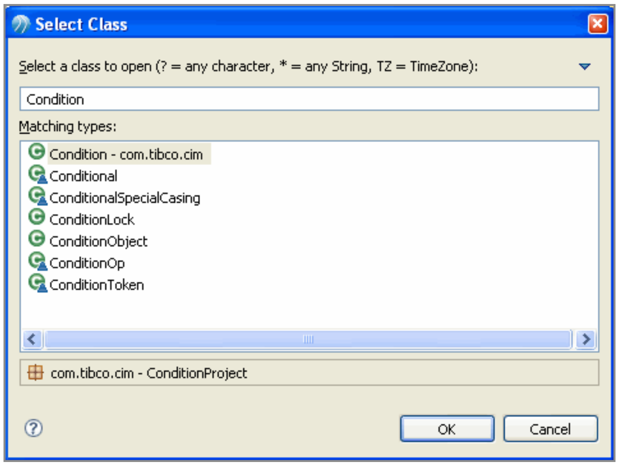
1. Switch to the Modeling view and select the transition to which you want to apply the java condition.



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



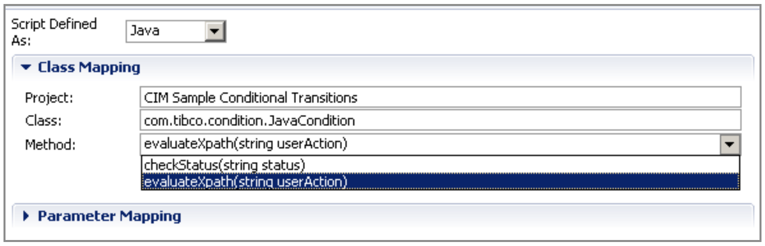
3. Click the **Select Class** button and enter the name (or the initial characters) of the previously defined condition.



Select a class to open (? = any character, * = any String, TZ = TimeZone):

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.



Script Defined As: Java

Project: CIM Sample Conditional Transitions

Class: com.tibco.condition.JavaCondition

Method: evaluateXpath(string userAction)

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

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 `eml.war` file of the core TIBCO MDM server.

Merging Jar With eml.war

You can merge your exported condition jar into eml.war by using Windows and UNIX.

Merging Jar with eml.war by using Windows

On Windows, to merge an external jar file with `eml.war\WEB-INF\lib\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 `com\tibco\com\tibco\mdm\workflow\engine\transition\*.class`.

Merging Jar With eml.war by using UNIX

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 eml.war file). $jar -cvf <new jar file name> classes

   Follow these steps to merge the new `condition.jar` with eml.war:
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 eml.war.
10. Restart the MDM Server. You can now use the new conditional transition in your workflow.

Result

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

Validation

Explanation of process Validation.

Validating

The MDM validation plug-in checks that changes are made in accordance with the validation rules.

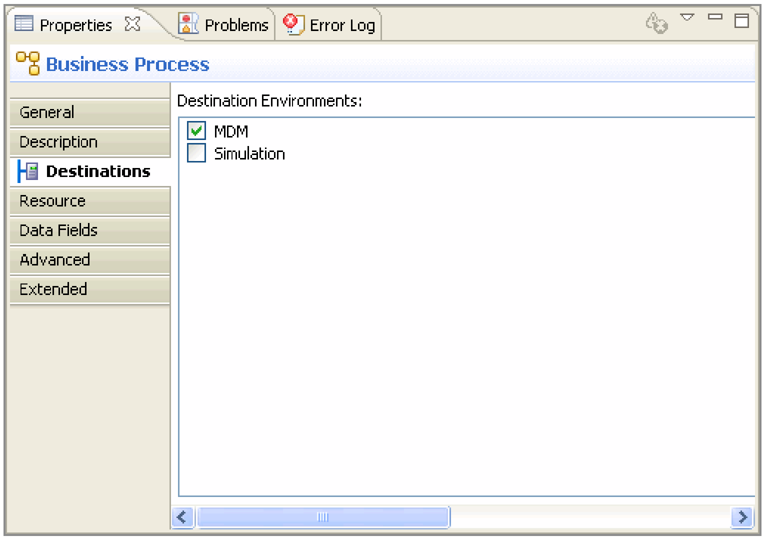
It runs in the background as you modify or edit the process.

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



Destination Environments:

☑ MDM
□ Simulation

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

Note: Validation will automatically occur if you import an existing TIBCO MDM process into the TIBCO 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 TIBCO MDM template.

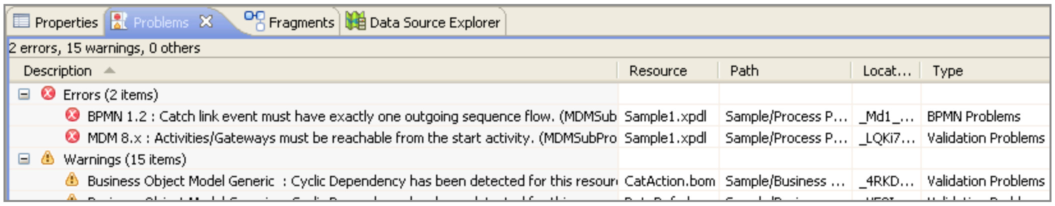
However, if you choose to create a process from scratch without using an TIBCO MDM template, you will need to set the process destination to MDM for validation to take place.

Problems Tab

The **Problems** tab displays errors related to specific elements in your process.

Information is categorized into Errors and Warnings.

Problems Tab



2 errors, 15 warnings, 0 others

Description	Resource	Path	Locat...	Type
⊟ Errors (2 items)				
BPMN 1.2 : Catch link event must have exactly one outgoing sequence flow. (MDMSub	Sample1.xpdl	Sample/Process P...	_Md1_...	BPMN Problems
MDM 8.x : Activities/Gateways must be reachable from the start activity. (MDMSubPro	Sample1.xpdl	Sample/Process P...	_LQKi7...	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 XPDL.

Validation Checks

Validation checks are performed to ensure conformity to TIBCO MDM standards.

Miscellaneous Validations

- 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

- 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

- 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

- 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 is 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

- 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

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

Data Field Restrictions

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

> **Note:** Business Studio elements that are not supported by TIBCO 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

You can import processes into the TIBCO MDM Process Designer, export processes created in TIBCO MDM Process Designer to TIBCO MDM, and deploy processes.

Import Processes into TIBCO MDM Process Designer

In addition to enabling you to graphically define processes, the TIBCO MDM Process Designer also supports import of existing TIBCO 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

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.

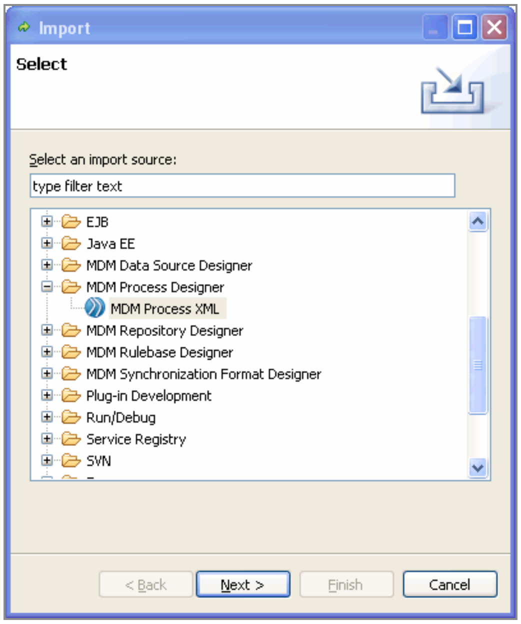
    d. Select **Business Processes, Business Objects, and Services** in the Asset type Selection dialog.

    e. Follow the rest of the wizard to complete the Process creation without using the MDM Template.

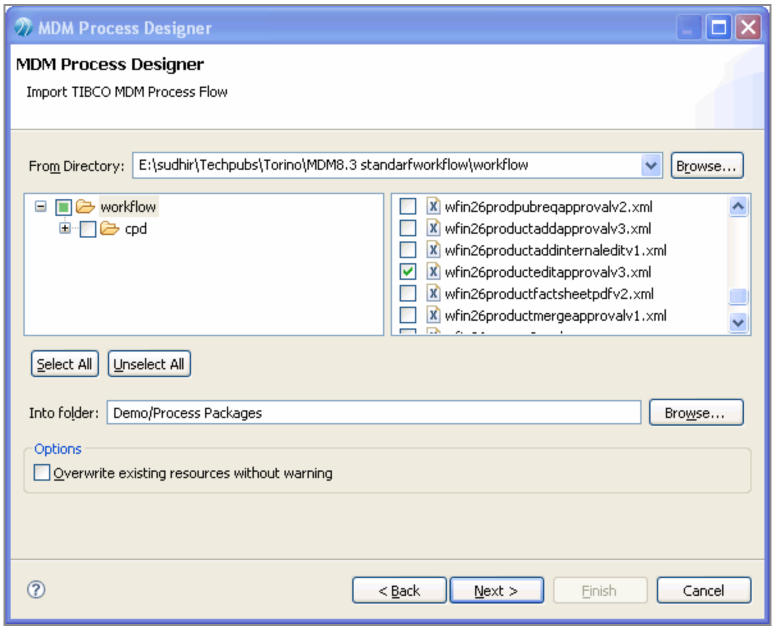
2. Use the **Import Wizard** to import an existing workflow:

    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**.



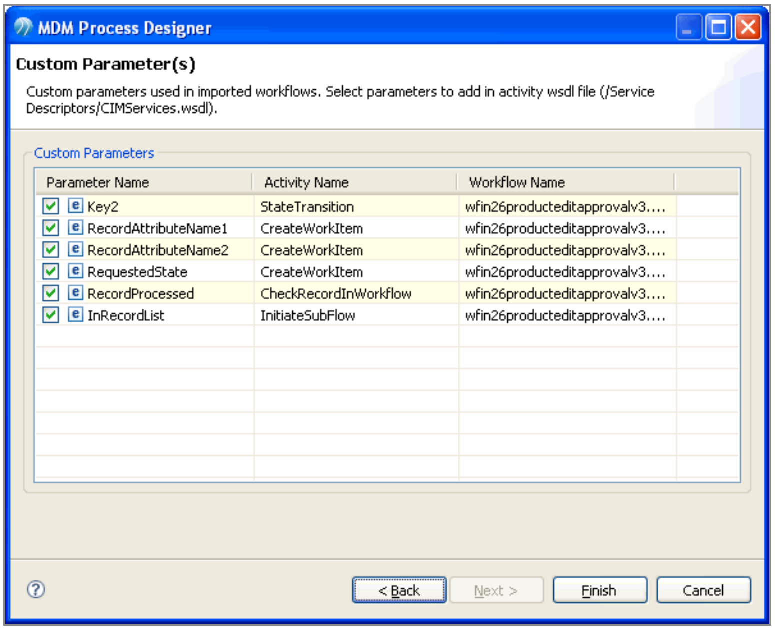
From Directory: E:\sudhir\Techpubs\Torino\MDM8.3 standarfworkflow\workflow

□ wfin26prodpubreqapprovalv2.xml
□ wfin26productaddapprovalv3.xml
□ wfin26productaddinternaleditv1.xml
☑ wfin26producteditapprovalv3.xml
□ wfin26productfactsheetpdfv2.xml
□ wfin26productmergeapprovalv1.xml

Into folder: Demo/Process Packages

□ Overwrite existing resources without warning

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



Custom Parameter(s)

Custom parameters used in imported workflows. Select parameters to add in activity wsdl file (/Service Descriptors/CIMServices.wsdl).

Custom Parameters

	Parameter Name	Activity Name	Workflow Name
☑	Key2	StateTransition	wfin26producteditapprovalv3....
☑	RecordAttributeName1	CreateWorkItem	wfin26producteditapprovalv3....
☑	RecordAttributeName2	CreateWorkItem	wfin26producteditapprovalv3....
☑	RequestedState	CreateWorkItem	wfin26producteditapprovalv3....
☑	RecordProcessed	CheckRecordInWorkflow	wfin26producteditapprovalv3....
☑	InRecordList	InitiateSubFlow	wfin26producteditapprovalv3....

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

Clean Up an Imported Process

You can clean up an imported process to avoid overlapping tasks and transitions.

When you import a process into TIBCO 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 unnconnected 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.

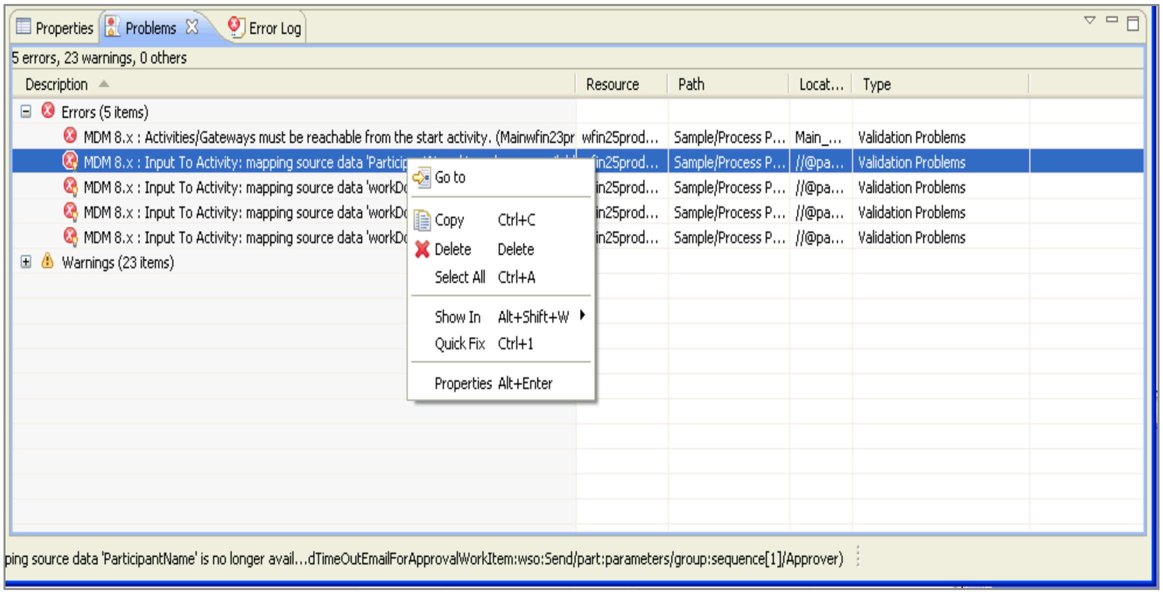
> **Note:** Unconnected activities will be overlapped after import. You need to clean up the layout and spread out the activities.

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.)

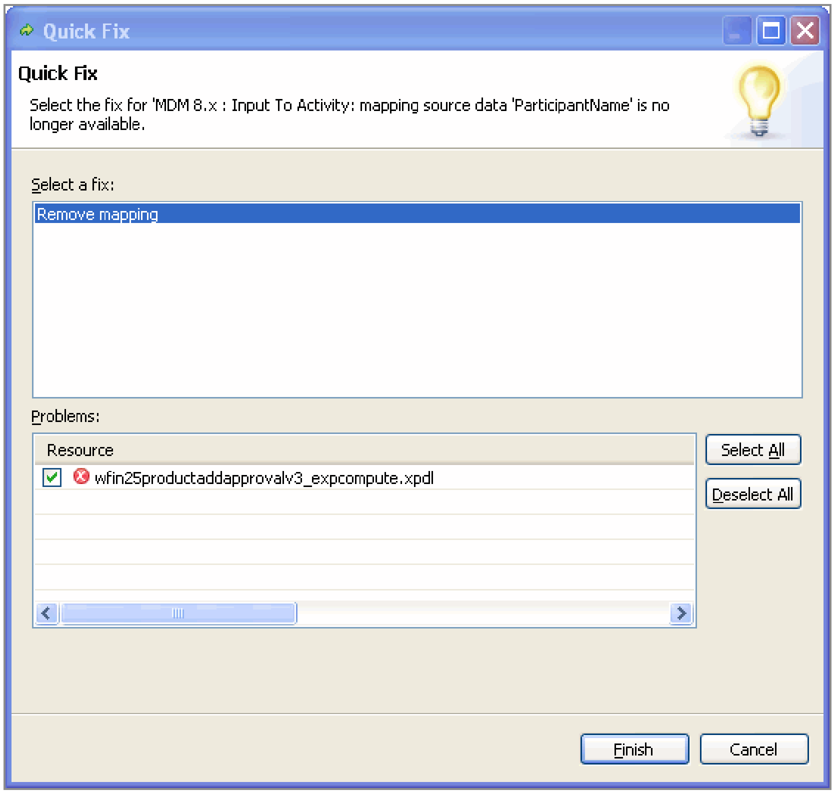
The Quick Fix dialog will display the problem and applicable resources. Click **Finish**.



5 errors, 23 warnings, 0 others

Description	Resource	Path	Locat...	Type
Errors (5 items)				
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 'workDc	in25prod...	Sample/Process P...	//@pa...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'workDc	in25prod...	Sample/Process P...	//@pa...	Validation Problems
MDM 8.x : Input To Activity: mapping source data 'workDc	in25prod...	Sample/Process P...	//@pa...	Validation Problems
Warnings (23 items)				

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



Quick Fix

Select the fix for 'MDM 8.x : Input To Activity: mapping source data 'ParticipantName' is no longer available.

Select a fix:

Remove mapping

Problems:

Resource
☑ wfin25productaddapprovalv3_expcompute.xpdl

Link Events

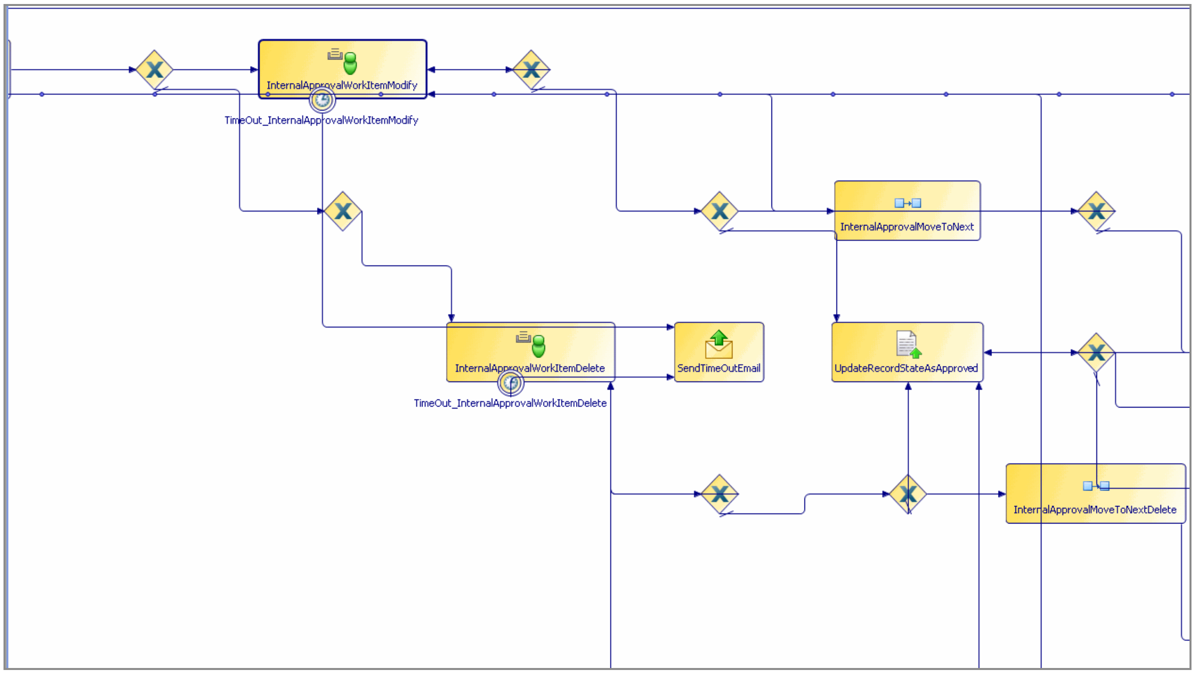
- Use Link Events for far stretching transitions.

Activity Placement and Names

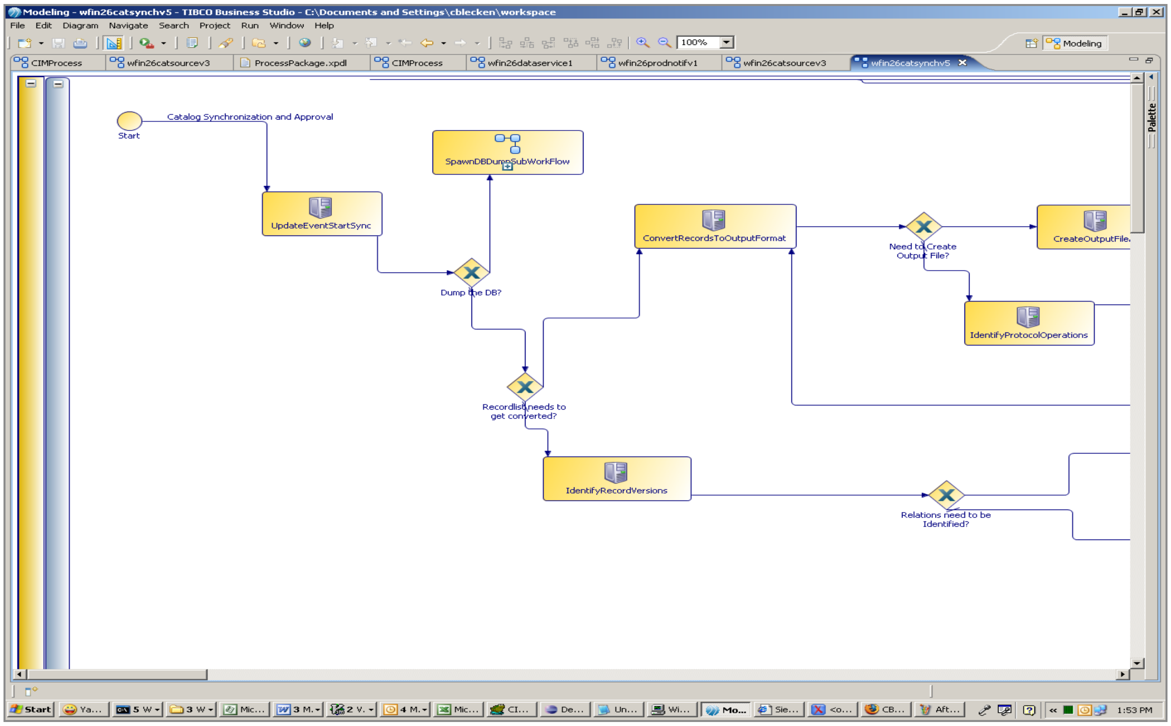
- Rearrange activities if required or overlapping.
- Expand activity rectangles to fit the name.

> **Note:** 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


InternalApprovalWorkItemModify
TimeOut_InternalApprovalWorkItemModify
InternalApprovalMoveToNext
InternalApprovalWorkItemDelete
TimeOut_InternalApprovalWorkItemDelete
SendTimeOutEmail
UpdateRecordStateAsApproved
InternalApprovalMoveToNextDelete

Sample of an imported process after cleanup


Modeling - wfin26catsynchv5 - TIBCO Business Studio - C:\Documents and Settings\cblecken\workspace
File Edit Diagram Navigate Search Project Run Window Help
100%
Modeling
CIMProcess
wfin26catsourcev3
ProcessPackage.xpdl
CIMProcess
wfin26dataservice1
wfin26prodnotifv1
wfin26catsourcev3
wfin26catsynchv5
Palette
Start
Catalog Synchronization and Approval
UpdateEventStartSync
SpawnDBDumpSubWorkFlow
Dump the DB?
ConvertRecordsToOutputFormat
Need to Create Output File?
CreateOutputFile
IdentifyProtocolOperations
Recordlist needs to get converted?
IdentifyRecordVersions
Relations need to be Identified?
Start
1:53 PM

Limitations of the Process Import

- 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.

Note: 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 TIBCO 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 TIBCO MDM Engine.

Migrating Processes

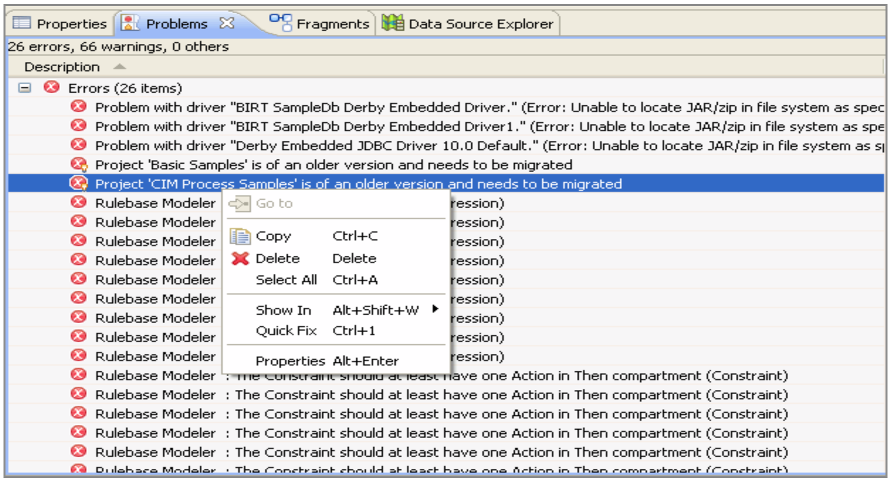
To migrate any older version projects into the current Process Designer, follow these steps:

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



26 errors, 66 warnings, 0 others

Errors (26 items)

Problem with driver "BIRT SampleDb Derby Embedded Driver." (Error: Unable to locate JAR/zip in file system as spec

Problem with driver "BIRT SampleDb Derby Embedded Driver1." (Error: Unable to locate JAR/zip in file system as spe

Problem with driver "Derby Embedded JDBC Driver 10.0 Default." (Error: Unable to locate JAR/zip in file system as s

Project 'Basic Samples' is of an older version and needs to be migrated

Project 'CIM Process Samples' is of an older version and needs to be migrated

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler ression)

Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)

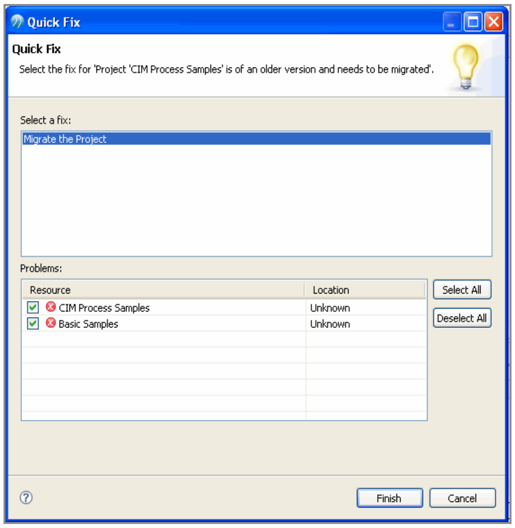
Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)

Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)

Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)

Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)

Rulebase Modeler : The Constraint should at least have one Action in Then compartment (Constraint)



Quick Fix

Select the fix for 'Project 'CIM Process Samples' is of an older version and needs to be migrated'.

Select a fix:

Migrate the Project

Problems:

Resource	Location
☑ CIM Process Samples	Unknown
☑ Basic Samples	Unknown

Directly Deploying Workflows

The Process Designer supports direct deployment of workflows. You do not need to manually export your process and import it into TIBCO MDM.

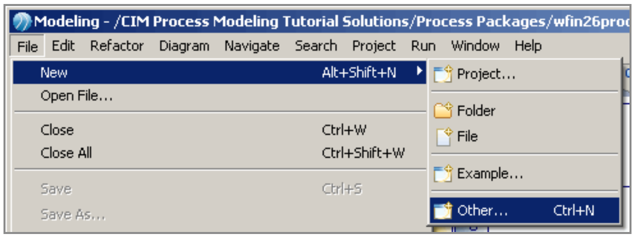
This network deployment is a web service, through which, you can list processes, deploy process definitions or undeploy process definitions.

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

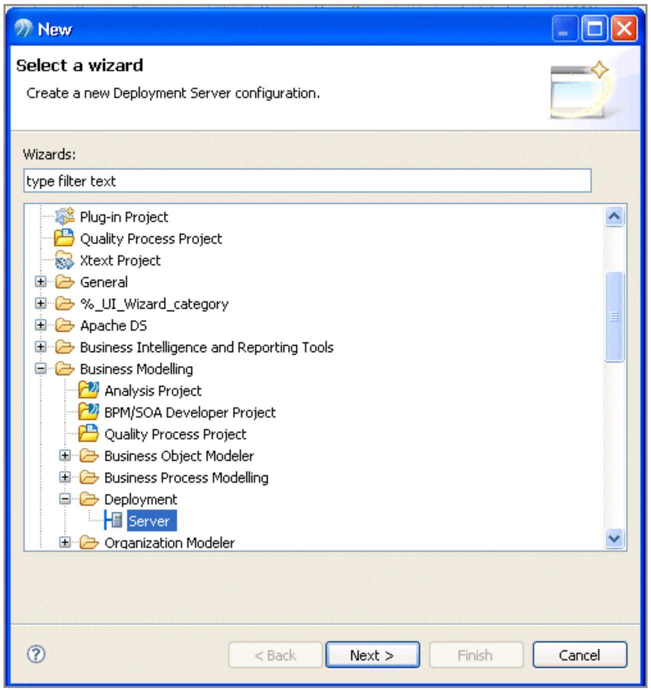
Creating a MDM Deployment Server

Procedure

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


Modeling - /CIM Process Modeling Tutorial Solutions/Process Packages/wfin26proc
File Edit Refactor Diagram Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Project...
Folder
File
Example...
Other... Ctrl+N

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


New

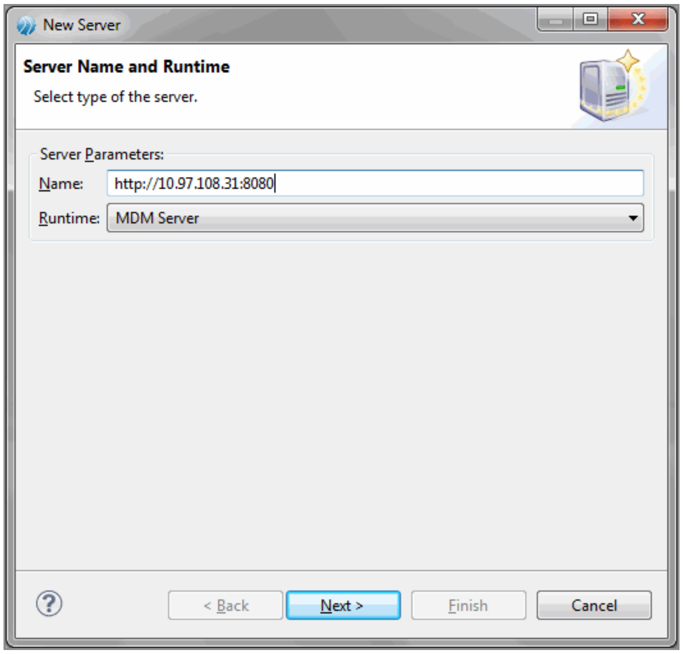
Select a wizard

Create a new Deployment Server configuration.

Wizards:

type filter text

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



New Server

Server Name and Runtime

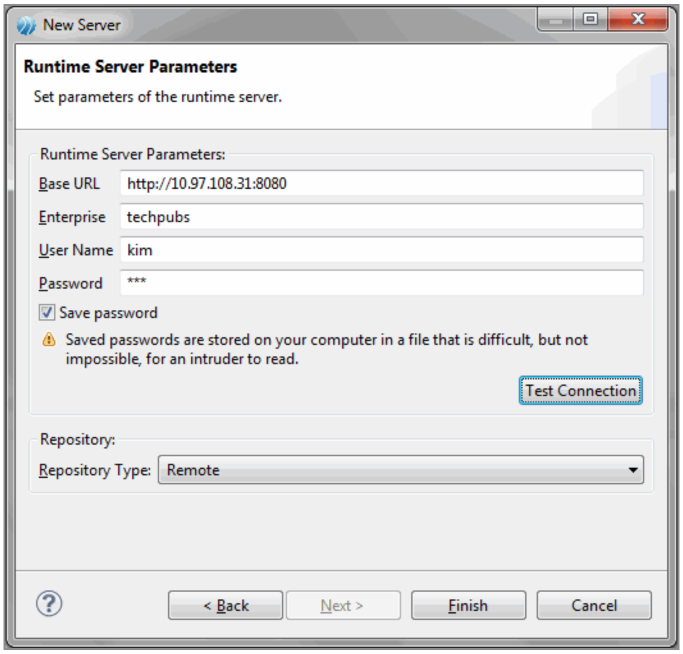
Select type of the server.

Server Parameters:

Name: http://10.97.108.31:8080

Runtime: MDM Server

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



New Server

Runtime Server Parameters

Set parameters of the runtime server.

Runtime Server Parameters:

Base URL: http://10.97.108.31:8080

Enterprise: techpubs

User Name: kim

Password: ***

☑ Save password

Saved passwords are stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

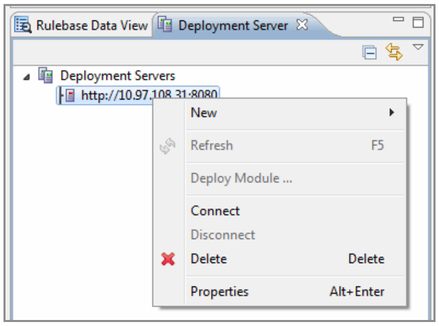
Repository:

Repository Type: Remote

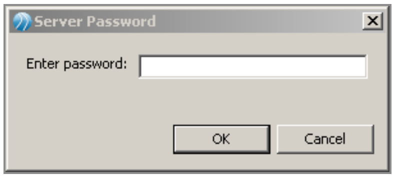
If you select the Save password option, you will not be prompted for a password in the following step.

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

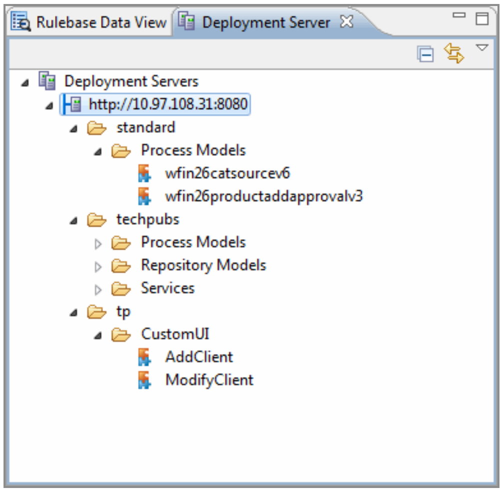
Result



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

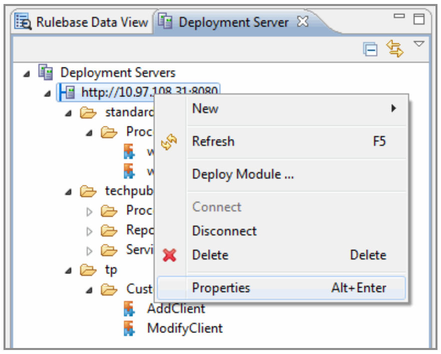


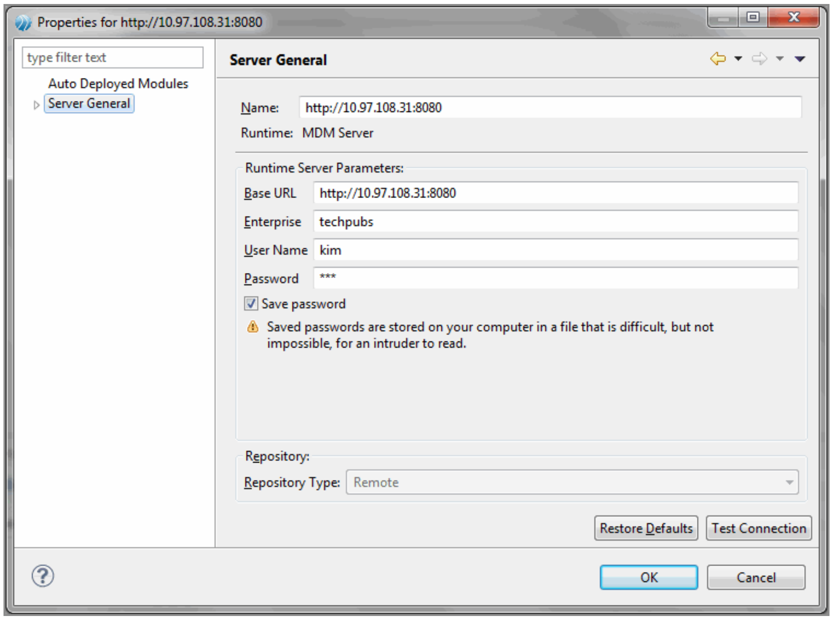
This will establish a connection between the MDM Server and the Process Designer client and will show all deployed workflows on the MDM server.



Editing Server Parameters

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





Server General

Name: http://10.97.108.31:8080

Runtime: MDM Server

Runtime Server Parameters:

Base URL http://10.97.108.31:8080

Enterprise techpubs

User Name kim

Password ***

☑ Save password

Saved passwords are stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

Repository:

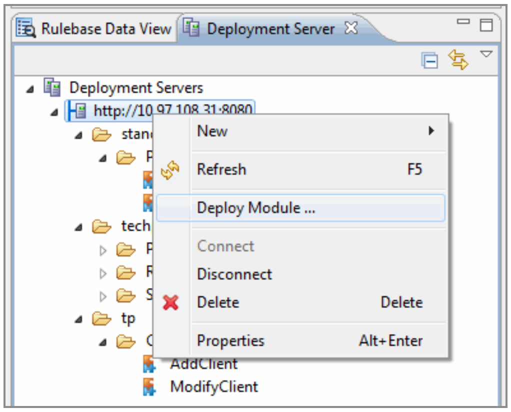
Repository Type: Remote

Deploying Workflows

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

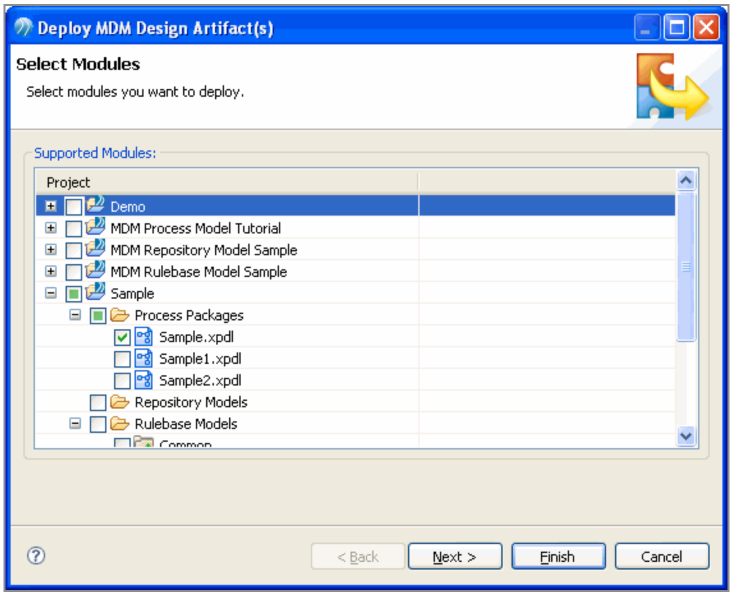
Procedure

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



> **Note:** 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**.

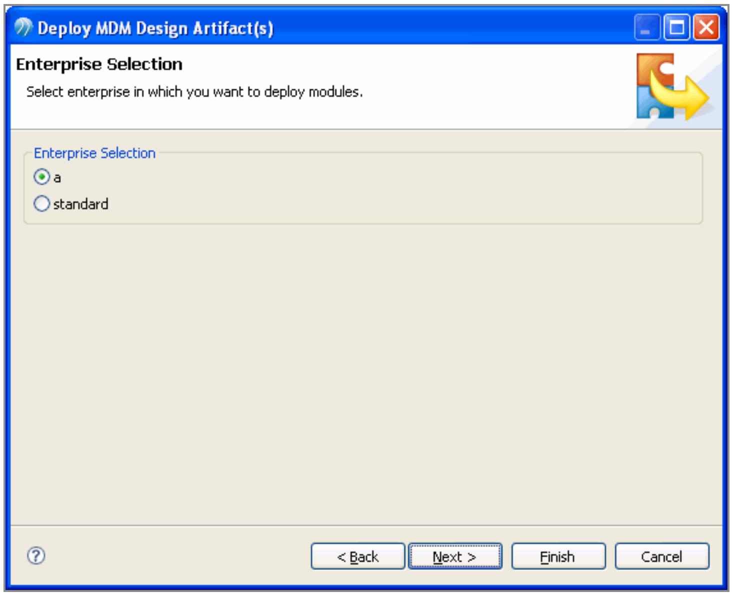


Select Modules

Select modules you want to deploy.

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

Result



Deploy MDM Design Artifact(s)

Enterprise Selection

Select enterprise in which you want to deploy modules.

Enterprise Selection

● a

○ standard

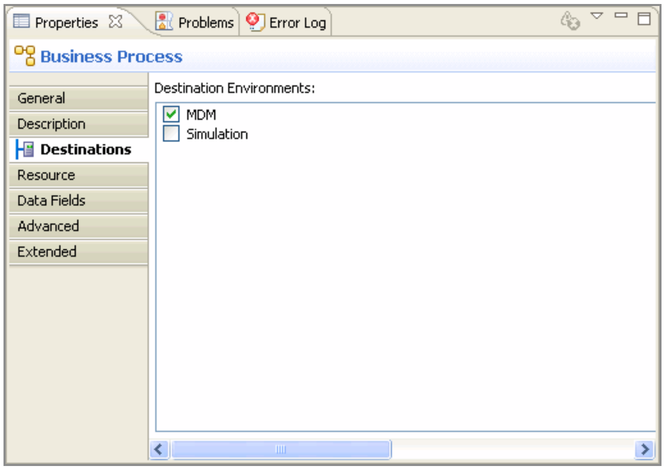
This will either deploy the selected workflow and will show a successful deployment message or an error message if the workflow could not be deployed.

Deploying Workflows through Export

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.

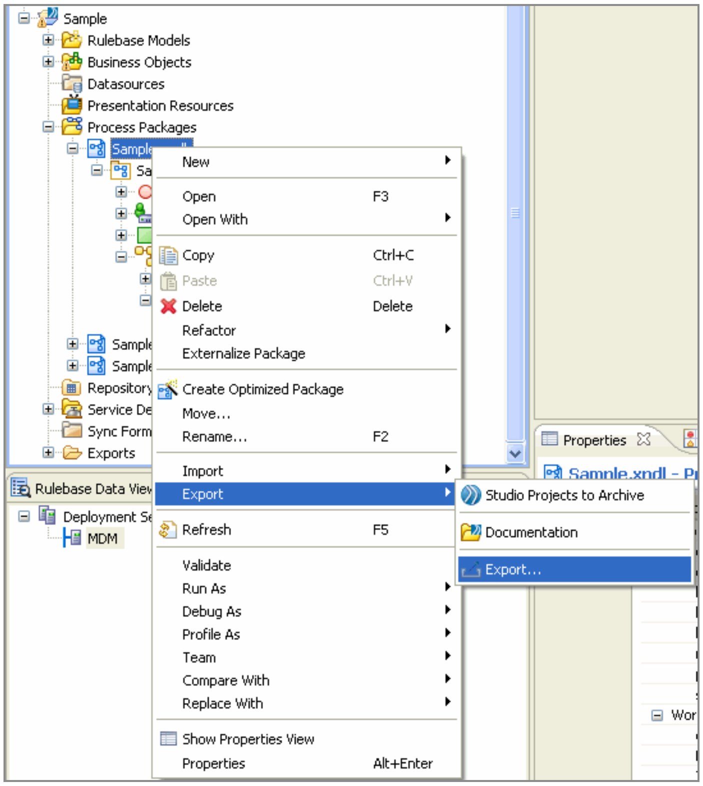


Destination Environments:

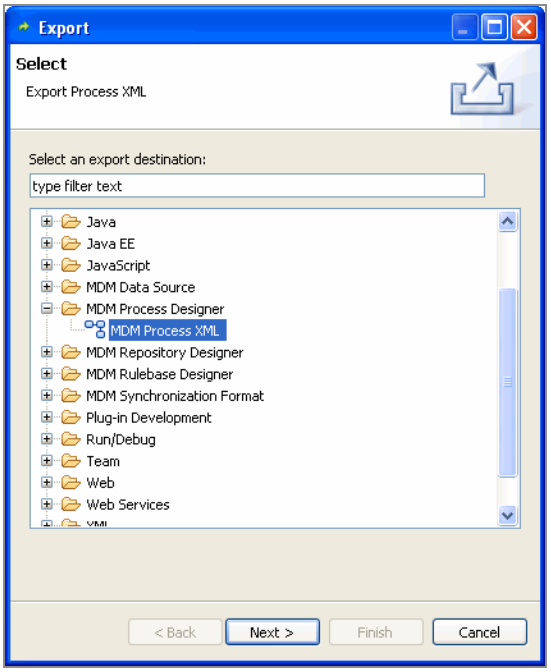
☑ MDM

□ Simulation

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



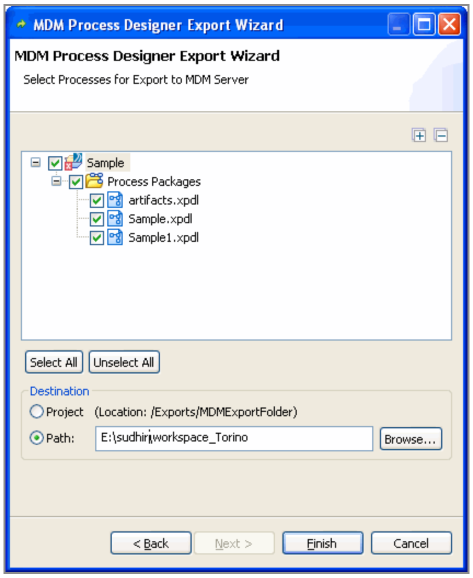
3. Select **MDM Process XML** under **MDM Process Designer** and click **Next**.



Select an export destination: type filter text

MDM Process Designer > MDM Process XML

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


MDM Process Designer Export Wizard
MDM Process Designer Export Wizard
Select Processes for Export to MDM Server
Select All
Unselect All
Browse...
< Back
Next >
Finish
Cancel

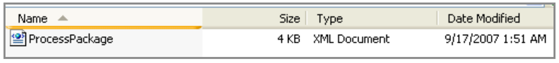
☑ Sample
☑ Process Packages
☑ artifacts.xpdl
☑ Sample.xpdl
☑ Sample1.xpdl

Destination

○ Project (Location: /Exports/MDMExportFolder)

● Path: E:\sudhir\workspace_Torino

5. 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

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

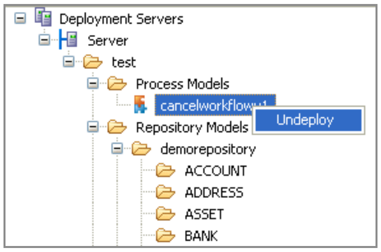
Workflows in MDM are saved in either of the following locations:

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

Note: 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

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

You can create and share processes with other users.

Overview

TIBCO MDM processes can be shared assuming that a subversion version control system has been set up and is ready to use.

TIBCO 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.

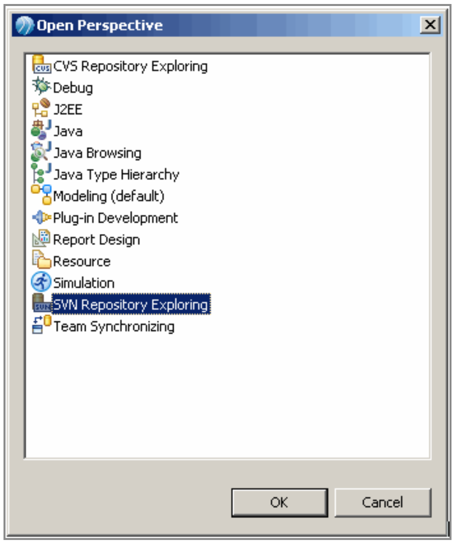
It is recommended that you use a standard version control system, such as CVS or Subversion/SVN. Both version control systems facilitate shared development of process designs and also allow versioning the projects.

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

Setting up MDM Process Designer to use the Subversion Repository

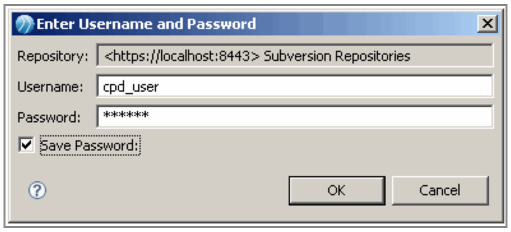
Procedure

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**.


Enter Username and Password

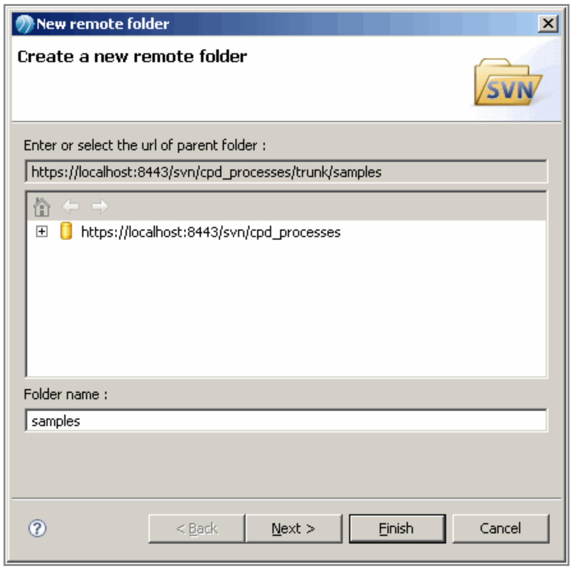
Repository: <https://localhost:8443> Subversion Repositories

Username: cpd_user

Password: ******

☑ Save Password:

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.



New remote folder

Create a new remote folder

Enter or select the url of parent folder :

https://localhost:8443/svn/cpd_processes/trunk/samples

https://localhost:8443/svn/cpd_processes

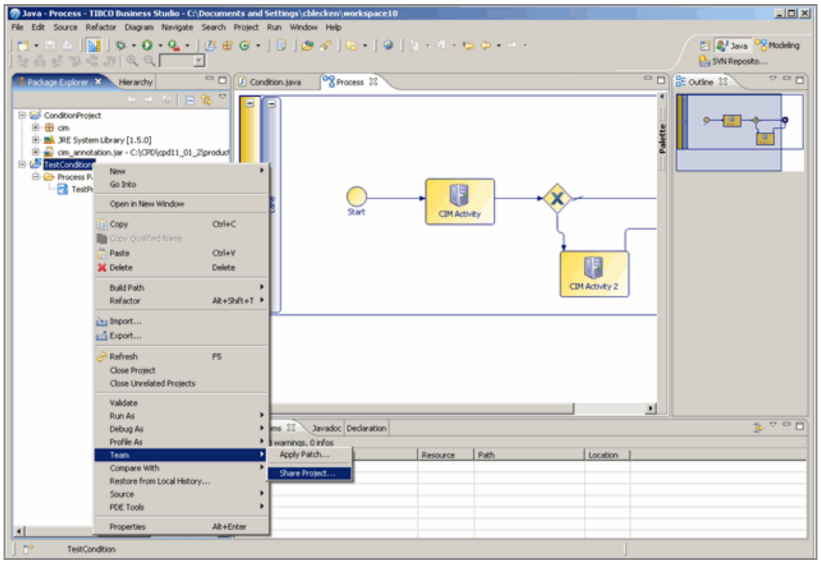
Folder name :

samples

Sharing a Process Design Project

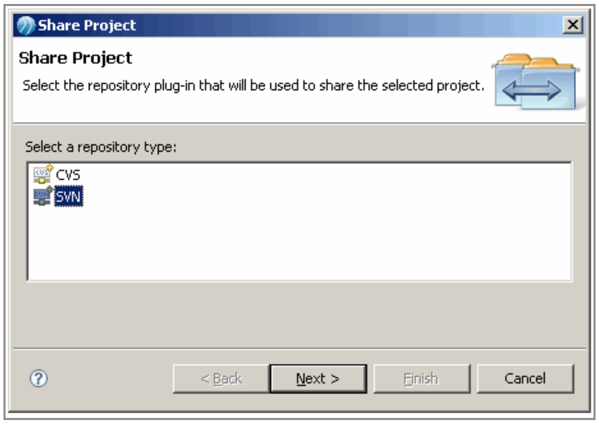
Procedure

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



Java - Process - TIBCO Business Studio - C:\Documents and Settings\cblecken\workspace10

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**.



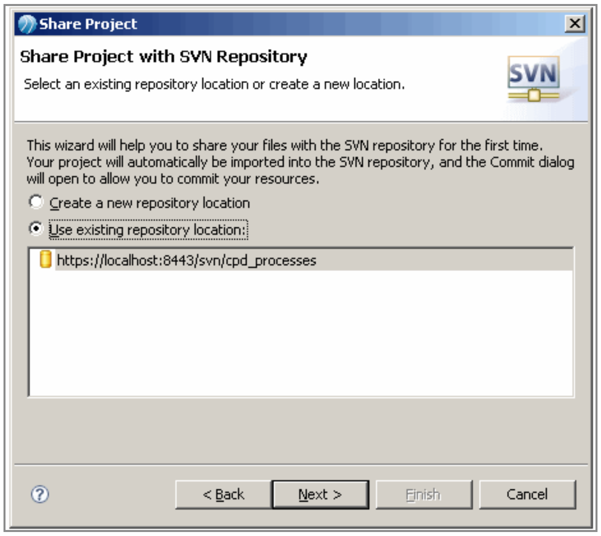
Share Project

Select the repository plug-in that will be used to share the selected project.

Select a repository type:

- CVS
- SVN

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.



Share Project with SVN Repository

Select an existing repository location or create a new location.

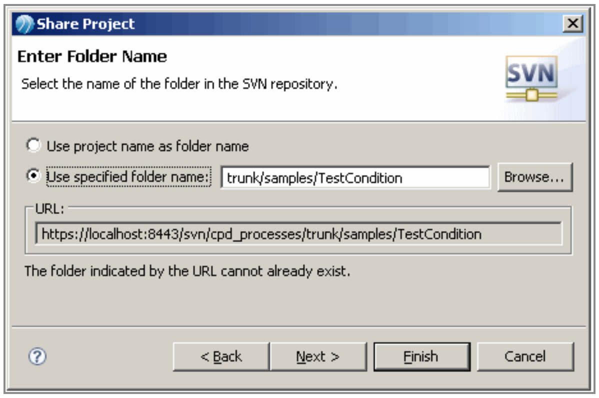
This wizard will help you to share your files with the SVN repository for the first time. Your project will automatically be imported into the SVN repository, and the Commit dialog will open to allow you to commit your resources.

○ Create a new repository location

● Use existing repository location:

https://localhost:8443/svn/cpd_processes

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



Enter Folder Name

Select the name of the folder in the SVN repository.

○ Use project name as folder name

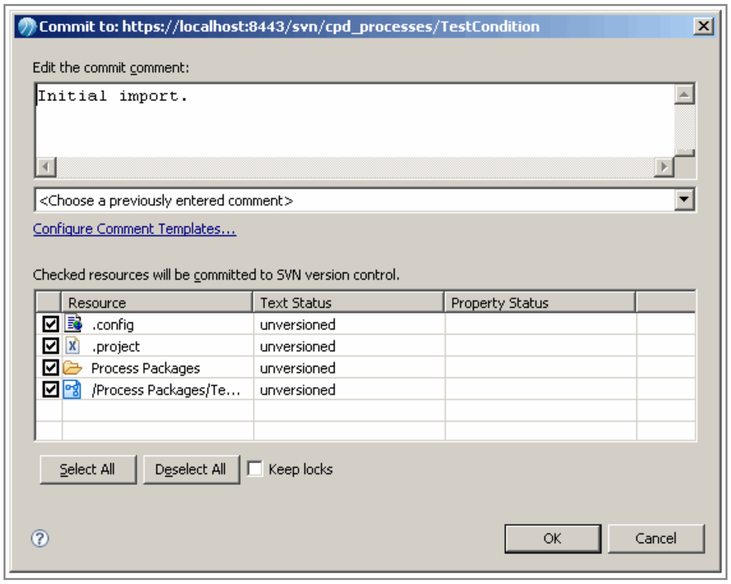
● Use specified folder name: trunk/samples/TestCondition Browse...

URL: https://localhost:8443/svn/cpd_processes/trunk/samples/TestCondition

The folder indicated by the URL cannot already exist.

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

Result



Commit to: https://localhost:8443/svn/cpd_processes/TestCondition

Edit the commit comment:

```
Initial import.
```

<Choose a previously entered comment>

Configure Comment Templates...

Checked resources will be committed to SVN version control.

Resource	Text Status	Property Status
☑ .config	unversioned	
☑ .project	unversioned	
☑ Process Packages	unversioned	
☑ /Process Packages/Te...	unversioned	

□ Keep locks

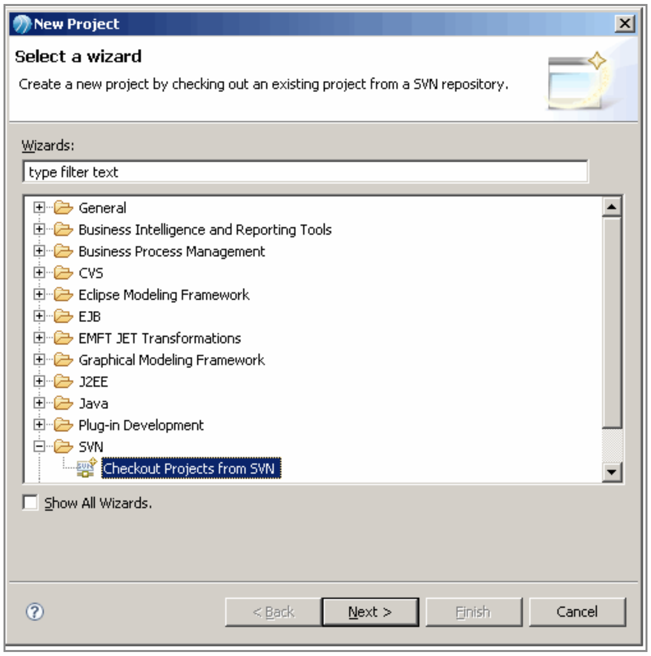
Your project is now shared and the following section explains how other users can access it.

Accessing a shared project

Note: 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**.



New Project

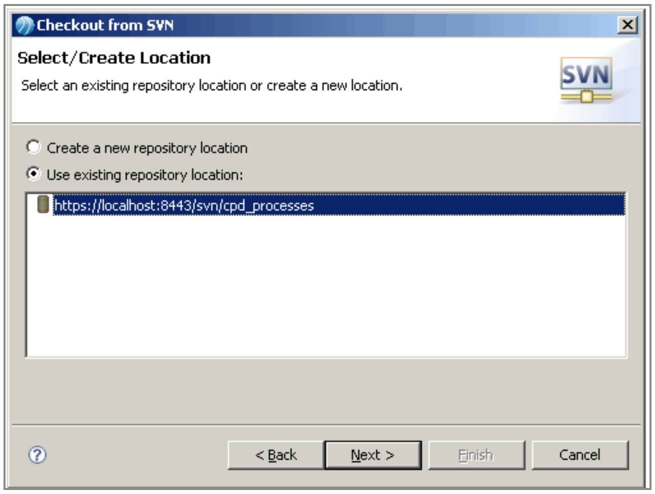
Select a wizard

Create a new project by checking out an existing project from a SVN repository.

Wizards: type filter text

General
Business Intelligence and Reporting Tools
Business Process Management
CVS
Eclipse Modeling Framework
EJB
EMFT JET Transformations
Graphical Modeling Framework
J2EE
Java
Plug-in Development
SVN
Checkout Projects from SVN
Show All Wizards.
< Back
Next >
Finish
Cancel

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



Checkout from SVN

Select/Create Location

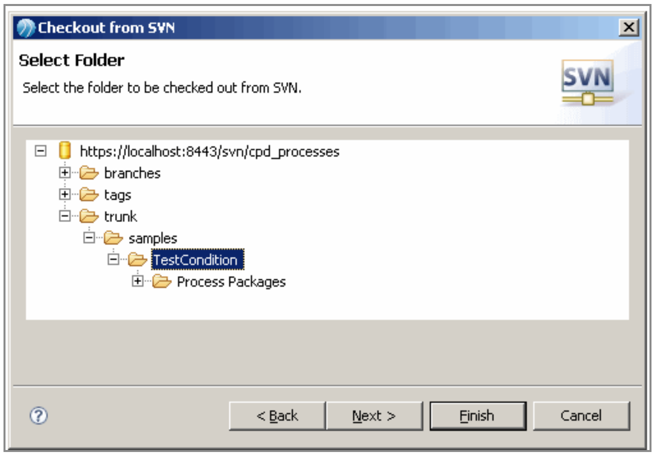
Select an existing repository location or create a new location.

○ Create a new repository location

● Use existing repository location:

https://localhost:8443/svn/cpd_processes

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



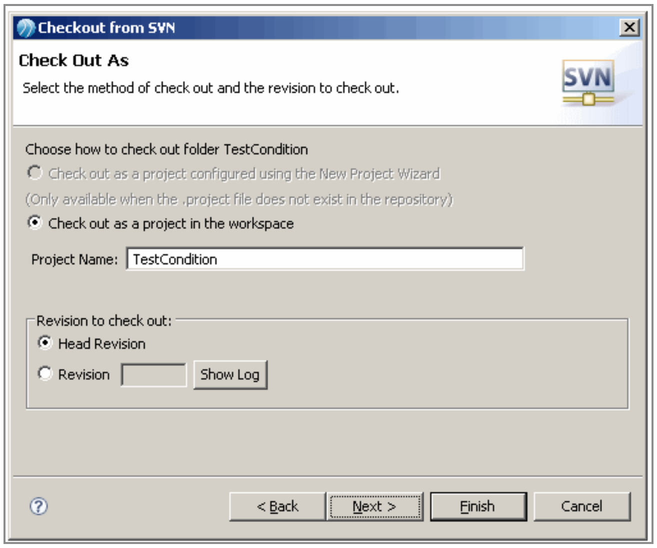
Checkout from SVN

Select Folder

Select the folder to be checked out from SVN.

https://localhost:8443/svn/cpd_processes
branches
tags
trunk
samples
TestCondition
Process Packages

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



Checkout from SVN

Check Out As

Select the method of check out and the revision to check out.

Choose how to check out folder TestCondition

○ Check out as a project configured using the New Project Wizard

(Only available when the .project file does not exist in the repository)

● Check out as a project in the workspace

Project Name: TestCondition

Revision to check out:

● Head Revision

○ Revision

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact Support, and join Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO® MDM Studio Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

Cloud SG 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 a specific version of Cloud SG software 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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.