



# **TIBCO® MDM Studio**

## **Process Designer Tutorial**

Version 6.1.1 | June 2024

Document Updated | October 2024

# Contents

---

<b>Contents</b>	<b>2</b>
<b>Tutorial to Create a Sample Process</b>	<b>4</b>
Creating a New Project Package and Process	4
Sample Flow Layout and Project Components	10
Modifying Exit Conditions from the Main Process	11
Defining Global and Local Variables	12
Defining Global Variables	12
Defining Local Variables	14
Add and Map Activity Parameters to the Process	16
Adding Activities	16
Defining Input Parameters	16
Defining Output Parameters	17
Adding an UpdateEvent Activity	17
Add a Conditional Transition	20
Adding DeleteRecord activity for Conditional Transition	21
Define Input Parameters to the SaveRecord Activity	25
Define Output Parameters to the SaveRecord Activity	26
Add More Activities	30
UpdateRecordState Activity	30
Setting SendProtocolMessage Activity	32
Define Output Parameters to the SendProtocolMessage Activity	36
UpdateEvent Activity	37
Validation	39
Exporting the Process to a File	40
Deploy the Process Model	42
<b>TIBCO Documentation and Support Services</b>	<b>43</b>

<b>Legal and Third-Party Notices .....</b>	<b>45</b>
--	-----------

# Tutorial to Create a Sample Process

---

To get started with the Process Designer, use this tutorial to create a simple process.

In this tutorial, we will do the following:

[Creating a New Project Package and Process](#)

[Modifying Exit Conditions from the Main Process](#)

[Defining Global and Local Variables](#)

[Add and Map Activity Parameters to the Process](#)

[Add a Conditional Transition](#)

[Add More Activities](#)

[Validation](#)

[Exporting the Process to a File](#)

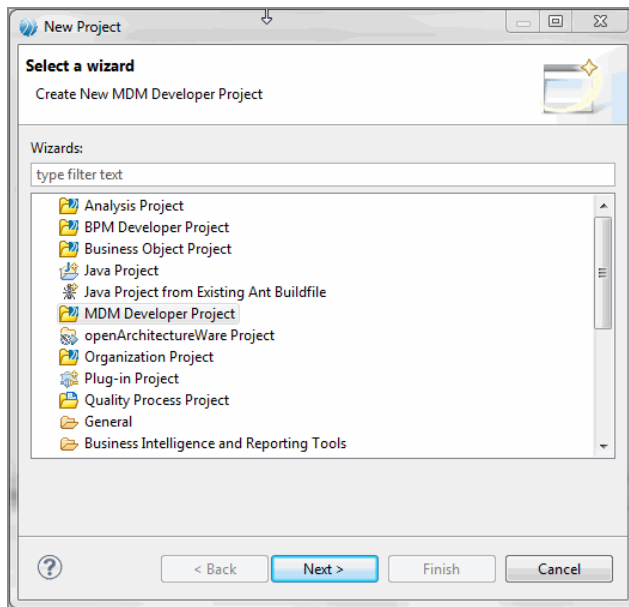
## Creating a New Project Package and Process

In this section, we follow a simple wizard to create a new Project to hold the process definition. The first step is to create a Project that will contain the Process. A Project contains a Package, and a Package contains Processes.

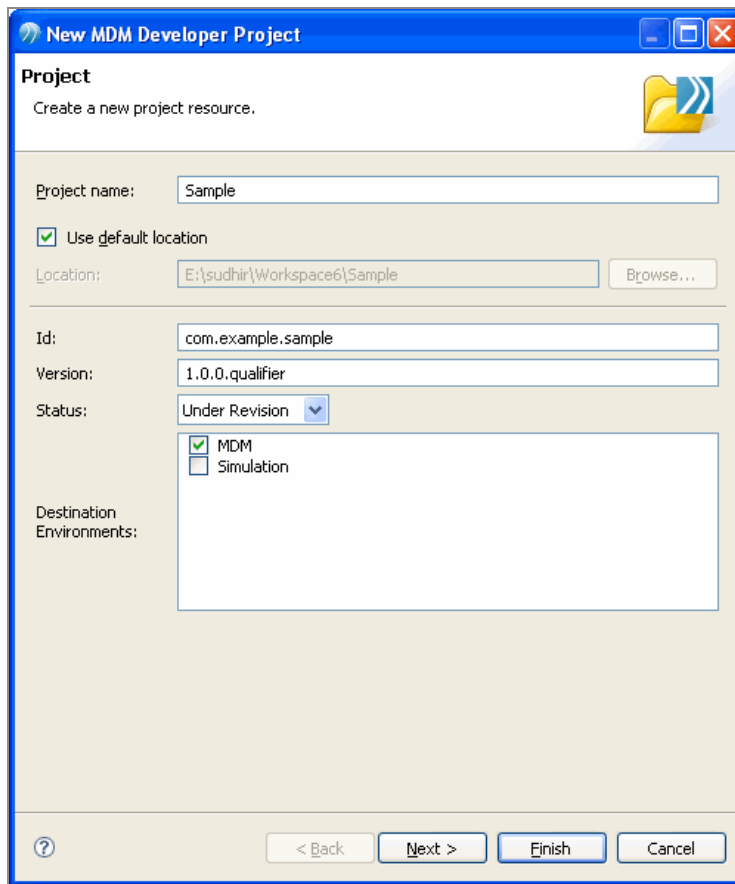
Follow these steps to create a Project:

### Procedure

1. In TIBCO Business Studio, select **File > New > Project**. The new project wizard is displayed.



2. Select **MDM Developer Project** and click **Next**.
3. In the new Project dialog, enter **Sample** for the Project name. The **Use default location** checkbox is selected by default, indicating that the project will be saved to the workspace directory. You can browse and select a different location. Select **MDM** as the Destination Environment. Click **Next**.

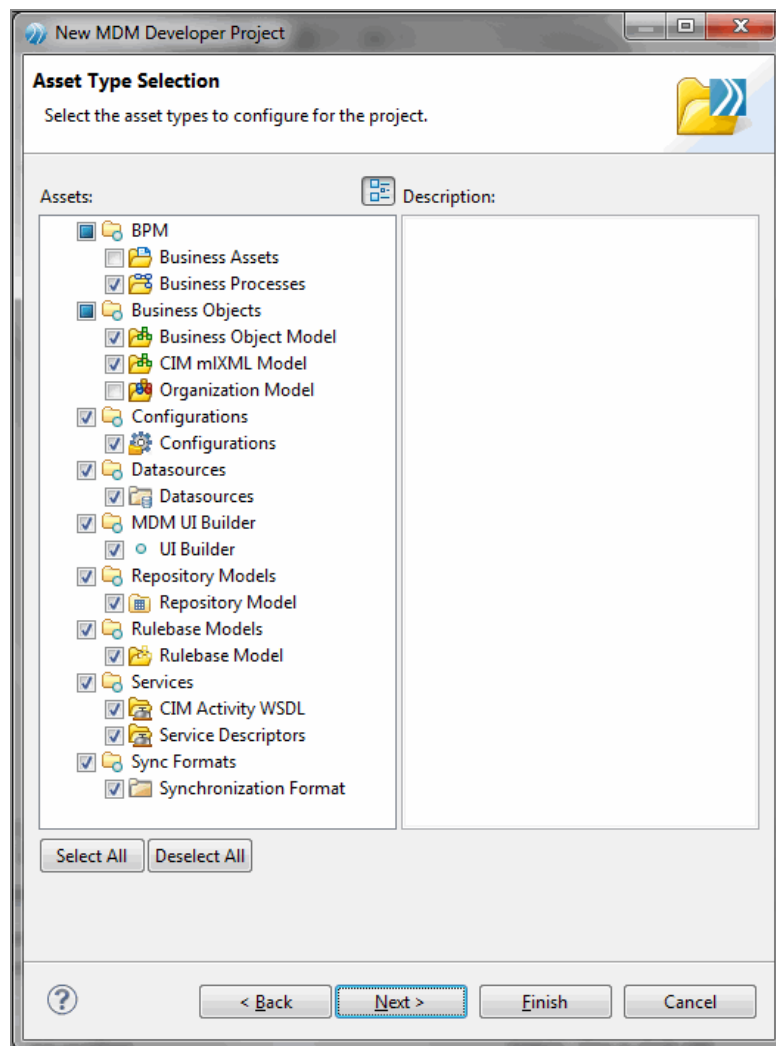


The image shows a Windows-style dialog box titled "New MDM Developer Project". The main heading is "Project" with the instruction "Create a new project resource." and a folder icon. The form contains the following fields and controls:

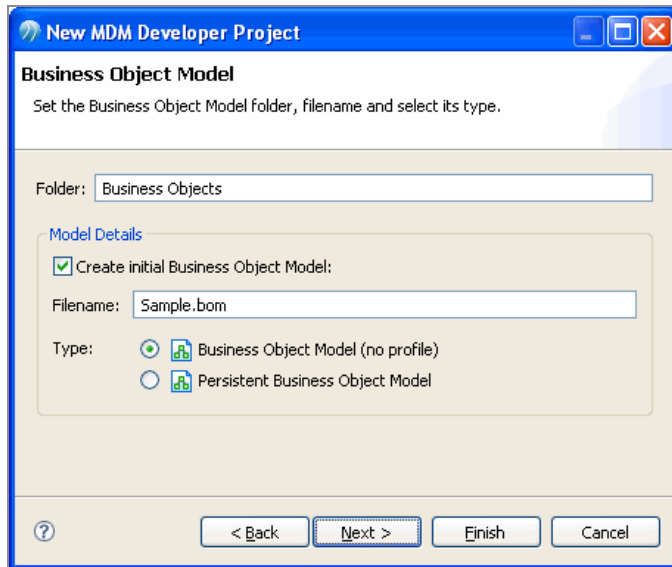
- Project name:** A text box containing "Sample".
- Use default location:** A checked checkbox.
- Location:** A text box containing "E:\sudhir\Workspace6\Sample" and a "Browse..." button.
- Id:** A text box containing "com.example.sample".
- Version:** A text box containing "1.0.0.qualifier".
- Status:** A dropdown menu currently showing "Under Revision".
- Destination Environments:** A list box containing two items: "MDM" (which is checked with a green box) and "Simulation" (which is unchecked).
- Navigation:** At the bottom, there are four buttons: "< Back", "Next >", "Finish", and "Cancel". A help icon (?) is located to the left of the "Back" button.

4. The Asset Type Selection dialog is displayed. Ensure that the following is selected and click **Next**.

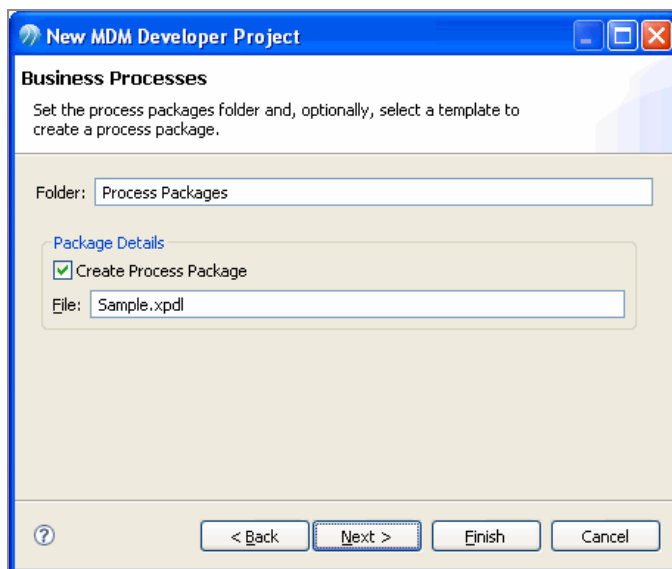
- **Business Processes** (under BPM)
- **Business Object Model** and **CIM mXML Model** (under Business Objects)
- **CIM Activity WSDL** and **Service Descriptors** (under Services)



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**.

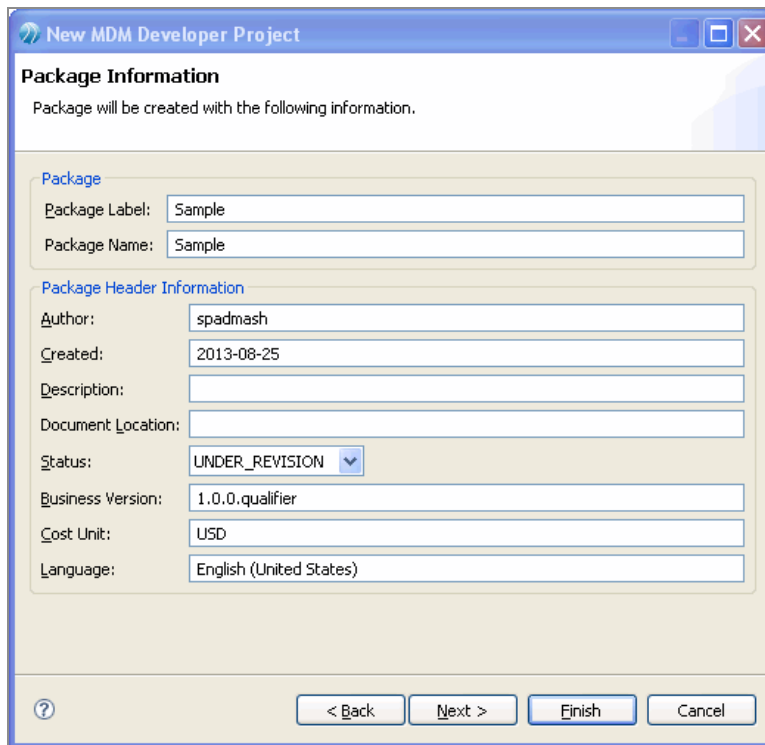


6. The Business Processes dialog is displayed next and prompts you to set the process packages folder. The new process package resource that will be created is displayed - this is the xpdL file. Click **Next**.



7. The process package details are displayed. Leave all the default names and click **Next**.





The image shows a Windows-style dialog box titled "New MDM Developer Project". It has a blue title bar with standard window controls. The main content area is titled "Package Information" and contains a sub-header "Package" with two text fields: "Package Label:" and "Package Name:", both containing the text "Sample". Below this is a sub-header "Package Header Information" followed by several fields: "Author:" (spadmash), "Created:" (2013-08-25), "Description:" (empty), "Document Location:" (empty), "Status:" (a dropdown menu showing "UNDER\_REVISION"), "Business Version:" (1.0.0.qualifier), "Cost Unit:" (USD), and "Language:" (English (United States)). At the bottom of the dialog are four buttons: a help button (question mark icon), "< Back", "Next >", and "Finish".

**New MDM Developer Project**

**Package Information**

Package will be created with the following information.

**Package**

Package Label: Sample

Package Name: Sample

**Package Header Information**

Author: spadmash

Created: 2013-08-25

Description:

Document Location:

Status: UNDER\_REVISION

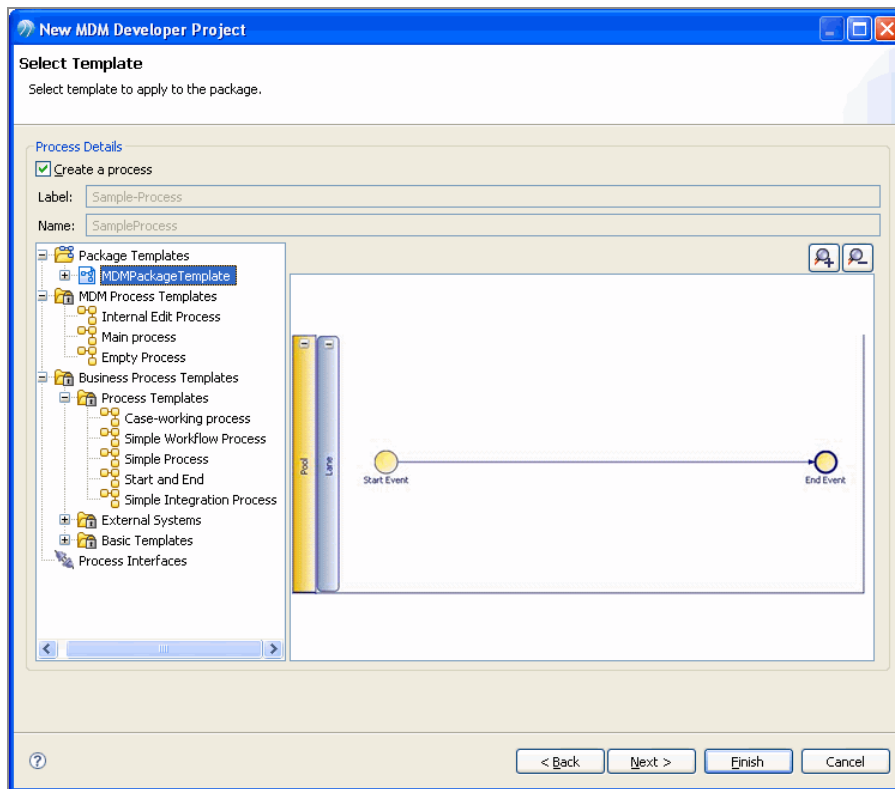
Business Version: 1.0.0.qualifier

Cost Unit: USD

Language: English (United States)

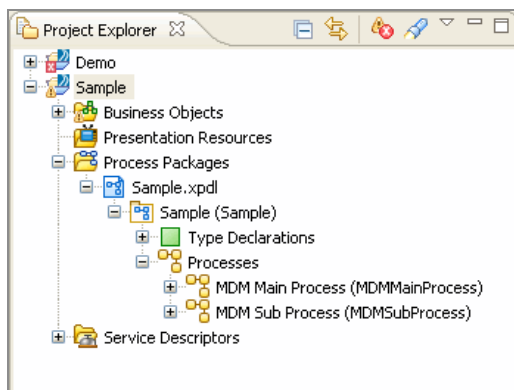
? < Back Next > Finish Cancel

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



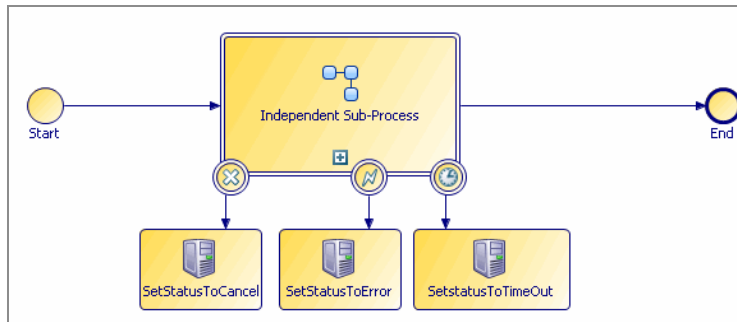
## Sample Flow Layout and Project Components

The newly-created Project is displayed in the Project Explorer. You can expand the Project to display the **Process Package** and **Processes**; expand the **Processes** to see a **MDM Main Process** and a **MDM Sub Process**.



Double-click the **MDM Main Process** in the **Project Explorer** to display the process diagram in the **Process Editor**. The Process Editor is where you create your business

process; it provides a modeling view where you can see all components in the process. It also includes a Palette with tools you can use to create your Process diagram.




This is a modeling view of the main process which contains:

- A start event.
- A main process, which in turn references an independent sub-process.
- An end event.
- Exit conditions - A cancel, exception, and timeout handler.

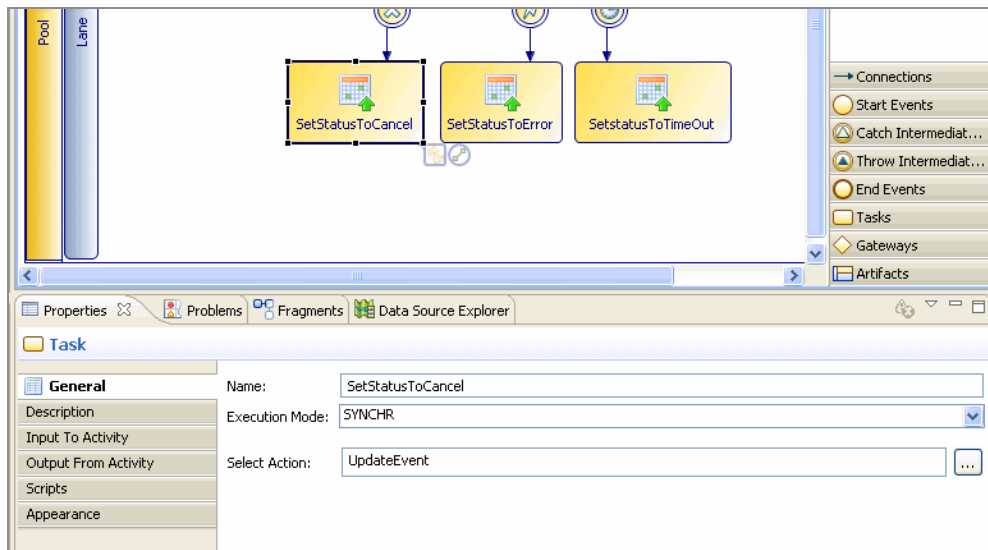
## Modifying Exit Conditions from the Main Process

In this section, we modify the Cancel Handler to handle cancellation from any point in the process.

### Procedure

1. Click on the Cancel Handler - its properties will be displayed in the **Properties** Window below.
2. Enter a name in the **Name** field, for example, "SetStatusToCancel".
3. Select the **Execution Mode** from the drop-down list, for example "SYNCHR".
4. Click the Select  button against **Select Action**.
5. In the Select Activity Action dialog that is displayed, select **UpdateEvent** and click **OK**.

## Result



Similarly, you can edit properties for the Timeout or Exception Handlers or just remove the activities with the timeout and exception handler.

# Defining Global and Local Variables

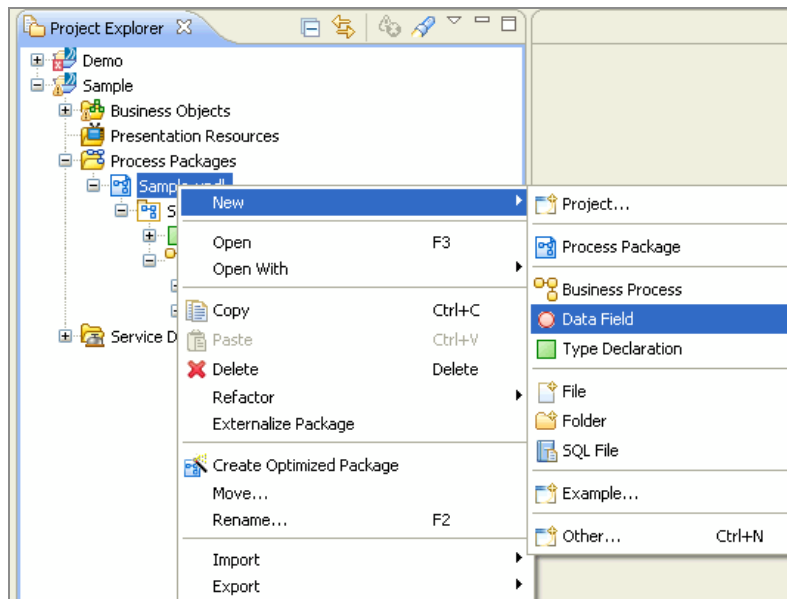
In this section, we define two global variables - InDoc and Outdoc - that are defined in the context of the process. We also define some local variables.

In the Process Designer, global variables are defined at the package level, and local variables at the process level. Once you create a new project, you have to create the **Data Fields** under the package name (you define global variables from here) as well as under the process itself (you define local variables from here).

## Defining Global Variables

### Procedure

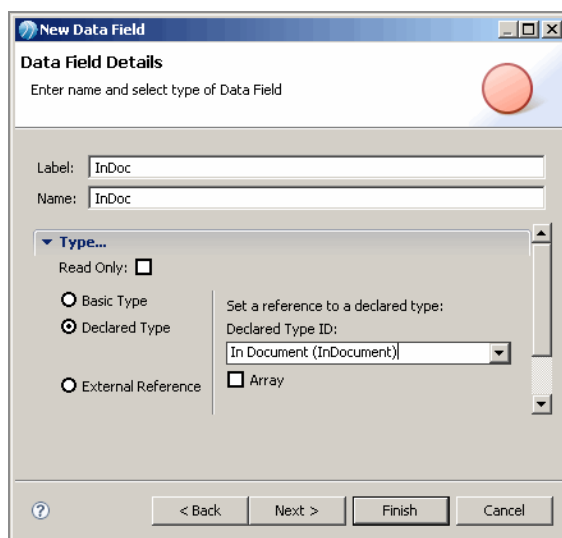
1. Go to the Project Explorer and navigate to your **Sample** project.
2. Right click on Package Name (For example, Sample) and select **New > Data Field**.



## InDoc

### Procedure

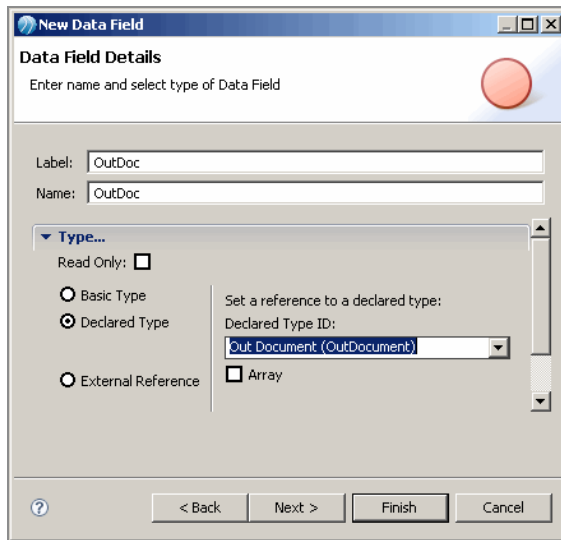
1. In the Data Field Details dialog, change the label to **InDoc** and set the **Declared Type** to **InDocument**.
2. Click **Finish**.



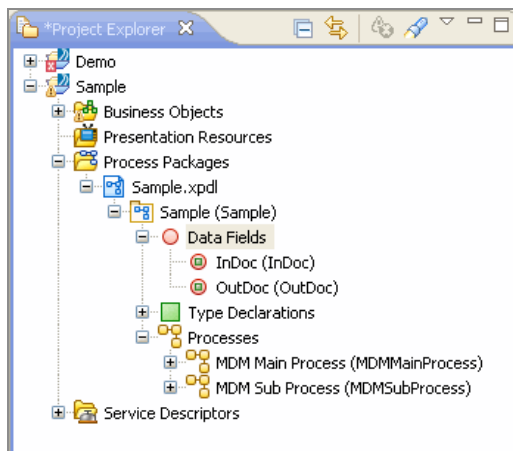
## OutDoc

### Procedure

1. In the Data Field Details dialog, change the label to **OutDoc** and set the **Declared Type** to **OutDocument**.
2. Click **Finish**.



3. The Global Variable is displayed on the Project Explorer.

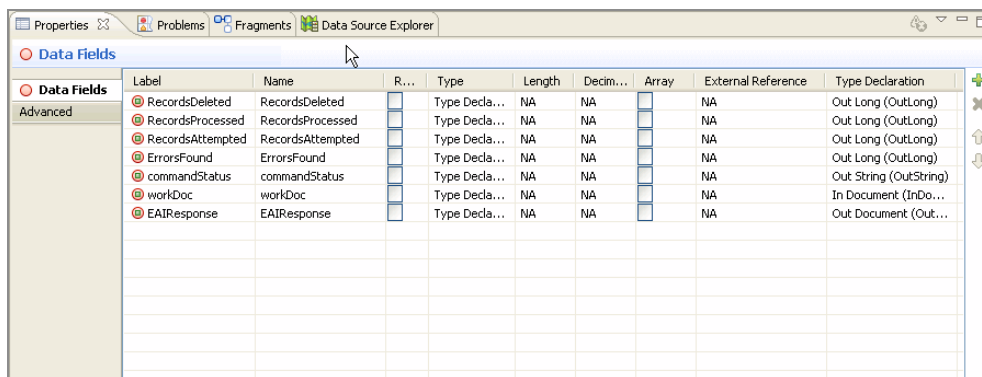


## Defining Local Variables

Do the following to define local variables:

- In the Project Explorer, go to **Data Field** within the MDM Sub Process.
- Right click on **Data Field** and select **New > Data Field**.
- The Data Field Details dialog displayed:
- In the Data Field Details dialog enter data field name in the **Label**, and set the Declared Type as shown in the following table.

Data Field Name	Set Declared Type to
RecordsDeleted	OutLong
RecordsProcessed	OutLong
RecordsAttempted	OutLong
ErrorsFound	OutLong
commandStatus	OutString
workDoc	InDocument
EAIResponse	OutDocument



The screenshot shows the 'Data Source Explorer' window with a table of data fields. The table has columns: Label, Name, R..., Type, Length, Decim..., Array, External Reference, and Type Declaration. The 'Label' column contains radio buttons next to each field name. The 'Type Declaration' column shows the declared type for each field.

Label	Name	R...	Type	Length	Decim...	Array	External Reference	Type Declaration
<input checked="" type="radio"/>	RecordsDeleted	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out Long (OutLong)
<input checked="" type="radio"/>	RecordsProcessed	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out Long (OutLong)
<input checked="" type="radio"/>	RecordsAttempted	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out Long (OutLong)
<input checked="" type="radio"/>	ErrorsFound	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out Long (OutLong)
<input checked="" type="radio"/>	commandStatus	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out String (OutString)
<input checked="" type="radio"/>	workDoc	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	In Document (InDo...
<input checked="" type="radio"/>	EAIResponse	<input type="checkbox"/>	Type Deda...	NA	NA	<input type="checkbox"/>	NA	Out Document (Out...

In the following sections, we map these local variables to some activity input and output parameters.

# Add and Map Activity Parameters to the Process

In this section, we begin defining our process by adding activities and mapping the activity parameters to the process.

We start by adding an UpdateEvent activity to update the event details. We also define input parameters for this activity.

## Adding Activities

The following sections will guide you through adding activities to define your process. Follow these generic instructions to add an activity to your process flow:

### Procedure

1. In the Project Explorer, navigate to Sample Project, double-click the **MDMSubProcess**.
2. In the Project Explorer, navigate to Sample Project and expand **Service Descriptors**.
3. Select **CIMServices.wsdl->CimActivities-><activityname>** and drag and drop the activity into MDM Sub Process model.
4. Select **Create Service Task To Invoke Operation** from the drop-down displayed when you add the activity to your flow.

## Defining Input Parameters

### 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. Follow the instructions to make modifications to set parameter values. For modification instruction refer [Modifying Exit Conditions from the Main Process](#).



## Defining Output Parameters

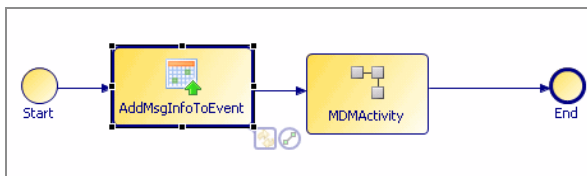
### Procedure

1. Select the activity (that you want to define parameters for) 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. Follow the instructions to make modifications to set parameter values. For modification instruction refer [Modifying Exit Conditions from the Main Process](#).

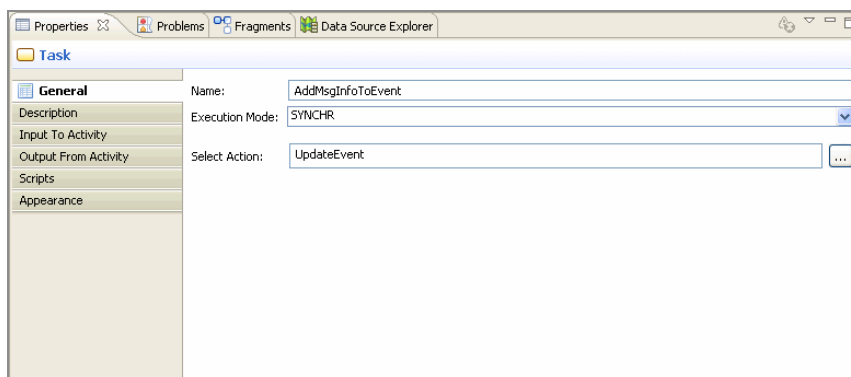
## Adding an UpdateEvent Activity

### Procedure

1. Add an **UpdateEvent** activity to the process by dragging and dropping it into your process model after the Start event. (this activity updates the event details).



2. Rename the UpdateEvent activity to **AddMsgInfoToEvent** by going to the Properties window and editing the Name field in the **General** tab.



## Define Input Parameters to the UpdateEvent Activity

[Setting eventState Parameter](#)

[Setting eventType Parameter](#)

[Setting eventDescriptor Parameter](#)

[Setting deploymentMode Parameter](#)

### Setting eventState Parameter

#### Procedure

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

### Setting eventType Parameter

#### Procedure

1. Select the **eventType** parameter.
2. Click on the text field corresponding to the eventType parameter and type the value **CAT**.

### Setting eventDescriptor Parameter

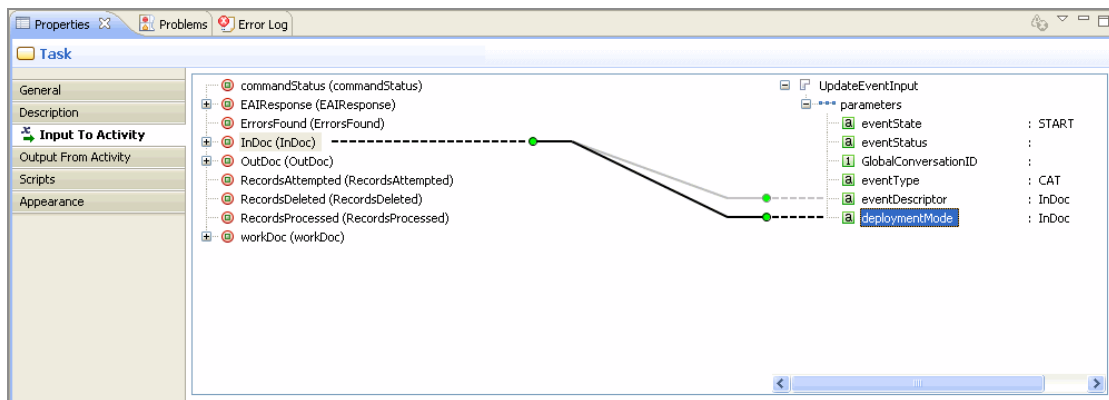
#### Procedure

1. Drag InDoc local variable and drop it on **eventDescriptor** input parameter.
2. Double click the **eventDescriptor** Parameter.
3. Select **Xpath** from the **Eval** drop-down list.
4. Add the following expression in **Source expression** text field  
\$InDoc/Body/Document/subtype.

## Setting deploymentMode Parameter

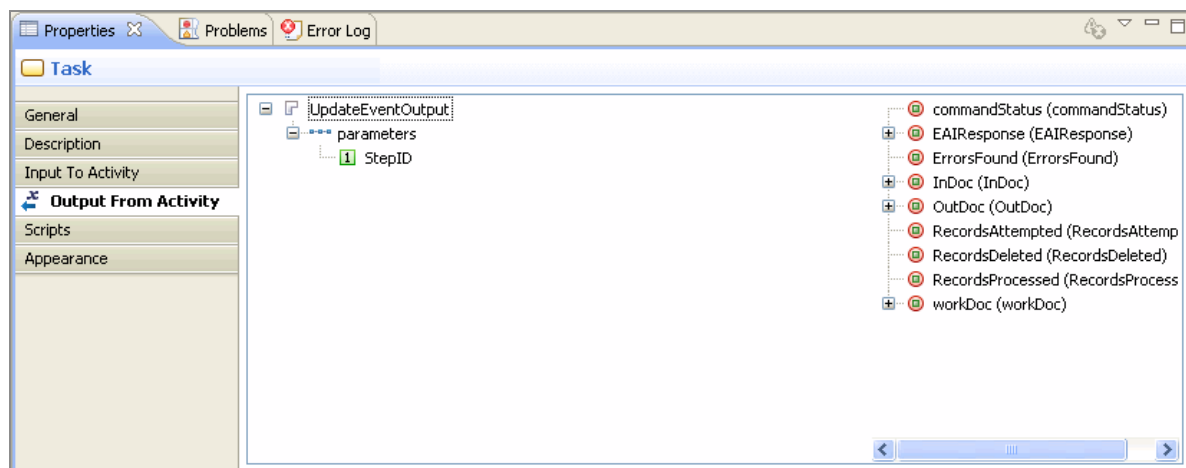
### Procedure

1. Drag InDoc local variable and drop it on **deploymentMode** input parameter.
2. Double click on **deploymentMode** Parameter.
3. Select **Xpath** from the **Eval** drop-down list.
4. Add the following expression in **Source expression** text field `$InDoc/MessageType`.



## Define Output Parameters to the UpdateEvent Activity

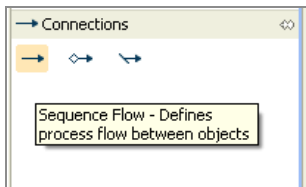
The Output From Activity tab displays the output parameters.



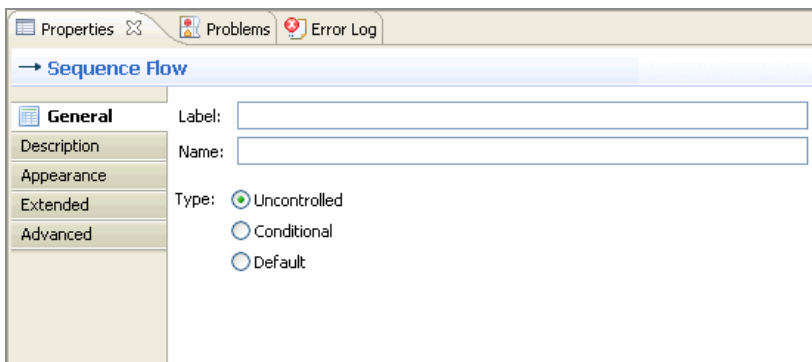
## Defining Sequence Flow

### Procedure

1. Select the **Sequence Flow** from the **Connections** Palette.

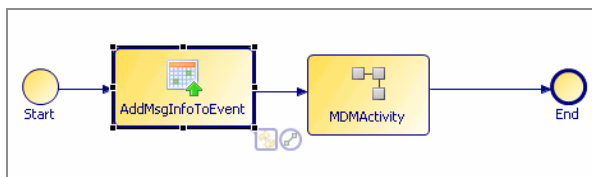


2. Click on the **Start** Event and drag it to **UpdateEvent** activity. In the **Sequence Flow** property section, select **Uncontrolled** Type.



3. Similarly click on the **UpdateEvent** activity and drag it to a **MDMActivity**.

### Look of your workflow at this point



## Add a Conditional Transition

In this section, we continue our process definition by adding SaveRecord and DeleteRecord activities, controlled by a conditional transition that looks for a Master Catalog name, and based on that, it either deletes or saves the record.

When we added the **UpdateEvent** activity (to the basic flow created by the MDM template) after dropped the activity on the Start Event, an uncontrolled transition was automatically created between the Start Event and the **UpdateEvent** activity.

For example, let's create a conditional transition. We will either save or delete the record depending on the Master Catalog name in the InDocument. By default, the record is saved, and only if the Master Catalog name is **ASSET**, the record will be deleted.

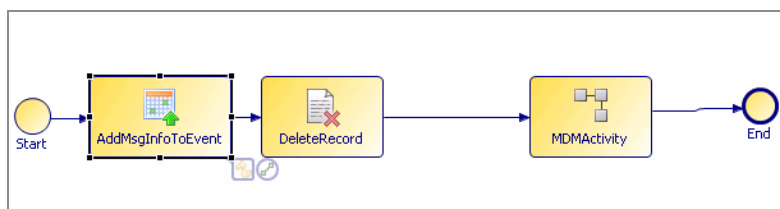
We will do the following in this section:

- [Adding DeleteRecord activity for Conditional Transition](#)
  - [Define Input Parameters to the DeleteRecord Activity](#)
  - [Define Output Parameters to the DeleteRecord Activity](#)
- [Adding SaveRecord activity for Conditional Transition](#)
  - [Define Input Parameters to the SaveRecord Activity](#)
  - [Define Output Parameters to the SaveRecord Activity](#)

## Adding DeleteRecord activity for Conditional Transition

### Procedure

1. Add a **DeleteRecord** activity to your process.
2. Add this activity to the right of **AddMsgInfoToEvent**.



## Define Input Parameters to the DeleteRecord Activity

[Setting InDocument Parameter](#)

[Setting MasterCatalog Parameter](#)

[Setting ProductIds Parameter](#)

[Setting RecordKey Parameter](#)

[Setting VersionOption Parameter](#)

[Setting eventState Parameter](#)

## Setting InDocument Parameter

### Procedure

1. Select inDoc variable in **Input To Activity** tab.
2. Drag and drop InDoc global variable on InDocument input parameter.

## Setting MasterCatalog Parameter

### Procedure

1. Select the **MasterCatalog** parameter.
2. Click on the text field corresponding to the MasterCatalog parameter and type the value **ASSEST**.

## Setting ProductIds Parameter

### Procedure

1. Drag InDoc global variable and drop it on **ProductIds** input parameter.
2. Double click on **ProductIds** Parameter.
3. Select **Xpath** from the **Eval** drop-down list.
4. Add the following expression in **Source expression** text field  
\$InDoc/Body/Document/BusinessDocument/CatalogAction/CatalogActionHeader/CatalogReference/RevisionID/DBID.

## Setting RecordKey Parameter

### Procedure

1. Drag InDoc global variable and drop it on **RecordKey** input parameter.
2. Double click on **RecordKey** Parameter.
3. Select **Xpath** from the **Eval** drop-down list.
4. Add the following expression in **Source expression** text field  
\$InDoc/Body/Document/BusinessDocument/CatalogAction/CatalogActionHeader/Supplier/Partner/Name

## Setting VersionOption Parameter

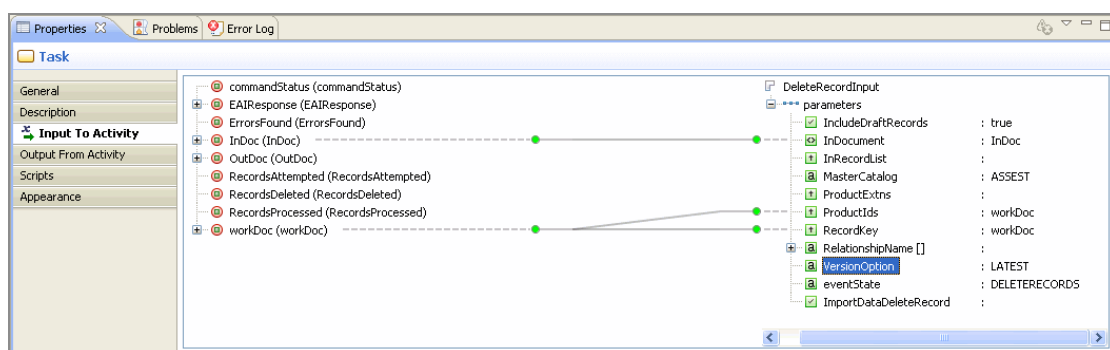
### Procedure

1. Select the **VersionOption** parameter.
2. Click corresponding to the VersionOption parameter and select **LATEST** from the drop-down list.

## Setting eventState Parameter

### Procedure

1. Select the **eventState** parameter.
2. Click corresponding to the eventState parameter and enter the value **DELETERECORDS** in the text field.



## Define Output Parameters to the DeleteRecord Activity

[Setting RecordsDeleted Parameter](#)

[Setting RecordsProcessed Parameter](#)

[Adding SaveRecord activity for Conditional Transition](#)

### Setting RecordsDeleted Parameter

#### Procedure

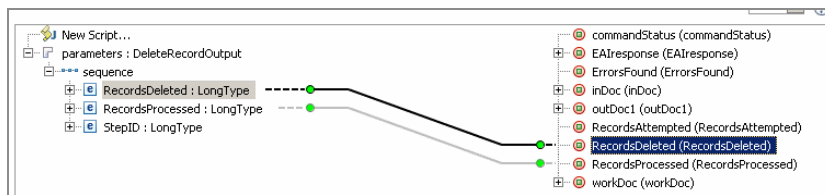
1. Select the **RecordsDeleted** output parameter on the left.
2. Drag it onto the **RecordsDeleted** local variable on the right.

### Setting RecordsProcessed Parameter

#### Procedure

1. Select the **RecordsProcessed** output parameter on the left.
2. Drag it onto the **RecordsProcessed** local variable on the right.

#### Result

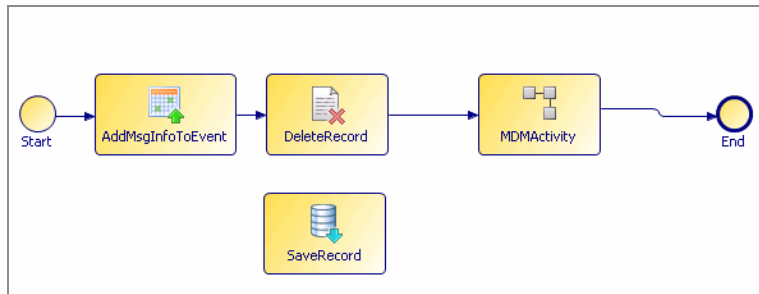


## Adding SaveRecord activity for Conditional Transition

#### Procedure

1. Add a **SaveRecord** activity in your process.
2. Add this activity below **DeleteRecord**.





## Define Input Parameters to the SaveRecord Activity

[Setting ErrorSeverity Parameter](#)

[Setting VersionOption Parameter](#)

### Setting ErrorSeverity Parameter

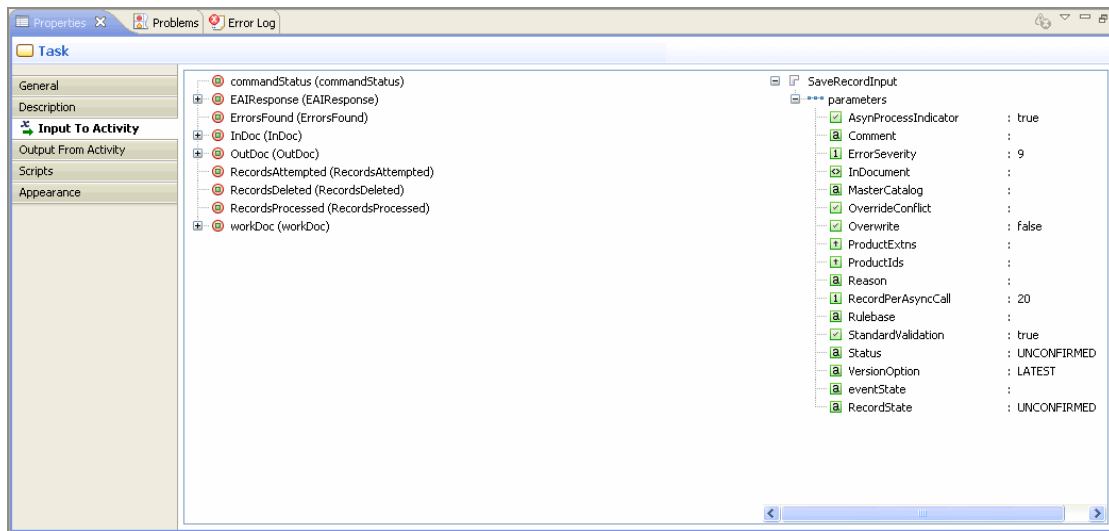
#### Procedure

1. Select the **ErrorSeverity** parameter.
2. Click the text field corresponding to the **ErrorSeverity** parameter and type the value 9.

### Setting VersionOption Parameter

#### Procedure

1. Select the **VersionOption** parameter.
2. Click corresponding to the **VersionOption** parameter and select **LATEST** from the drop-down list.



## Define Output Parameters to the SaveRecord Activity

[Setting RecordsDeleted Parameter](#)

[Setting RecordsAttempted Parameter](#)

[Setting WorkDoc Parameter](#)

[Adding Condition Logic](#)

## Setting RecordsDeleted Parameter

### Procedure

1. Select the **RecordsWithErrors** output parameter on the left.
2. Drag it onto the **ErrorsFound** local variable on the right.

## Setting RecordsAttempted Parameter

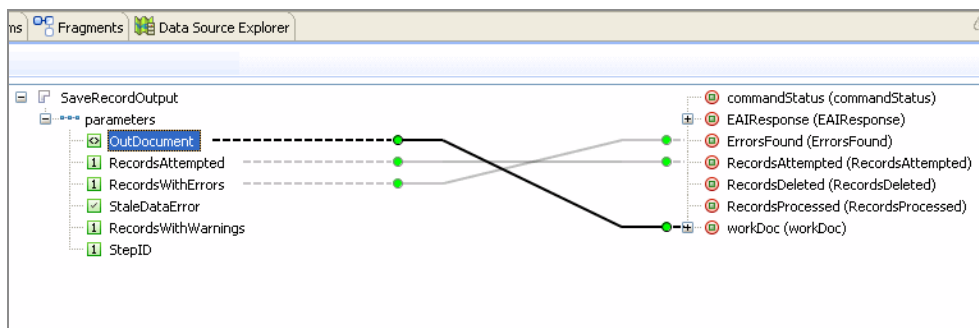
### Procedure

1. Select the **RecordsAttempted** output parameter on the left.
2. Drag it onto the **RecordsAttempted** local variable on the right.

## Setting WorkDoc Parameter

### Procedure

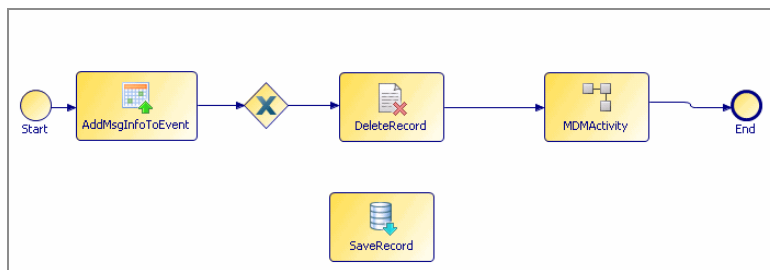
1. Select the **OutDocument** output parameter on the left.
2. Drag it onto the **workDoc** local variable on the right.



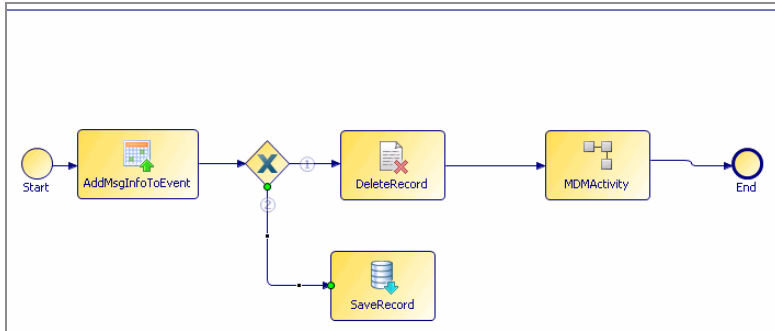
## Adding Condition Logic

### Procedure

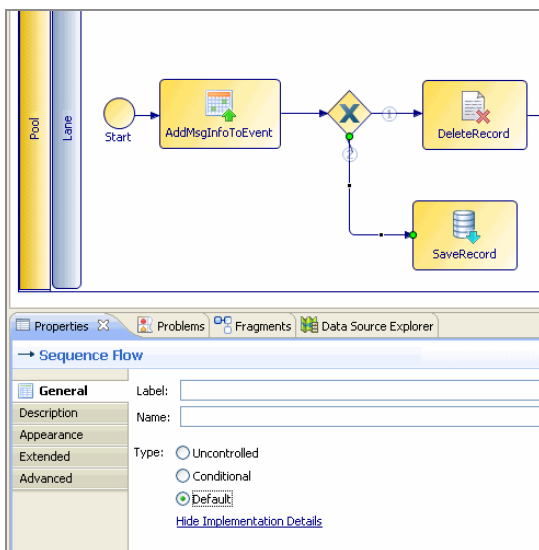
1. On the Palette, click **XOR Data Gateway** and place your cursor on the sequence flow between the **AddMsgInfoToEvent** and **DeleteRecord**.



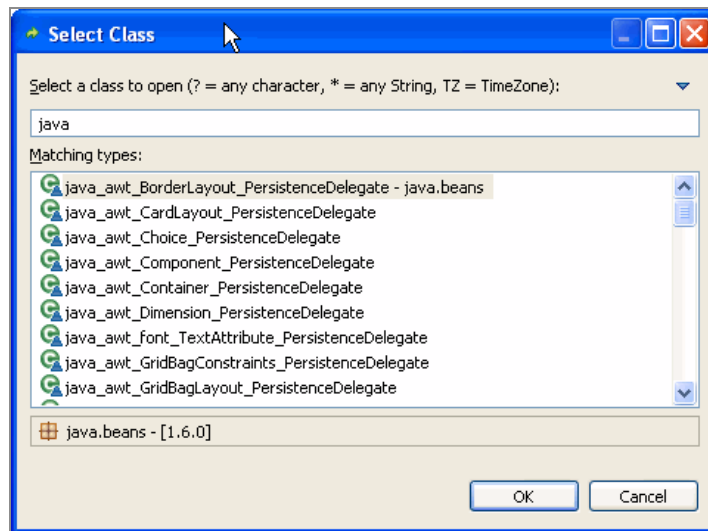
2. Add a Sequence flow from the **XOR Gateway** to the Save Record (by selecting the Sequence Flow tool under connections in the Palette).



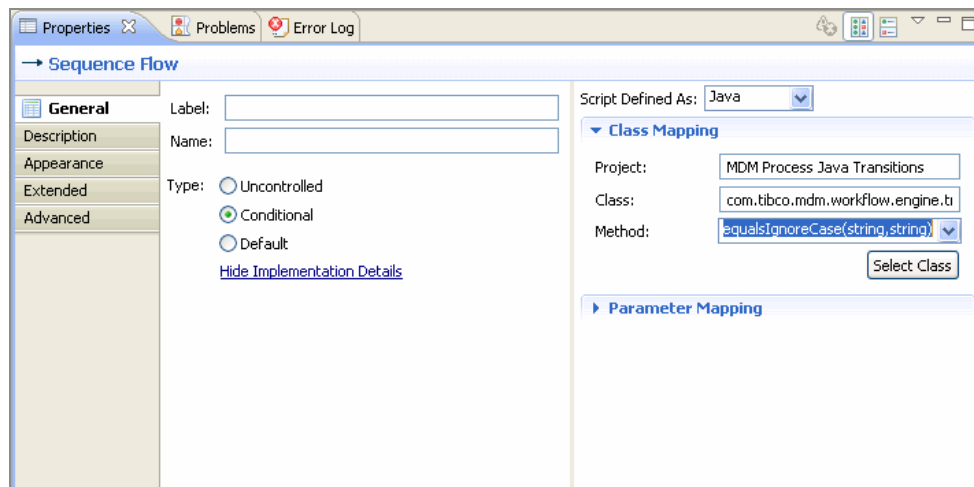
3. Mark the transition from the XOR gateway to **SaveRecord** as default by selecting the transition and then selecting **Default** in the **Properties** Windows, **Sequence Flow**, **General** tab. The transition to the **SaveRecord** task will then get a small slash to indicate it is default.



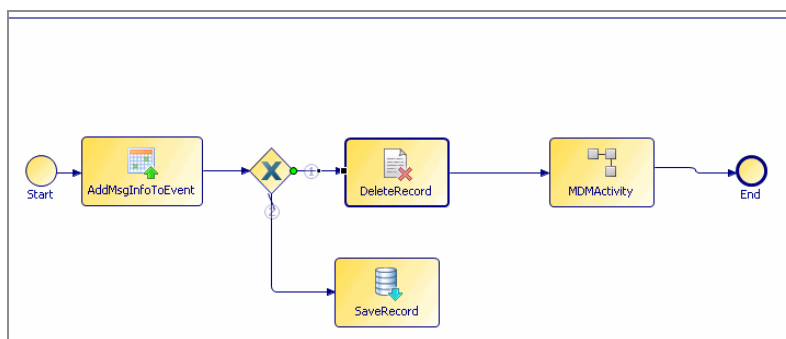
4. Click the sequence flow to the **DeleteRecord** task.
  - a. In the **Properties** Window, **General** tab, set the Sequence Flow Type to **Conditional**.
  - b. Set the **Script Defined As** to **Java**.
  - c. Click **Select Class** and select a java class and click **OK**.



d. Select the method from the Method drop-down list.



**Look of the workflow at this point**



## Add More Activities

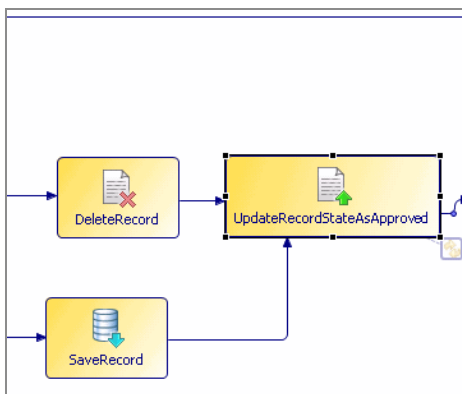
Next, we add activities for the rest of the process (after deletion or save of the record).

- Add an activity to set the record state to approved (**UpdateRecordState**).
- Add an activity to publish to EAI (**SendProtocolMessage**).
- Set the status to success (**UpdateEvent**).

## UpdateRecordState Activity

### Procedure

1. Add an **UpdateRecordState** activity (after the DeleteRecord activity).
2. Ensure that both the **DeleteRecord** and **SaveRecord** activities are connected to the **UpdateRecordState** activity (use the Sequence tool to make the required connections).
3. Change the label of the **UpdateRecordState** activity to **UpdateRecordStateAsApproved**.



## Define Input Parameters to the UpdateRecordState activity

[Setting Status Parameter](#)

[Setting InDocument Parameter](#)

## Setting Status Parameter

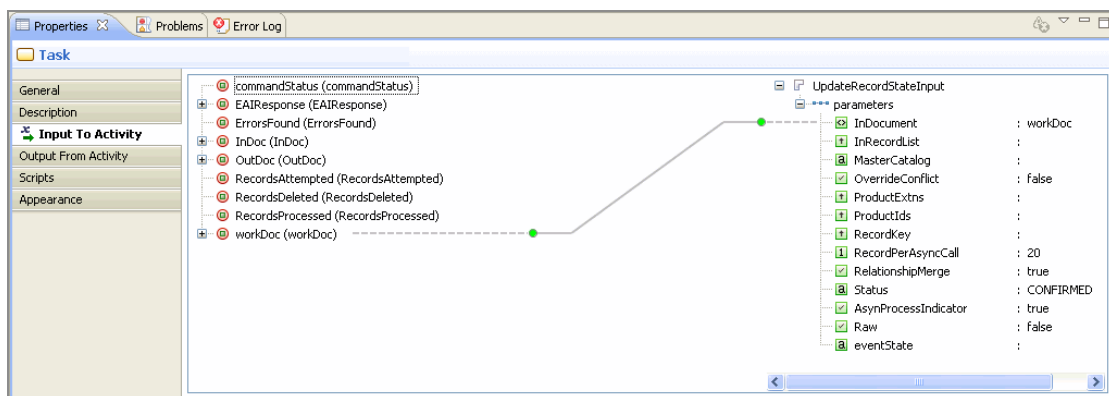
### Procedure

1. Select the **Status** parameter.
2. Click corresponding to the Status parameter and select **CONFIRMED** from the dropdown list.

## Setting InDocument Parameter

### Procedure

1. Select the inDoc variable in the **Input To Activity** tab.
2. Drag and drop the workdDoc local variable on the InDocument input parameter.



## Define Output Parameters to the UpdateRecordState activity

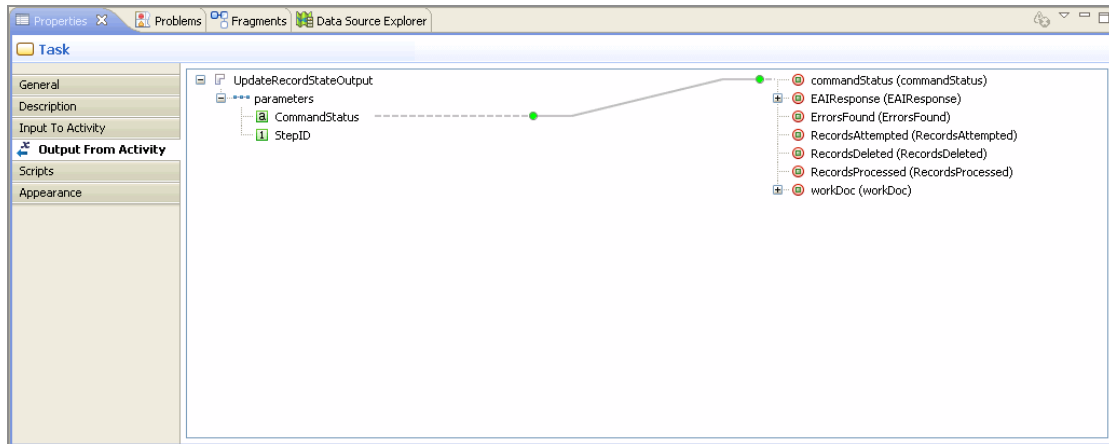
[Setting commandStatus Parameter](#)

## Setting commandStatus Parameter

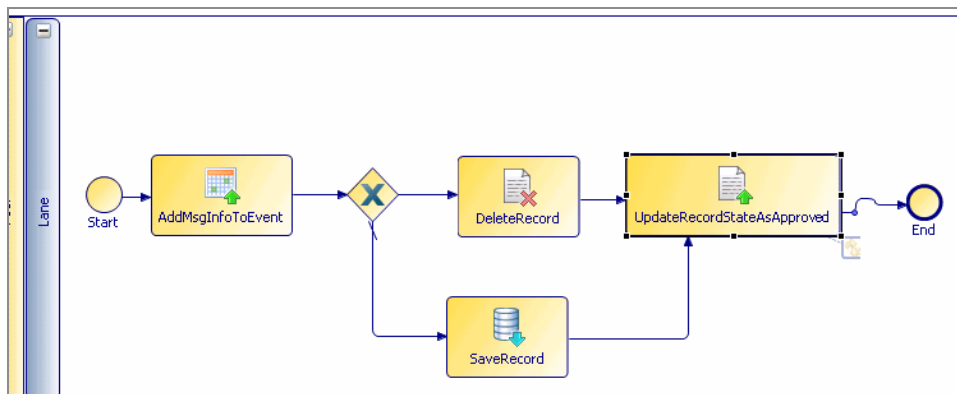
### Procedure

1. Select the **commandStatus** local variable on the right.

2. Drag it onto the **CommandStatus** output parameter on the left.



Look of the workflow at this point

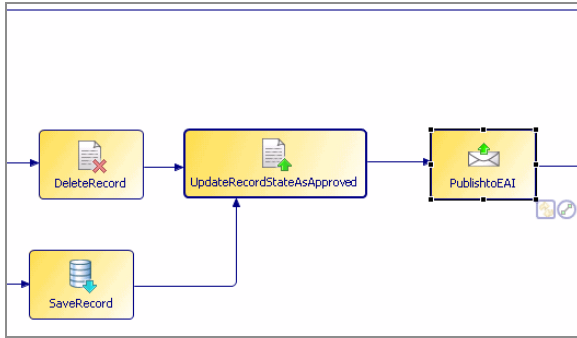


## Setting SendProtocolMessage Activity

### Procedure

1. Add a **SendProtocolMessage** activity (after the **UpdateRecordState** activity).
2. Change the Name of the **SendProtocolMessage** activity to **PublishToEAI**.





## Define Input parameters to the SendProtocolMessage activity

[Setting BizProtocol Parameter](#)

[Setting ChannelCredential Parameter](#)

[Setting ExpiryDate Parameter](#)

[Setting ExpiryType Parameter](#)

[Setting InDocument Parameter](#)

[Setting MessageID Parameter](#)

[Setting PayloadPackingScheme Parameter](#)

[Setting SenderCredential Parameter](#)

[Setting DefaultDomain Parameter](#)

[Setting eventState Parameter](#)

## Setting BizProtocol Parameter

### Procedure

1. Select the **BizProtocol** parameter.
2. Click corresponding to the BizProtocol parameter and select **JMS** from the drop-down list.

## Setting ChannelCredential Parameter

### Procedure

1. Select the **ChannelCredential** parameter.
2. Click corresponding to the ChannelCredential parameter and enter the value 0065063183367 in the text field.

## Setting ExpiryDate Parameter

### Procedure

1. Select the **ExpiryDate** parameter.
2. Click corresponding to the ExpiryDate parameter and enter the value **0:0:5:0** in the text field.

## Setting ExpiryType Parameter

### Procedure

1. Select the **ExpiryType** parameter.
2. Click corresponding to the ExpiryType parameter and select **RELATIVE** from the drop-down list.

## Setting InDocument Parameter

### Procedure

1. Select inDoc variable in **Input To Activity** tab.
2. Drag and drop workDoc local variable on InDocument input parameter.

## Setting MessageID Parameter

### Procedure

1. Drag workDoc local variable and drop it on **MessageID** input parameter.
2. Double click on **MessageID** Parameter.
3. Select **Xpath** from the **Eval** drop-down list.
4. Add the following expression in **Source expression** text field  
`$workDoc/externalControlNumber.`

## Setting PayloadPackingScheme Parameter

### Procedure

1. Select the **PayloadPackingScheme** parameter.
2. Click on the text field corresponding to the PayloadPackingScheme parameter and type the value **STANDARD\_INTEGRATION**.

## Setting SenderCredential Parameter

### Procedure

1. Select the **SenderCredential** parameter.
2. Click corresponding to the SenderCredential parameter and enter the value **0065063183367** in the text field.

## Setting DefaultDomain Parameter

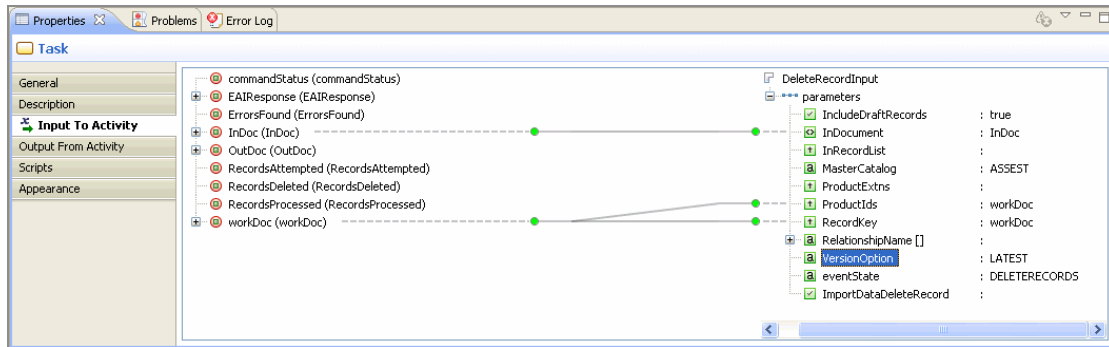
### Procedure

1. Select the **DefaultDomain** parameter.
2. Click the text field corresponding to the DefaultDomain parameter and type the value **GLN**.

## Setting eventState Parameter

### Procedure

1. Select the **eventState** parameter.
2. Click corresponding to the eventState parameter and enter the value **DELETERECORDS** in the text field.



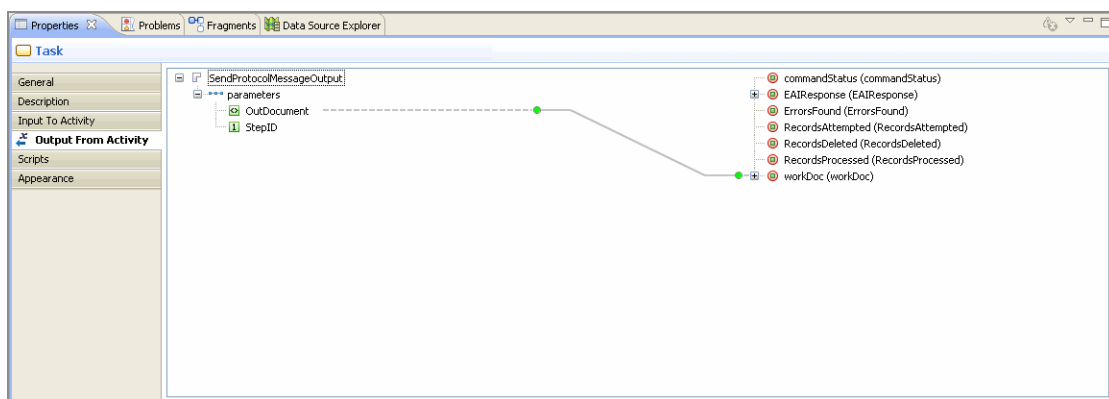
## Define Output Parameters to the SendProtocolMessage Activity

### Setting OutDocument Parameter

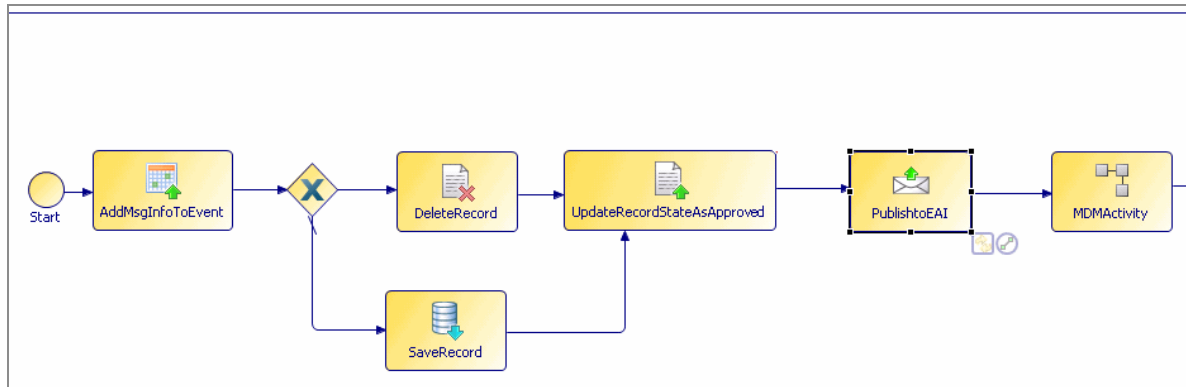
## Setting OutDocument Parameter

### Procedure

1. Select the **Outdocument** parameter on the left.
2. Drag it onto the **workDoc** variable on the right.



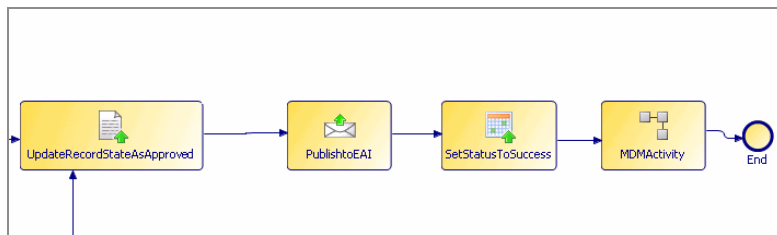
## Look of the workflow at this point



## UpdateEvent Activity

### Procedure

1. Add a **UpdateEvent** activity (after the **SendProtocolMessage** activity).
2. Change the label of the **UpdateEvent** activity to **SetStatusToSuccess**.



## Add Input Parameters to the UpdateEvent Activity

[Setting eventState Parameter](#)

[Setting eventStatus Parameter](#)

## Setting eventState Parameter

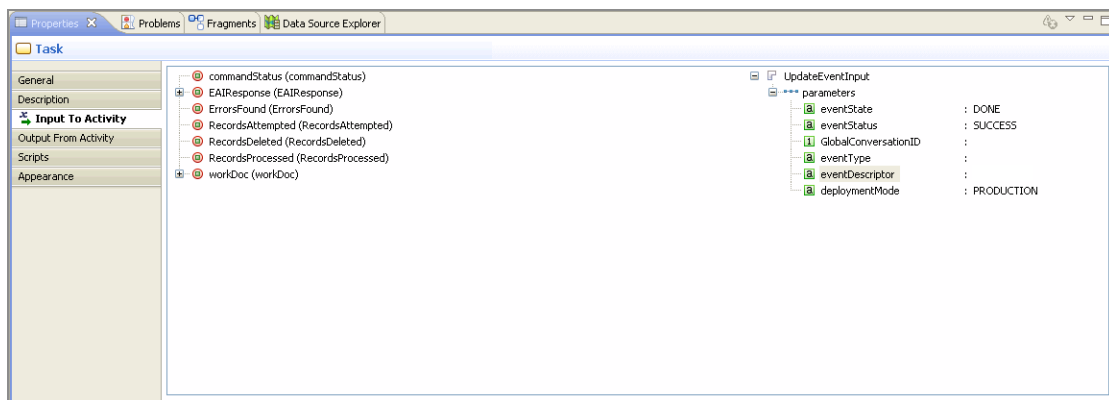
### Procedure

1. Select the **eventState** parameter.
2. Click the text field corresponding to the eventState parameter and type the value **DONE**.

## Setting eventStatus Parameter

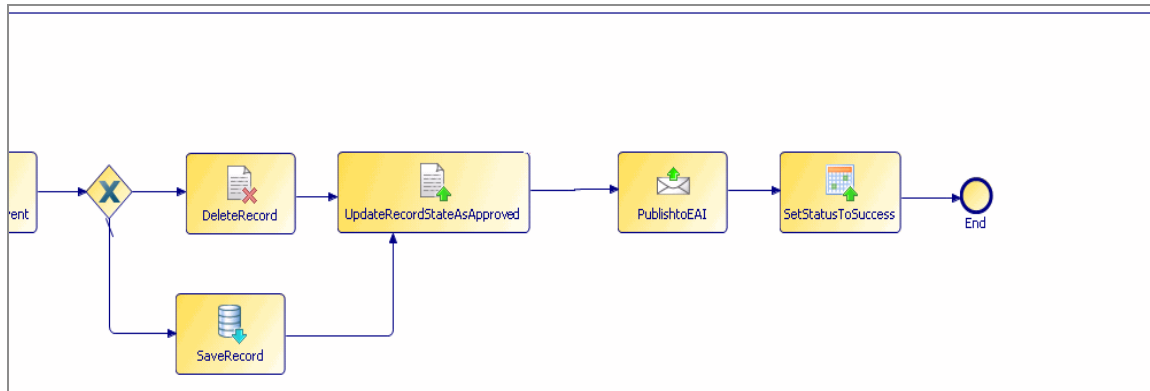
### Procedure

1. Select the **eventStatus** parameter.
2. Click the text field corresponding to the eventStatus parameter and type the value **SUCCESS**.



**Note:** Ensure that you delete the default MDM Activity from the process.

## Look of your workflow at this point



## Validation

The process now needs to be validated to ensure that there are no errors.

The MDM validation plug-in runs in the background as the user modifies or edits the process; it checks that changes are made in accordance with the Validation rules.

Validations are triggered when the destination of the Process is set as **MDM**.

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

The following validations are performed to ensure conformity to MDM standards:

- **All Activity Tasks to Web Service:** All Activity types should be set to Web Service. Usage of any other activity type will result in an error.
- **All Service Tasks to CIM Service:** CIM Service should be chosen for all Service Tasks. No other services are supported.
- **Single starting point for process** The process should have only a single starting point. Activities which do not have an incoming transitions are considered a starting point.
- **Unique Activity Names:** Each activity should have a unique name, which is mapped to the MDM activity name.
- **Only XOR and Parallel Gateways:** Gateway Type should be **XOR** (Exclusive Decision/Merge Data or Event based) or **AND** (Parallel Fork/Join). No other Gateway Types are supported.

- **Properly defined parameters:** Each global Process parameter and data field must have the attributes Direction, Eval, and Type properly and sufficiently defined.
- **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.
- **Intermediate Event Restrictions:** Intermediate events are restricted to exception, timeout, cancel, and Link Event. Use of any other type will result in an error.

**Note:** These are some of the validations; for a complete list of validations, see the TIBCO MDM Studio *Process Designer User Guide*.

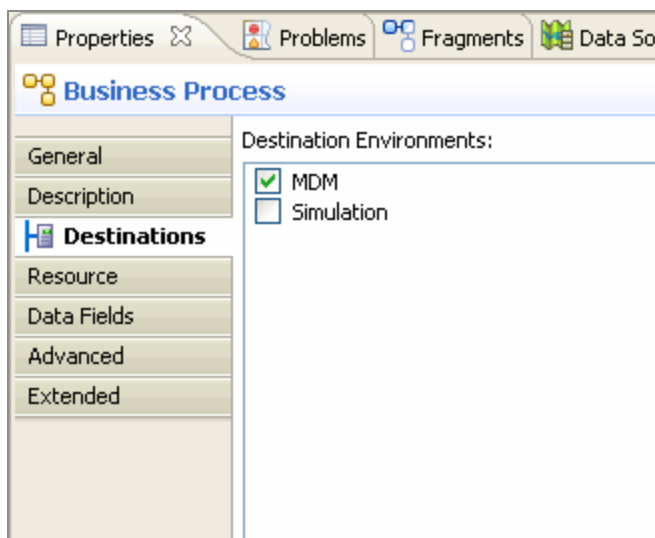
## Exporting the Process to a File

Now that the MDM Workflow has been defined and edited, it can be exported in MDM workflow format.

Follow these steps to export the process:

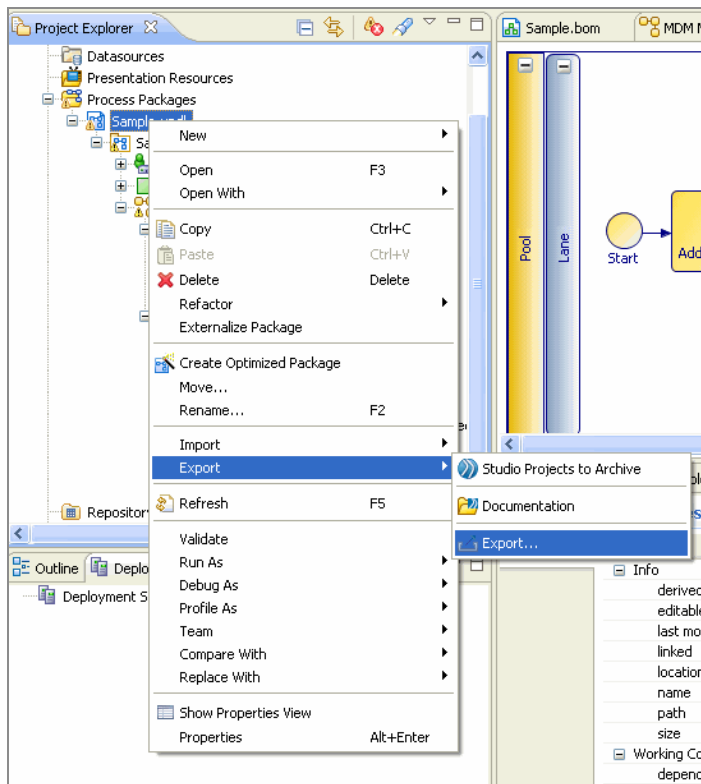
### Procedure

1. First, ensure that the **Destination** for the Process is set to **MDM**.

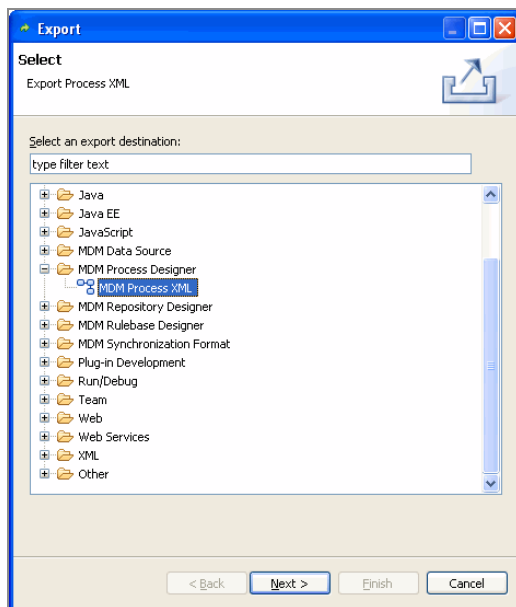


2. Right-click the sample (.xpd1 file) in the Project Explorer and select the **Export** option.

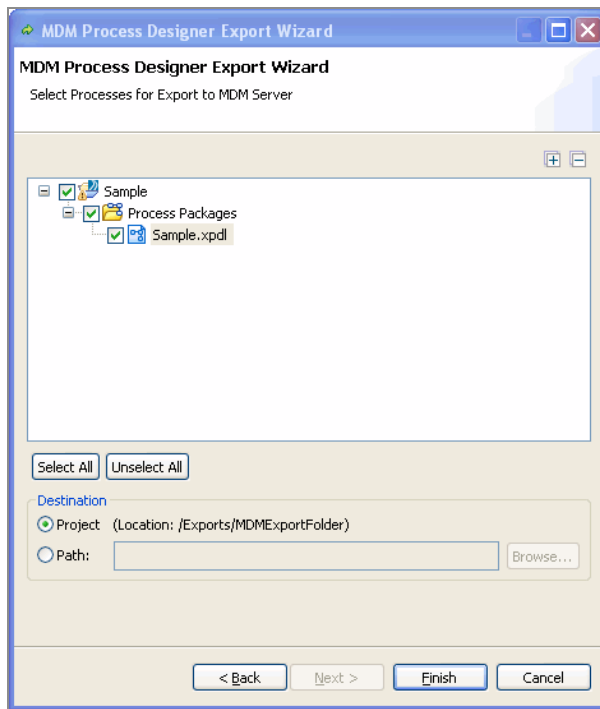




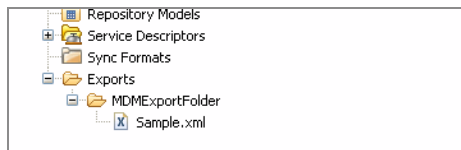
3. Select **MDM Process XML** under **MDM Process Designer** and click **Next**.



4. Select the location to export the sample (xpd1) to. You can either export it to the default projects export folder, or to a file. Click **Finish**.



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 MDM Engine `xml` format.



## Deploy the Process Model

The created Process model can be directly deployed to MDM.

For deployment steps, see the **Import, Export, and Deploying Processes** chapter in the TIBCO MDM Studio *Process Designer User Guide*.

# 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

---

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, BusinessConnect, ActiveMatrix BusinessWorks, and Enterprise Message Service are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

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>.

Copyright © 2007-2024. Cloud Software Group, Inc. All Rights Reserved.