# Adeptia Suite 5.0

## Evaluation Guide

**Release Date January 20, 2010**

Adeptia Inc.
443 North Clark Ave,
Suite 350
Chicago, IL 60654, USA
Phone: (312) 229-1727

## Copyright



## Trademarks

The Adeptia ™ logo is a trademark of Adeptia, Inc.

## Statement of Conditions

Adeptia, Inc. provides this publication "as is" without warranty of any kind, either express or implied. In no event shall Adeptia be liable for any loss of profits, loss of business, loss of use or data, interruption of business, or for indirect, special, punitive, incidental, or consequential damages of any kind.

No part of this work covered by copyright herein may be reproduced in any form or by any means—graphic, electronic or mechanical—including photocopying, recording, taping, or storage in an information retrieval system, without prior written permission of the copyright owner.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, contact **www.adeptia.com**.

## Document Conventions

| Convention | Description |
|---|---|
| Text Matter in font Verdana and font size 9 point. | Explains the evaluation guide. |
| Text matter | Click link to reach target. |
|  | Note: |

## Abbreviations Used

| Abbreviation | Description |
|---|---|
| JMS | Java Messaging Service |
| CDO | Collaboration Data Object |

# Contact Information

In case of any queries, please contact us at:

| Contact For | Email ID |
|---|---|
| Sales | sales@adeptia.com |
| Support | support@adeptia.com |

For latest updates and information, please visit us at **www.adeptia.com**

# Table of Contents

# 1 ABOUT THIS DOCUMENT

Adeptia Suite is shipped with pre-built Process Flows that are based on real business scenarios. This Evaluation Guide describes how to create and run these Process Flows. It provides an overview of these Process Flows, the activities that comprise these Process Flows and the steps describing how to execute these Process Flows.

**Pre-Requisite**

It is assumed at this point that you have read the **Getting Started Guide** and logged into the Adeptia Suite application.

This document is divided into the following sections:

- **Chapter 1** provides an overview of Adeptia Suite Evaluation Guide.
- **Chapter 2** gives information about the target audience.
- **Chapter 3** describes the *Data Transformation* Process Flow that uses different mapping functions to transform source data into the required format. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.
- **Chapter 4** describes the *Scripted Service* Process Flow that demonstrates the use of a scripted service and generation of fDynamic File Name. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.
- **Chapter 5** describes the *Process Designer* Process Flow that demonstrates the use of different features of Process Designer. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.
- **Chapter 6** describes the *Process Flow to Process Excel Data* that demonstrates the use of different features of process designer and complex mapping functions. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.
- **Chapter 7** describes the *JMS Event Driven* Process Flow that demonstrates the use of JMS event for triggering a process flow. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.
- **Chapter 8** describes the *Record to Record Service* Process Flow that is used to process data of a positional file. This process flow is pre-created and is shipped with the Adeptia Suite. This chapter is divided into various sections that describe the Purpose, Services used, Data Description, Pre-requisites, and the Execution Steps separately.

Before using the evaluation guide, download and install the Adeptia Suite from our website **www.adeptia.com.** To know how to install Adeptia Suite, refer to the **Installation Guide**.

# 2 TARGET AUDIENCE

This document is intended for the users who are evaluating Adeptia Suite and will execute the pre-bundled process flows. It is recommended that you should first read the Getting Started Guide, before reading this guide.

# 3 DATA TRANSFORMATION PROCESS FLOW

This section describes the Data Transformation Process Flow.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This sample Process Flow demonstrates the use of different mapping function. This Process Flow is used to extract data (Insurance Policy information) from a database source and convert it into an excel format. Conversion of data from database specific format into excel format is done using different mapping functions. After conversion, data is written in an excel file and that excel file is saved into the specified location. At the end of the Process Flow, a native call is used to execute a batch file, which creates a log file. This log file contains the details of the excel file, which is created by the process flow.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- Database Driver and Database Info
- Database Schema
- Database Source
- Different Mapping function
- Excel Schema
- File Target
- Native Call

## DESCRIPTION

This sample Process Flow can be outlined as below:

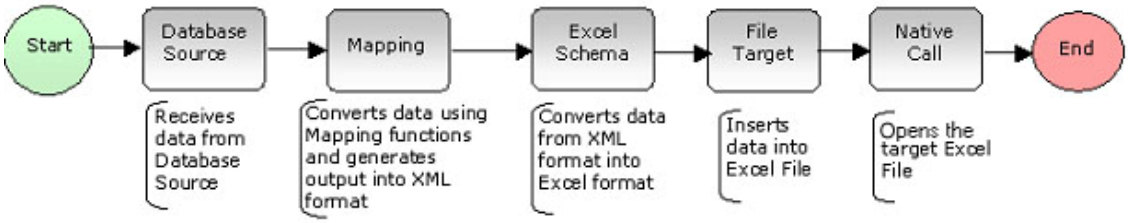



Figure 3.1: Data Transformation Process Flow

### Database Source (EvalXform_Source)

Database Source is used to retrieve the data from the database. This contains insurance data for policyholders. Database Source uses Database Schema (EvalXform_DBSchema).

### Database Schema (EvalXform_Schema)

Database Schema is used to parse data received from the database and converts it into an intermediate XML format. Database Schema uses Database Driver (EvalXform_DBDriver) and Database Info (EvalXform_DBInfo) to connect to Database Server.

### Mapping (EvalXform_Mapping)

Mapping is used to map the data fields of the source data to data fields of the target data. In this Process Flow different mapping functions e.g. Math, String, aggregation and Custom Method are used to manipulate the data.

### Excel Schema (EvalXform_ExcelSchema)

Excel Schema is used to convert data from an intermediate XML format to Excel format.

### File Target (EvalXform_FileTarget)

File Target is used to specify the name of the target Excel file and the path, where the target excel file is saved.

### Native Call (EvalXform_NativeCall)

Native call is used to run native program (e.g. .exe or bat on windows) on the server where Adeptia Suite is running. In this Process Flow, native call executes a batch program (if Adeptia Suite is running on windows) or a shell program (if Adeptia Suite is running on Linux). This program creates a log file which contains the details of the excel file created by the Process Flow.

## USAGE SCENARIO

This sample Process Flow can be used whenever you want to transform the data using different mapping functions.

## DATA DESCRIPTION AND MAPPING INFORMATION

Data used in this sample Process Flow contains information of insurance policyholders e.g. Name and address of insured person, Premium amount and policy expiration date etc.

The structure of database table used as source is outlined in the table below.

Table 3.1: Structure of Database Table used as Source

| Field Name | Description | Data Type |
|---|---|---|
| PolicyNumber | Unique number for each policy | Number |
| NameInsured | Name of the insured person | String |
| Address | Address of the insured person | String |
| DOB | Date of Birth of insured Person | Date |
| TelephoneNo | Contact no. of the insured person | Number |
| ExpiryDate | Expiry date of the policy | Date |
| PremiumMedical | Premium amount for medical insurance | Number |
| PremiumDental | Premium amount for dental insurance | Number |
| SecurityCode | Secret Code used to verify the insured person | Number |

The structure of Excel File used as target and mapping information is outlined in the table below.

Table 3.2: Structure of Excel Table used as Target

| Field Name | Description | Data Type |
|---|---|---|
| SerialNo. | Serial Number of the source record. Position function is used to calculate the Serial Number. | Number |
| PolicyNumber | Unique number for each policy | Number |
| NameInsured | Name and address of the insured person, Concat function is used between NameInsured and Address field of source | String |
| Age | Age of the insured person, Custom Method is used to call a java class, which calculates age based on DOB of insured person | Number |
| TelephoneNo | Contact no. of the insured person | String |
| ExpiryDate | Expiry date of the policy, record of already expired policy is not shown. | String |
| Premium | Sum of PremiumMedical and PremiumDental. Math function is used to add PremiumMedical and PremiumDental of source record | Number |

# PREREQUISITES

- Data records must be present in database server used as source.

To know, how to populate the records into your database, refer to **Populating Data into Database** section.

- Before executing this process flow, you must edit the following activities to point to the database which is used as source:
- EvalXform_DBDriver
- EvalXform_DBInfo

To know, how to edit these activities refer to **Editing Activities** section.

## POPULATING DATA INTO DATABASE

In this process flow, a database table is used as source. A SQL script is provided with Adeptia Suite to create a table in your database and populate data (used as source) into this table. This SQL script is located in ***../AdeptiaServer-5.0/Serverkernel/Solutions/Demo/EvalXform*** folder. To create table and populate data into your source database, you need to run the respective SQL Script, using the database client application.

## EXECUTING AND MONITORING

This section describes the execution of sample Process Flow and monitoring Process Flow execution.

***Steps to execute the Process Flow***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed.
3. Select the radio button adjacent to *EvalXform_ProcessFlow* and then click **Execute** link. The View Process Flow Log screen is displayed (see Figure 3.2).

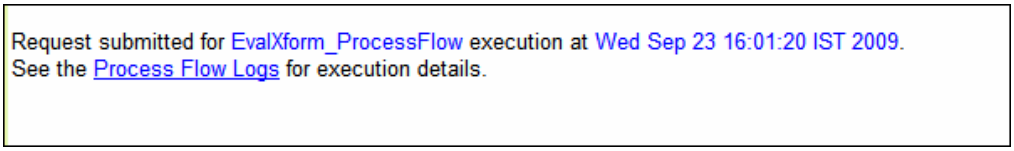



Figure 3.2: View Process Flow Monitor

4. Click the link **Process Flow Logs** to view the status of the Process Flow execution. The Process Flow Logs is displayed (see Figure 3.3).

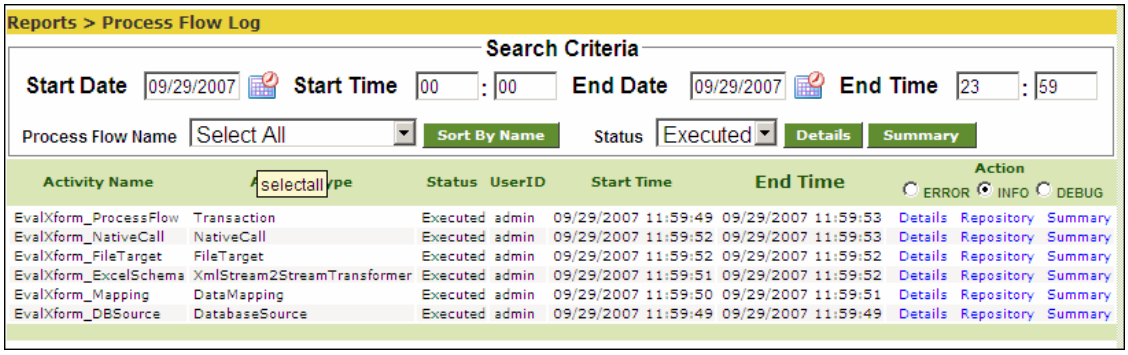



Figure 3.3: Process Flow Logs

To view the summary of all instances of the process flow execution, click the **Summary** button.

5. Click **Details** under the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 3.4).

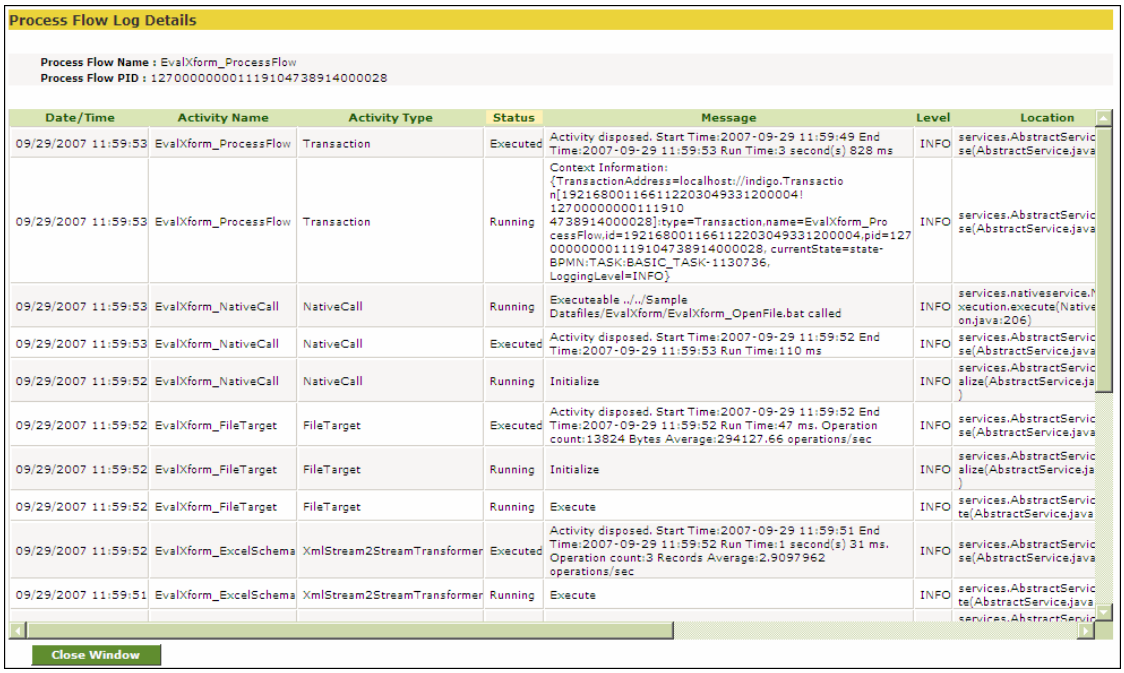



Figure 3.4: Process Flow Log Details

6. After the process flow is executed successfully, you can view the log file (EvalXform_OpenFile.log), which is created by the Native call activity.
7. This log file created in **`../AdeptiaServer-5.0/Serverkernel/Solutions/Demo/EvalXform`** folder on server, where Adeptia Suite is running.

# EDITING ACTIVITIES

Activities used in this sample Process Flow are pre-created. This section describes the process of editing these activities.

## Editing Database Driver

## (EvalXform_DBDriver)

Database Driver is used to specify the type of database and to upload driver jar files required to connect to that database.

***Steps to edit Database Driver***

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.

2. Click **Database Driver**. The Manage Database Driver screen is displayed with the list of existing database drivers.
3. Select the radio button adjacent to *EvalXform_DBDriver* activity and then click **Edit** link. This displays the Edit *EvalXform_DBDriver* activity, with the properties of the activity displayed in their respective fields (see Figure 3.5).

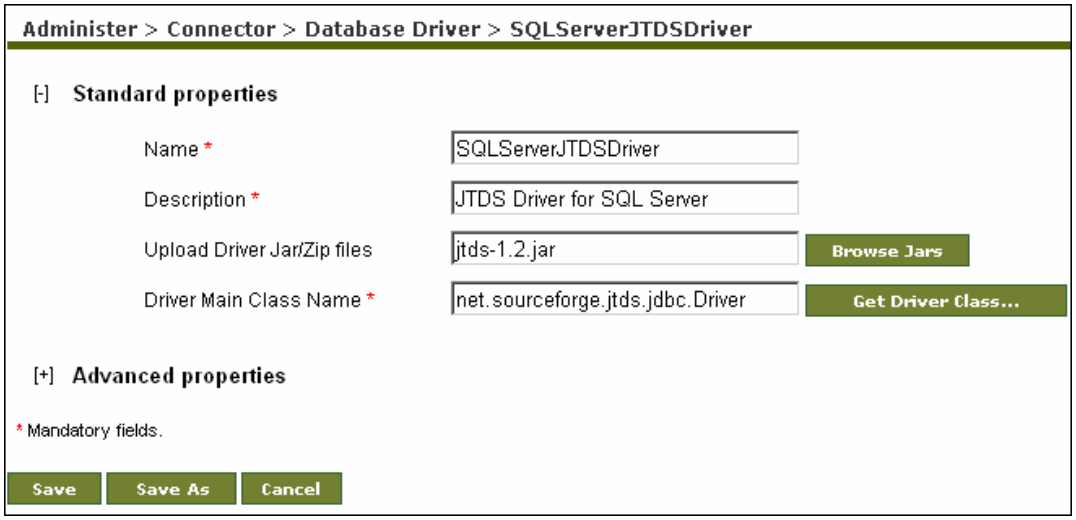



Figure 3.5: Edit *EvalXForm_DBDriver*

A detailed description of fields on this screen is explicated in the table below.

Table 3.3: Details of Fields on Edit Database Driver Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Driver |
| Description | Description of the Database Driver |
| Upload Driver Jar Files | JDBC Driver files, which are used to connect Database Server. Click the **Browse Jars** button to select Jar files. Following is the list of databases and the required Jar files:<br>Oracle — Classes12.jar<br>IBM DB2 (Ver 7.1) — db2java.zip (7.1 version)<br>IBM DB2 (Ver 8.1) — db2jcc.jar<br>JTDS- SQL Server — Jtds.jar<br>HSQL DB — hsqldb-1.8.0.7.jar |

| | | |
|---|---|---|
| Driver Main Class Name | Driver Main Class Name is a fully qualified java class name for the main database driver class. The driver class name typically starts with a com., net. or org. followed by the company domain, for example the JDBC driver class from mysql.com is called com.mysql.jdbc.Driver. Click the **Help** button to select Driver Main Class Name from the drop-down list. Following is the list of Driver Main Class Name of different databases: | |
| | Oracle | oracle.jdbc.driver.OracleDriver |
| | IBM DB2 (Ver 7.1) | COM.ibm.db2.jdbc.net.DB2Driver |
| | IBM DB2 (Ver 8.1) | com.ibm.db2.jcc.DB2Driver |
| | JTDS-SQL Server | net.sourceforge.jtds.jdbc.Driver |
| | HSQLDB | org.hsqldb.jdbcDriver |
| | MS Access | sun.jdbc.odbc.JdbcOdbcDriver |
| | MS Excel | sun.jdbc.odbc.JdbcOdbcDriver |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Driver has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database driver (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the database driver has been updated successfully.

## Editing Database Info

## (EvalXform_DBInfo)

Database Info activity is used to specify the JDBC URL or Server URL and the Username and Password to access the database. Database Info uses Database Info to connect to specified Database Server.

***Steps to edit the Database Information***

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **Database Info**. The Manage Database Info screen is displayed with the list of existing Database Info.
3. Select the radio button adjacent to *EvalXform_DBInfo* activity and then click **Edit** link. This displays the Edit *EvalXform_DBInfo* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.6).

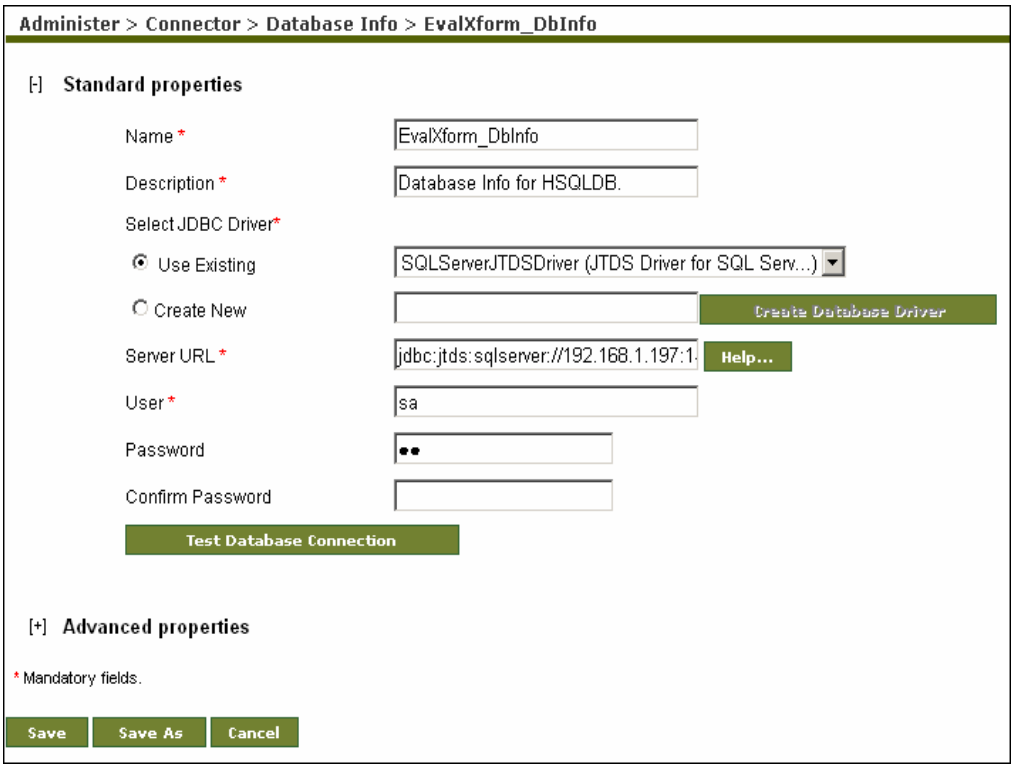



Figure 3.6: Edit *EvalXForm_DBInfo*

A detailed description of fields on this screen is explicated in the table below.

Table 3.4: Details of Fields on Edit Database Info Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Info |
| Description | Description of the Database Info |
| JDBC Driver | Database Driver is created to connect to the database Server. For more details refer to section **Editing Database Driver**. You can use an existing driver or create a new database driver. |
| Server URL | Server URL or JDBC URL points to a specific database on a specified database server. There is no standard for JDBC URL. Every JDBC driver uses a slightly different syntax. For Example a JDBC URL for a MySQL database using the `com.mysql.jdbc.Driver` direct from MySQL might look like this: `jdbc:mysql://localhost/databaseName`. To specify Server URL, Click the **Help** button and enter the following information: |

| | |
|---|---|
| Database Type | Type of the database |
| Host Name | Name of the server on which database server is running |
| Port | Port at which database server is running |
| Database Name | Name of the database |

| | |
|---|---|
| | Following is the list of Server URL's of different databases: |
| | Oracle: jdbc:oracle:thin:@databaseserver:1521:orcl |
| | IBM DB2 (Ver 7.1): jdbc:db2://databaseserver:6789/TOOLSDB |
| | IBM DB2 (Ver 8.1): jdbc:db2://databaseserver:50000/TOOLSDB |
| | SQL JTDS: jdbc:jtds:sqlserver://databaseserver:1433/master |
| | MS Access: jdbc:odbc:Driver={MicroSoft Access Driver (*.mdb)}; DBQ=c:/test/db1.mdb |
| | MS Excel: Jdbc:odbc:*ExcelJDBCTest*<br>where *ExcelJDBCTest* is the ODBC object that you need to create using DSN. |
| | HSQL DB: jdbc:hsqldb:hsql://databaseserver:2476 |
| | Here **database server** is the name of the server where database is running. |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Info has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Database Info (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the Database Info has been updated successfully.

## Editing Database Schema

## (EvalXform_DBSchema)

Database Schema defines the structure of the database table. A Database Schema is used to define how records can be read from a database table or can be written into a database table. In this sample process Database Schema is being used at the source end. At the source end it converts data from database specific format into an intermediate XML format. Database Schema uses Database Info activity to connect to the database Server. It should be noted that the Database Schema does not directly take part in creation of Process Flow. It is used by the Database Source activity and the Database Source activity is used in the Process Flow.

***Steps to edit the Database Schema activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Database**. The Manage Database Schema screen is displayed with a list of existing Database schemas.
4. Select the radio button adjacent to *EvalXform_DBSchema* activity and then click **Edit** link. This displays the Edit *EvalXform_DBSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.7).

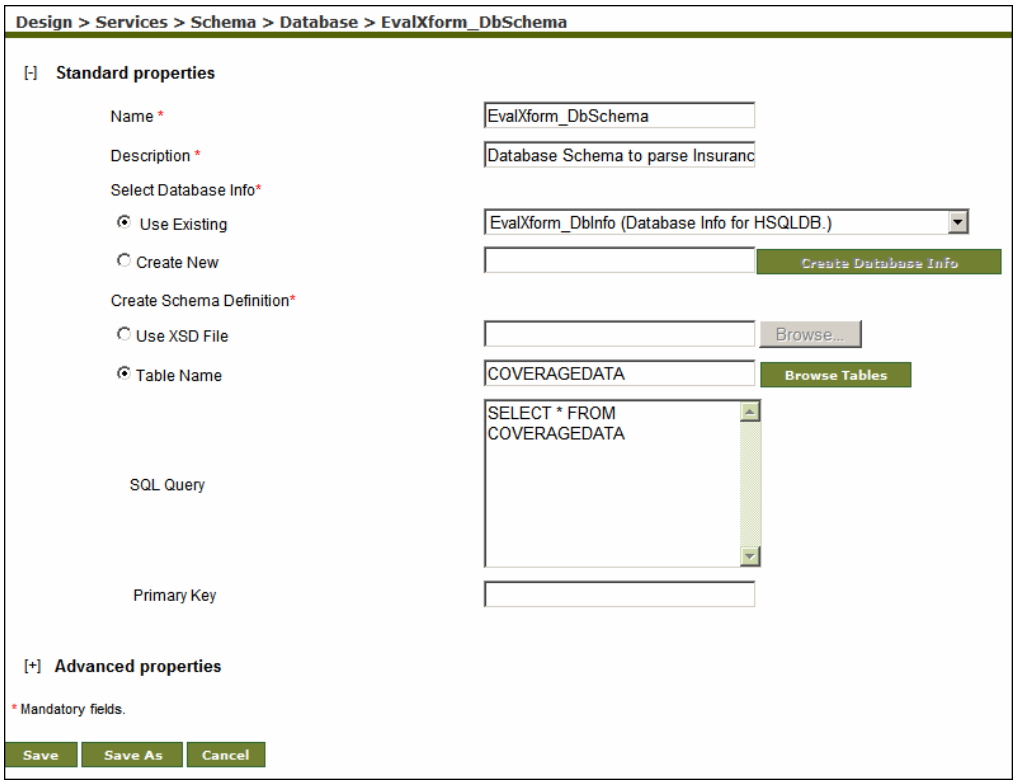



Figure 3.7: Edit *EvalXForm_DBSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 3.5: Details of Fields on Edit Database Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Schema |
| Description | Description of the Database Schema |
| Database Info | Database Info created to connect to the specified Database Server. For more details refer to **Editing Database Info** section. You can use an existing Database Info activity or create a new one. |

| | |
|---|---|
| Create Schema Definition | Schema Definition can be created using one of the following options:<br>▪ Use XSD File<br>▪ Table Name<br>Database Schema used in this Process Flow is created using second option i.e. Table Name. To select database tables, select **Table Name** radio button and then Click the **Browse Tables**. *Select Table* screen is displayed with the list of database Table. Select the required table and click **Get Columns** button. Click **Close** button to close the *Select Table* screen and return to Database Schema screen.<br>SQL Query box automatically gets populated after selecting database tables. |

8. Make the necessary changes.
9. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Schema has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database schema (refer to Figure 7.10).
10. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the database schema has been updated successfully.

## Editing Database Source

## (EvalXform_DBSource)

Database Source is used to insert data into a database server. Database Source uses Database Info for Server URL and login information, and Database Schema to get information for database tables and data type etc.

***Steps to edit the Database Source***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **Database**. The Manage Database Source screen with the list of existing Database Source activities is displayed (see Figure 3.8).

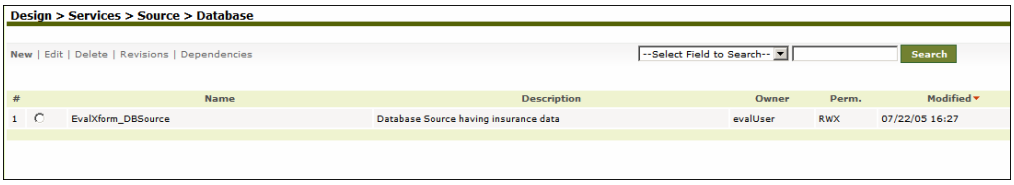



Figure 3.8: Manage Database Source

4. Select the radio button adjacent to *EvalXform_DBSource* activity and then click **Edit** link. This displays the Edit *EvalXform_DBSource* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.9).

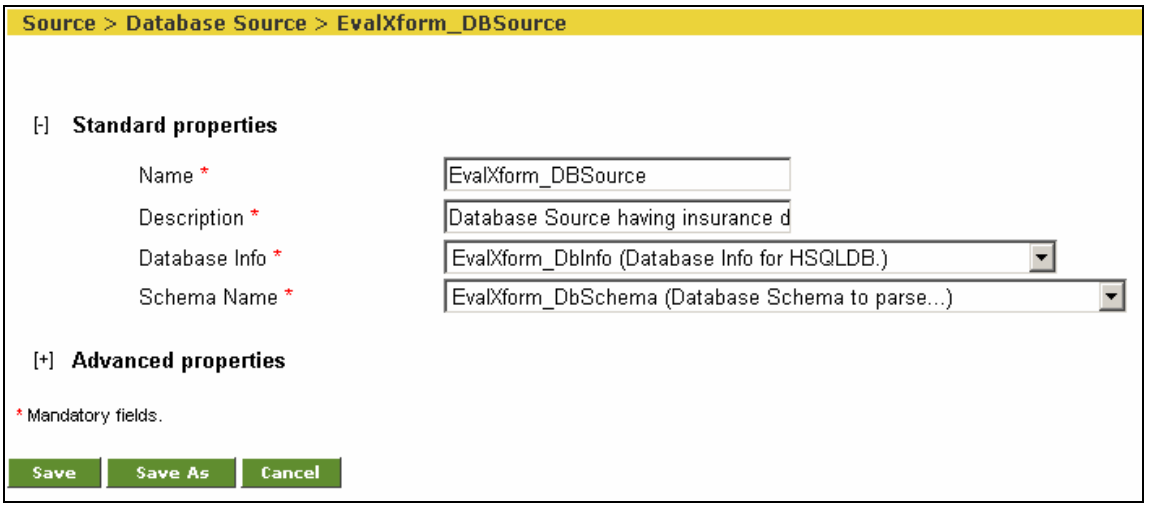



Figure 3.9: Edit *EvalXForm_DBSource* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 3.6: Details of Fields on Edit Database Source Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Source |
| Description | Description of the Database Source |
| Database Info | Database Info created to connect to the specified Database Server. For more details refer to **Editing Database Info** section. |
| Schema Name | Database Schema, which describes the structure of database table. For more details refer to **Editing Database Schema** section. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Source has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database schema (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the database schema has been updated successfully.

## Editing Mapping (EvalXform_Mapping)

Mapping is used to map data fields of the source Schema with the data fields of the target Schema. In this sample Process Flow, advanced mapping functions e.g. Math, String, Aggregation and Custom Method are used.

***Steps to edit the Mapping activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed with the list of existing mapping activities (refer to Figure 7.15).
4. Select the radio button adjacent to *EvalXform_Mapping* activity and then click **Edit** link. This displays the Edit *EvalXform_Mapping* activity screen with the name and description of the activity displayed in their respective fields (see Figure 3.10).

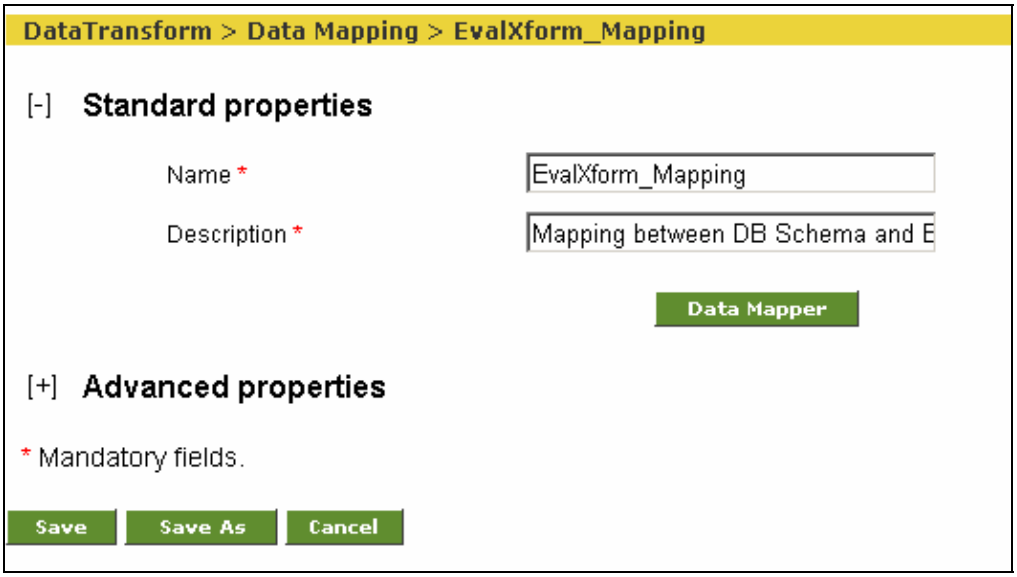



Figure 3.10: Edit *EvalXForm_Mapping* Activity

5. Click **Data Mapper** button**.** The Data Mapper applet is displayed showing mapping between data fields of source and target schema (see Figure 3.11).

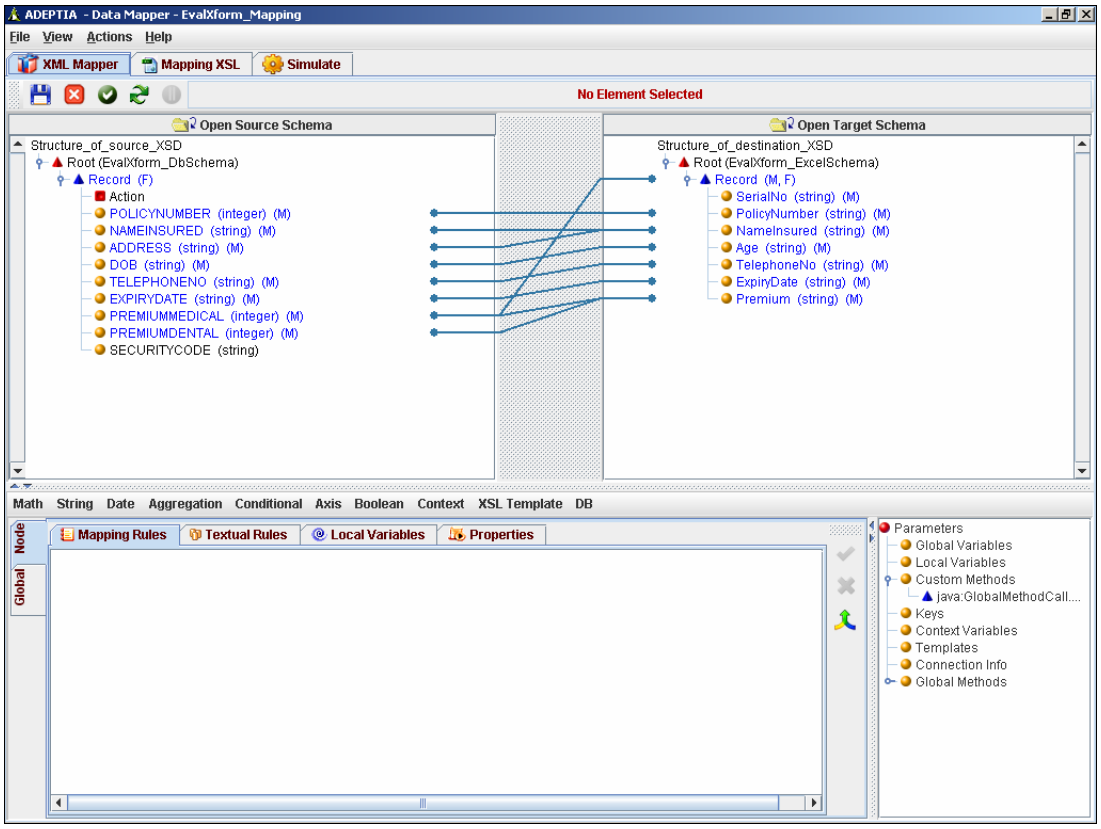



Figure 3.11: *EvalXForm_Mapping* Activity in the Data Mapper Applet

6. To view how mapping function is used, click any of the target elements (*Record*) in the Target Panel. The mapping function used for the selected element is displayed in the Mapping Graph Area (see Figure 3.12).

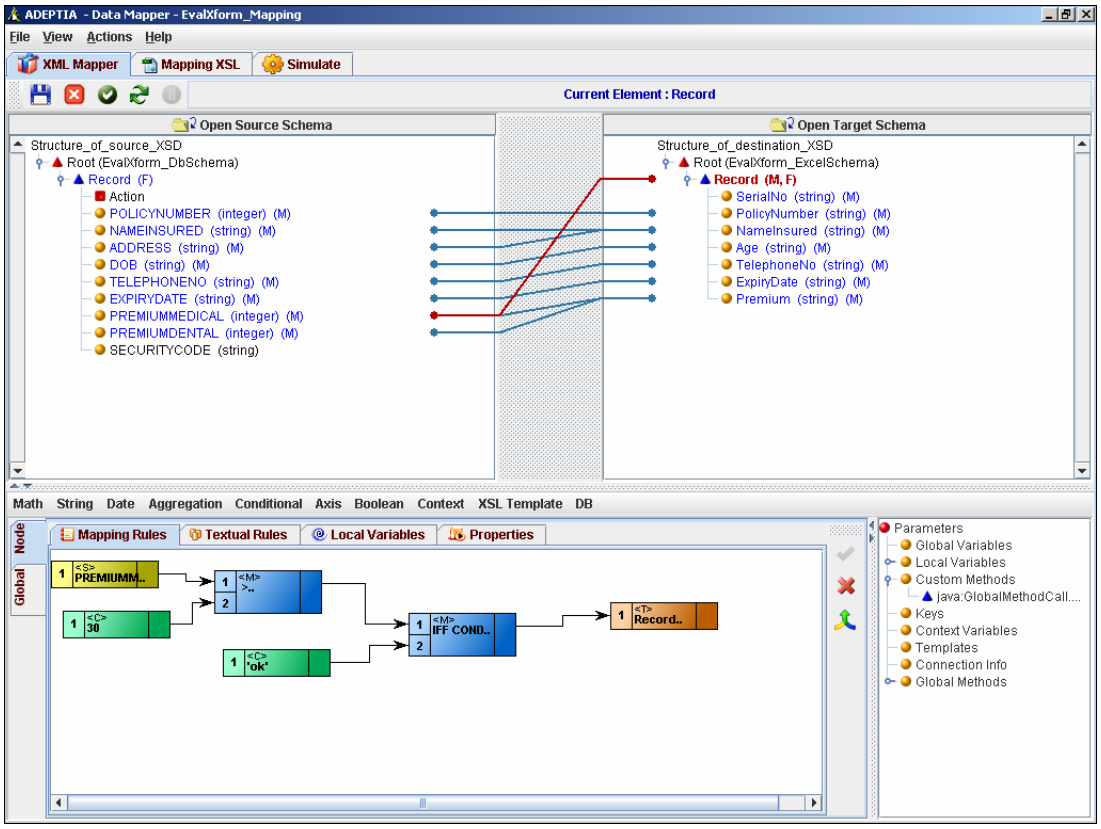



Figure 3.12: View Mapping Functions used in *EvalXForm_Mapping* Activity

7. To analyze the impact of mapping function on output records, right-click any of function node (*IFF Condition*) and select **Information** (see Figure 3.13).

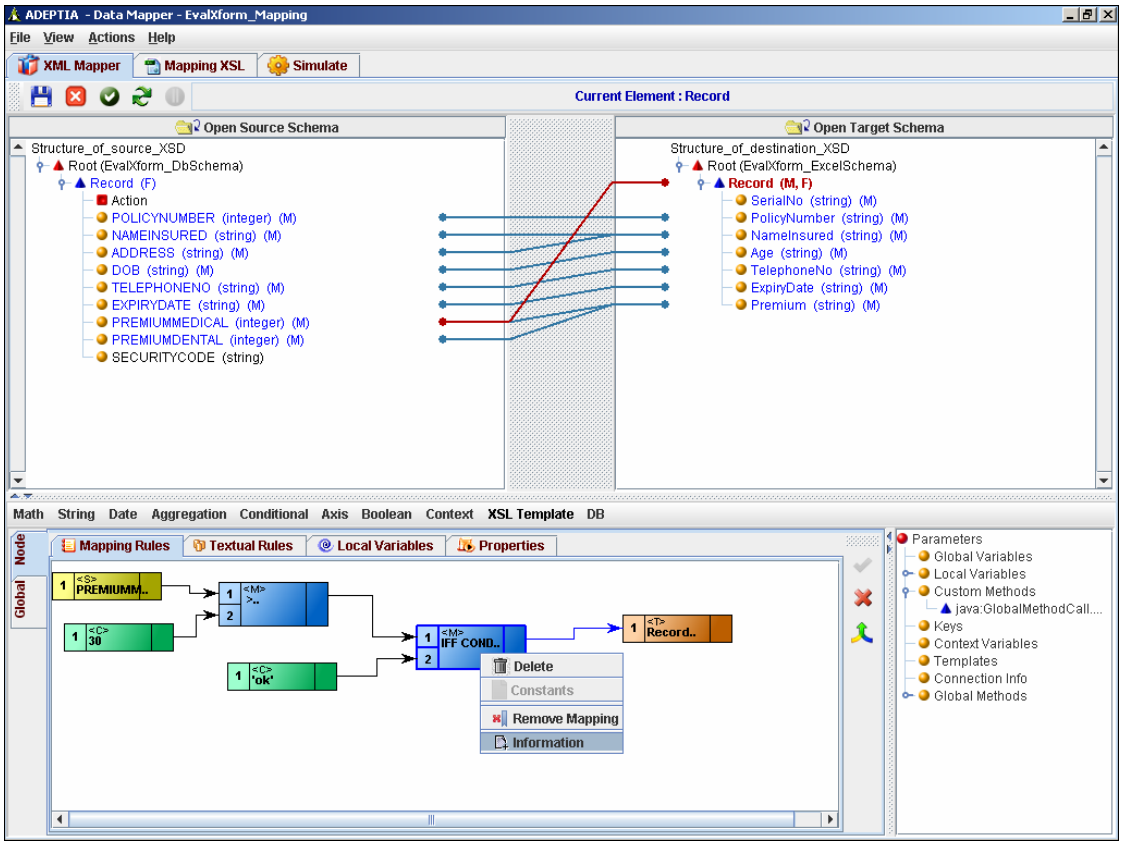



Figure 3.13: Select Mapping Function Information

8. A dialog box is displayed that shows the information about that mapping function (see Figure 3.14).

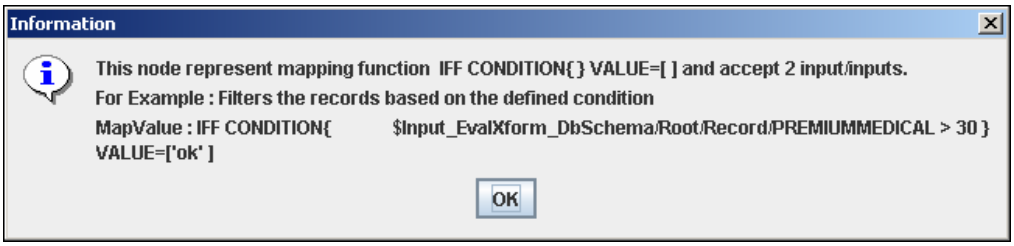



Figure 3.14: View Mapping Function Information

9. Click **OK** to close the dialog box.

10. Make the necessary changes to the mapping between the source and target data fields.

11. Once you have made the required changes, save the mapping by clicking **File** menu and selecting **Save**. A dialog box is displayed confirming that the mapping activity has been saved successfully.

| | |
|---|---|
| | Alternately, you can save the mapping by clicking **Save** ( ) button on the Tool Bar. |

12. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping (refer to Figure 7.18).
13. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

14. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.
15. Exit the Data Mapper applet by clicking **File** menu and selecting **Exit**.

| | |
|---|---|
| | To know, how this mapping activity has been created, refer to **Creating Mapping Activity** section. |

## Editing Excel Schema

## (EvalXform_ExcelSchema)

Excel Schema defines the structure of Excel file. Excel Schema defines how the data can be read from an excel sheet or how it can be written into an excel sheet. In this sample Process Flow, Excel schema is used at the target end. At target end Excel Schema converts the data from an intermediate XML format into Excel format.

***Steps to edit the Excel Schema activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Excel**. The Manage Excel Schema screen is displayed with the list of existing Excel Schemas (see Figure 3.15).

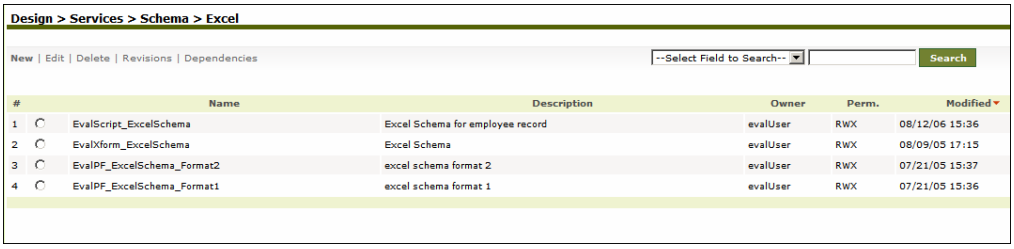



Figure 3.15: Manage Excel Schema

9. Select the radio button adjacent to *EvalXform_ExcelSchema* activity and then click **Edit** link. This displays the Edit *EvalXform_ExcelSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.16).

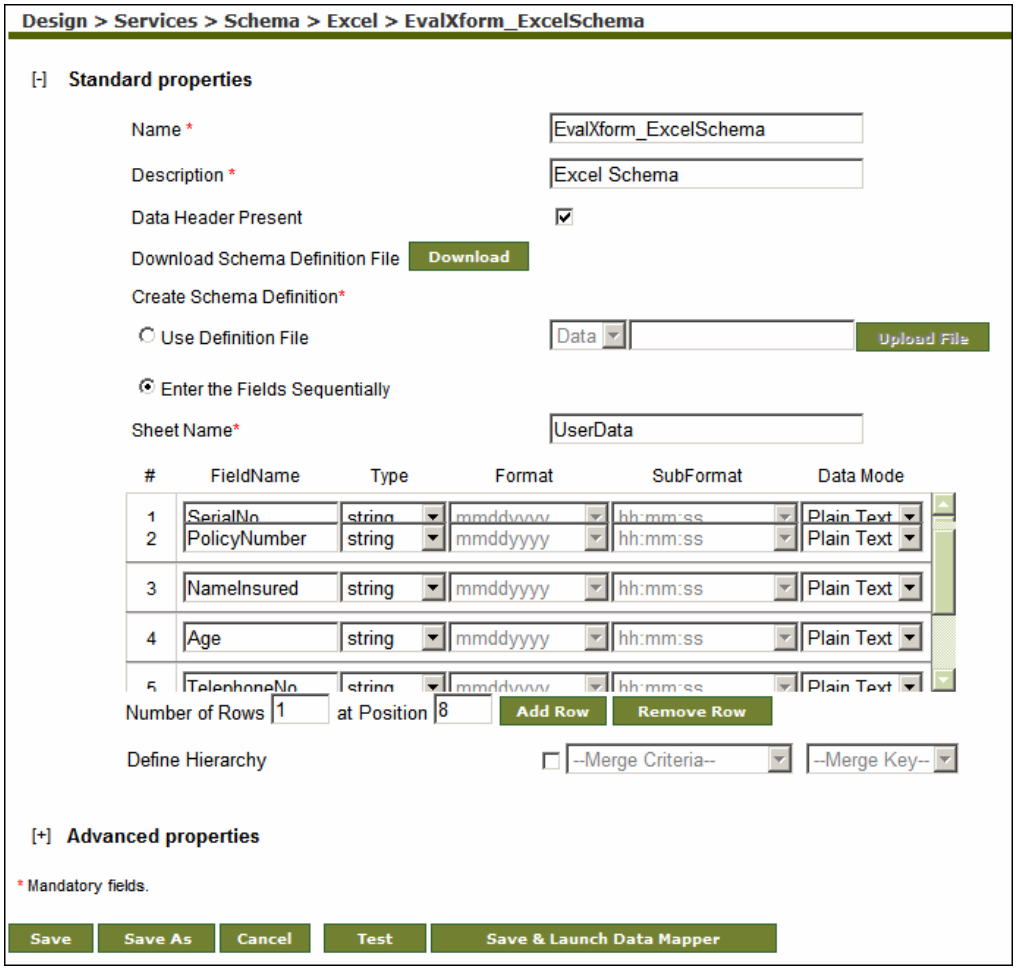



Figure 3.16: Edit *EvalXForm_ExcelSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 3.7: Details of Fields on Edit Excel Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Excel Schema |
| Description | Description of the Excel Schema |
| Data Header Present | Data Header usually contains the titles of the fields in a file. If you enable this option, Field Names of Excel Schema are written as column's name in target excel file |
| Sheet Name | Name specified here becomes the Book Name of the excel sheet. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File<br>▪ Enter the Field Sequentially<br><br>Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |

| Field Name | Name of the Fields | |
|---|---|---|
| Type | There are three data types: | |
| | String | String can be used for any type of data. |
| | Number | Contains numbers |
| | Date | Contains Date and Time |
| | Currency | Contains Currency Value |
| Format | If data type is *Date*, select the data format from *Format* drop-down list. | |
| SubFormat | If data type is *Date*, select time format from *SubFormat* drop-down list. | |
| Data Mode | If data is in Plain Text, select *Plain Text* option. If data is encrypted, then select *Encrypted* option. | |

10. Make the necessary changes.
11. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Excel Schema has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the excel schema (refer to Figure 7.10).
12. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the excel schema has been updated successfully.

## Testing Excel Schema (EvalXForm_ExcelSchema)

You can verify the excel schema activity at design time.

***Steps to verify schema activity***

1. Click **Test** button on the Edit Excel Schema screen. The Test Schema screen is displayed (refer to Figure 7.14).
2. Select the type of schema to test, from the *Type* drop-down list. By default, *Source* is selected. Since this Excel schema is used at target end, select Target from *Type* drop-down list.
3. Enter the full path of the XML file generated by EvalXForm_Mapping activity, in the *Source File Name* field.
4. Enter the full path (with file name) of the target file, where it will be generated, in the *Target File Name* field.
5. Enter the full path of the XML file where errors will be stored, in the *Error File Name* field.
6. Click **Submit** button. This tests the validity of the excel schema.

## Editing File Target (EvalXform_FileTarget)

File target is used to specify the name of the target file and path, where the target file is saved. In this sample Process Flow, target file is saved in **../../AdeptiaServer-5.0/Sample Datafiles/EvalXform** directory**.**

***Steps to edit the File Target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **File**. The Manage File Target screen is displayed with the list of existing File Target activities (see Figure 3.17).

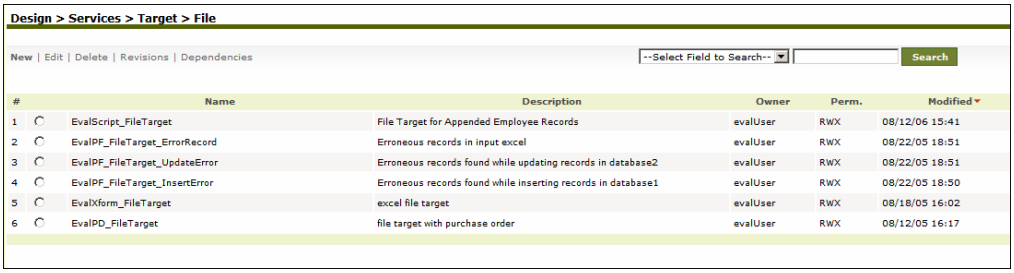



Figure 3.17: Manage File Target

4. Select the radio button adjacent to *EvalXform_FileTarget* activity and then click **Edit** link. This displays the Edit *EvalXform_FileTarget* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.18).

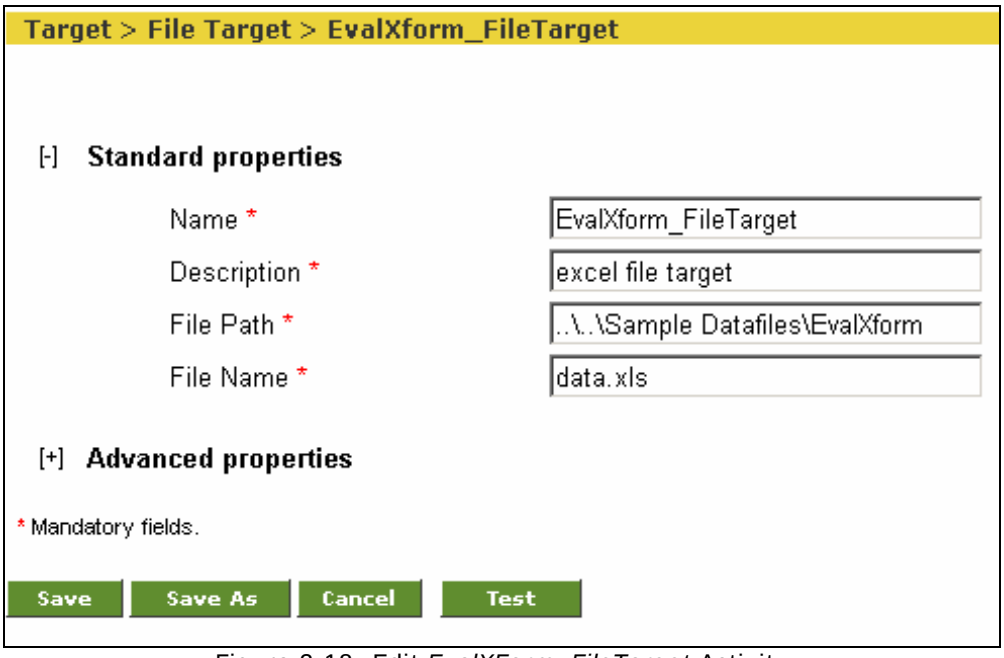



Figure 3.18: Edit *EvalXForm_FileTarget* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 3.8: Details of Fields on Edit File Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Target |
| Description | Description of the File Target |
| File Path | Path of the target file.<br>For example:<br>**`../../Solutions/Demo/EvalXform/`** |
| File Name | Name of the target file. For example: **`data.xls`** |

5. Make the necessary changes.
6. Once you have the made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file target (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

8. Click **OK** to save the comments. This displays a screen confirming that the file target has been updated successfully.

| | |
|---|---|
|  | You can verify the file target activity at design time. For this, click **Test Connection**. This verifies the values in the *File Name* and *File Path* field and checks whether the file actually exists in the specified location. |

## Edit Native Call

## (EvalXform_NativeCall)

Native call is used to run any .bat, .exe or .sh file at any point during the execution of a Process Flow. In this sample Process Flow, Native Call is used to run a .bat file, which opens the Target Excel file created after the execution of the Process Flow.

***Steps to edit the Native Call:***

1. n the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Native Service** to expand the tree, and then click **Native Call**. The Manage Native Call screen is displayed with the list of existing Native Calls (see Figure 3.19).

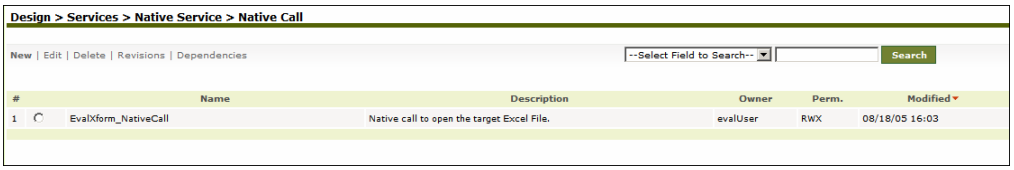



Figure 3.19: Manage Native Call

4. Select the radio button adjacent to *EvalXform_NativeCall* activity and then click **Edit** link. This displays the Edit *EvalXform_NativeCall* activity screen, with the properties of the activity displayed in their respective fields (see Figure 3.20).

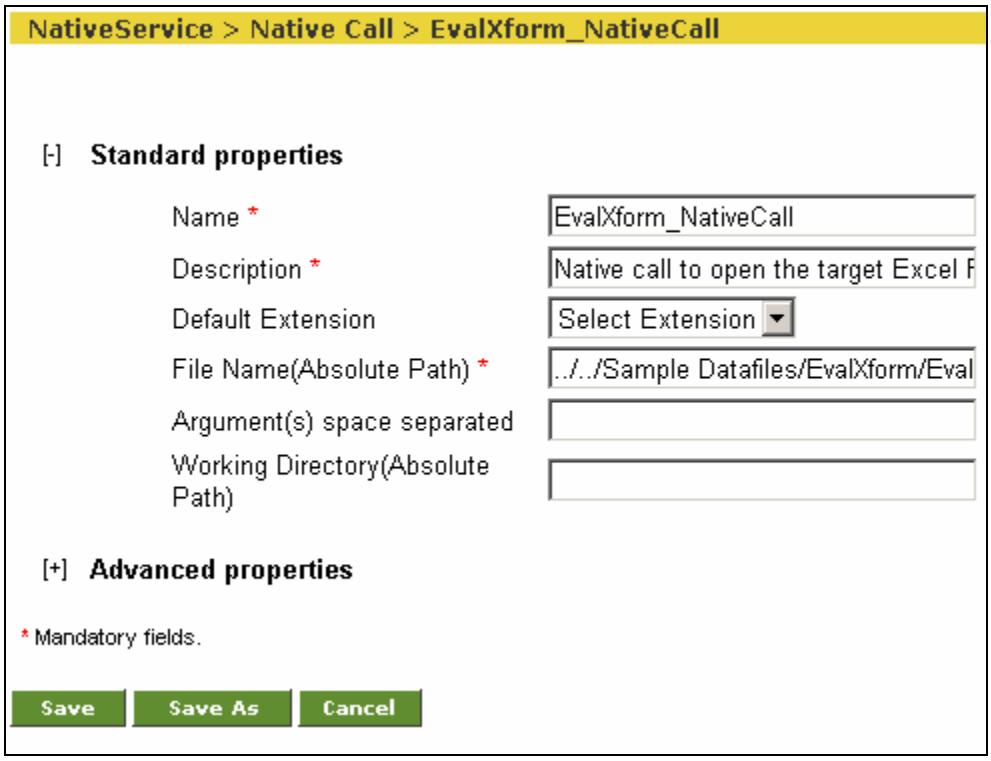



Figure 3.20: Edit *EvalXForm_NativeCall* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 3.9: Details of Fields on Edit Native Call Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Native Call activity |
| Description | Description of the Native Call activity |
| Default Extension | Extension of the file whether .bat, .exe or .sh |
| File Name (Absolute Path) | Name of the file with absolute path<br>For example:<br>**`../../Solutions/Demo/EvalXform/EvalXform_OpenFile.bat`** |
| Arguments | Any arguments for selected batch or executable file |
| Working Directory | Directory, where you want the run the specified batch or executable file |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Native Call activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the native call activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the native call activity has been updated successfully.

## CREATING MAPPING ACTIVITY

Mapping is used to map data fields of source schema and target schema. In this Process Flow, different mapping functions (e.g. String, Math, Aggregation, Custom Method and Conditional Functions) are used.

***Step to create the Mapping activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed with the list of existing mapping activities (refer to Figure 7.15)
4. Click **New** link. The Create Mapping Activity screen is displayed (see Figure 3.21).

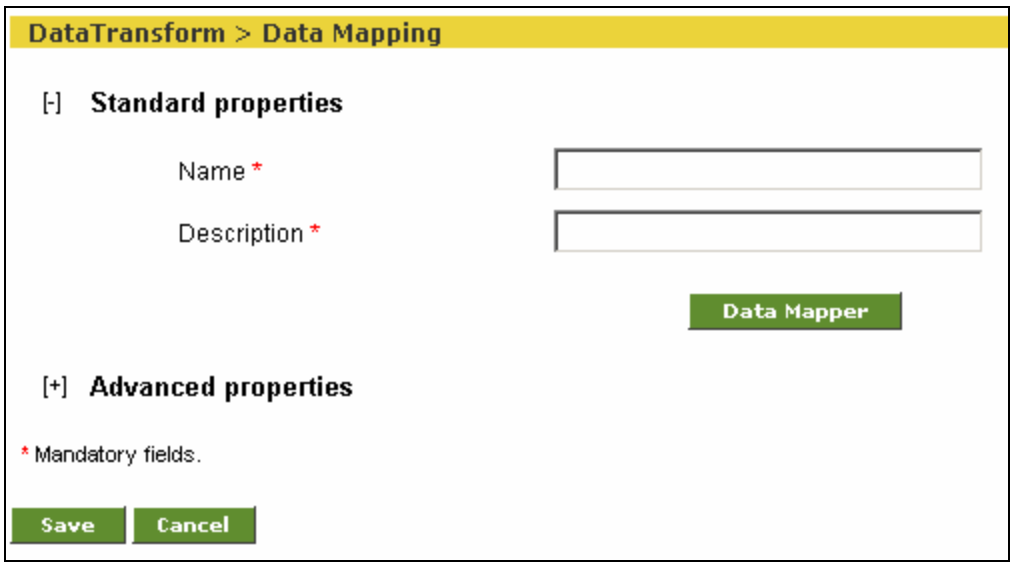



Figure 3.21: Create Data Mapping Activity

5. Enter the name and description of the new mapping activity in the *Name* and *Description* fields respectively.
6. Click **Data Mapper** button. This displays the Data Mapper applet (see Figure 3.22)

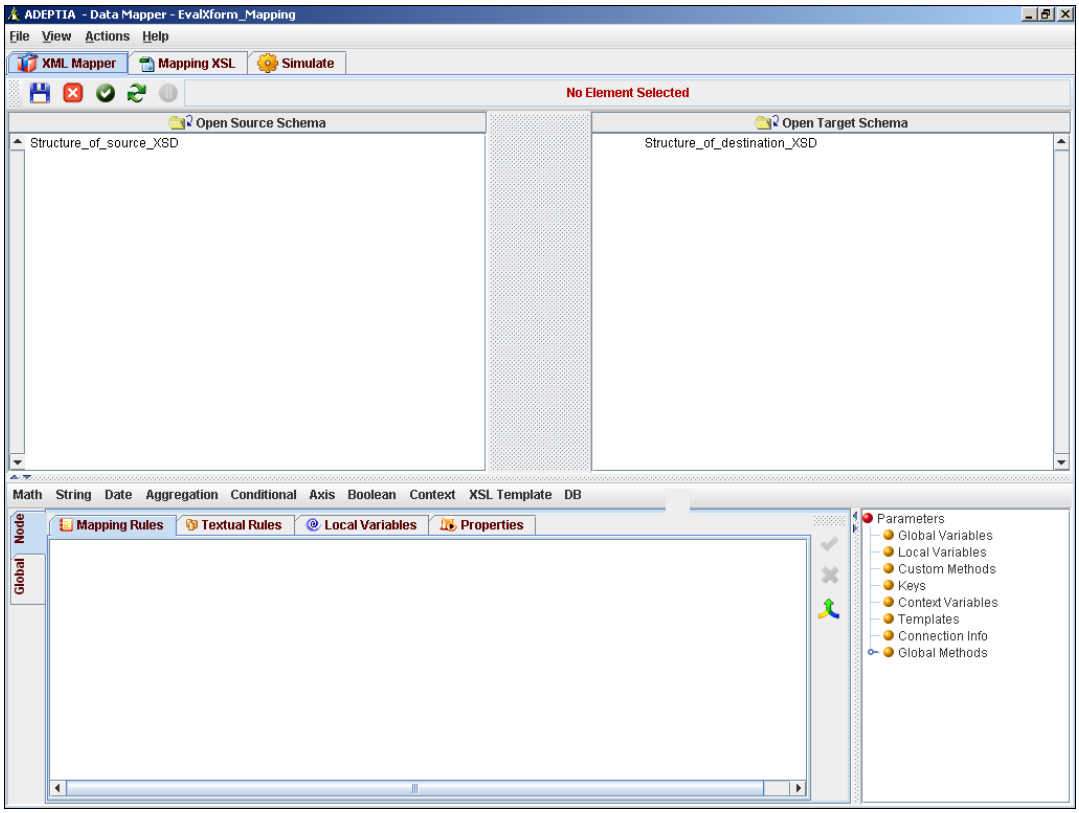



Figure 3.22: Data Mapper Applet

7. Click **Open Source Schema** button to load the source schema. This displays the Select Schema screen (see Figure 3.23).

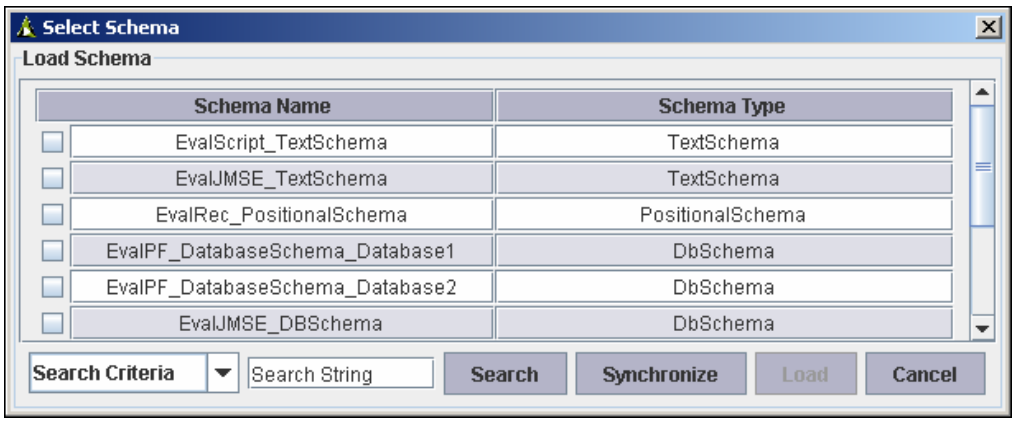



Figure 3.23: Select Schema

8. Select the Source Schema (*EvalXform_DBSchema*) checkbox from the *Source Schema* list and click **Load** button. This loads the selected schema in the Source Panel of the Data Mapper applet.

9. Similarly, load the Target Schema (*EvalXform_ExcelSchema*) in the Target Panel of the Data Mapper applet.

10. Click ( ) to expand the tree structure for Source Schema and the Target Schema and display their elements (see Figure 3.24).

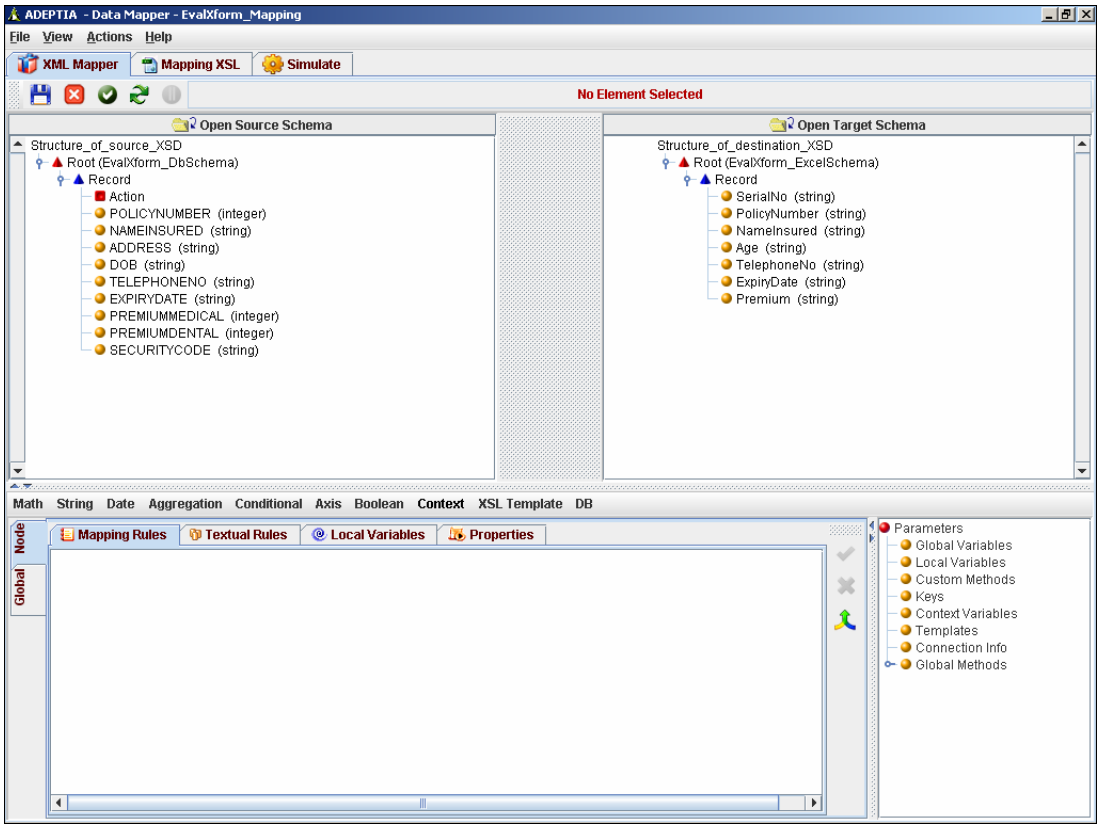



Figure 3.24: Source and Target Elements

11. Select the **Record** element of the target schema and then click **Properties** tab in Mapping Graph Area.

12. Click the *For Each* field and then double-click the **Record** element of the source schema. Click **Save Properties** to save the *For Each* property (see Figure 3.25).

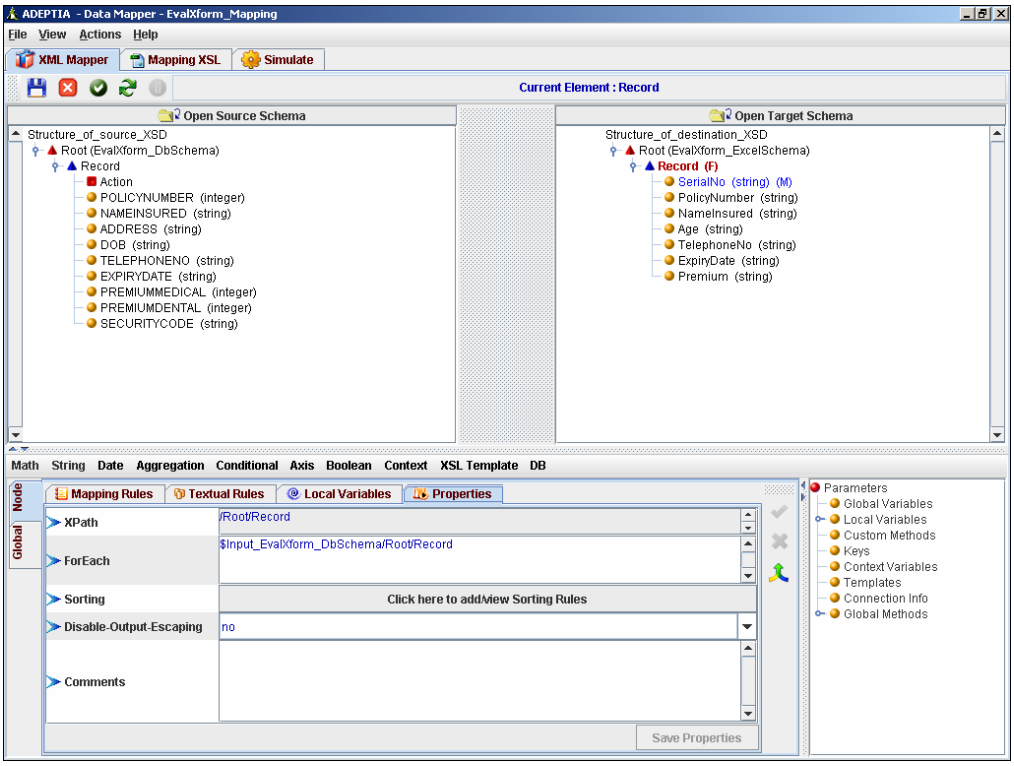



Figure 3.25: Apply *For Each* Property

13. Click the node **SerialNo.** on the Target Panel. The node *SerialNo.* is displayed in the Mapping Graph Area (see Figure 3.26).

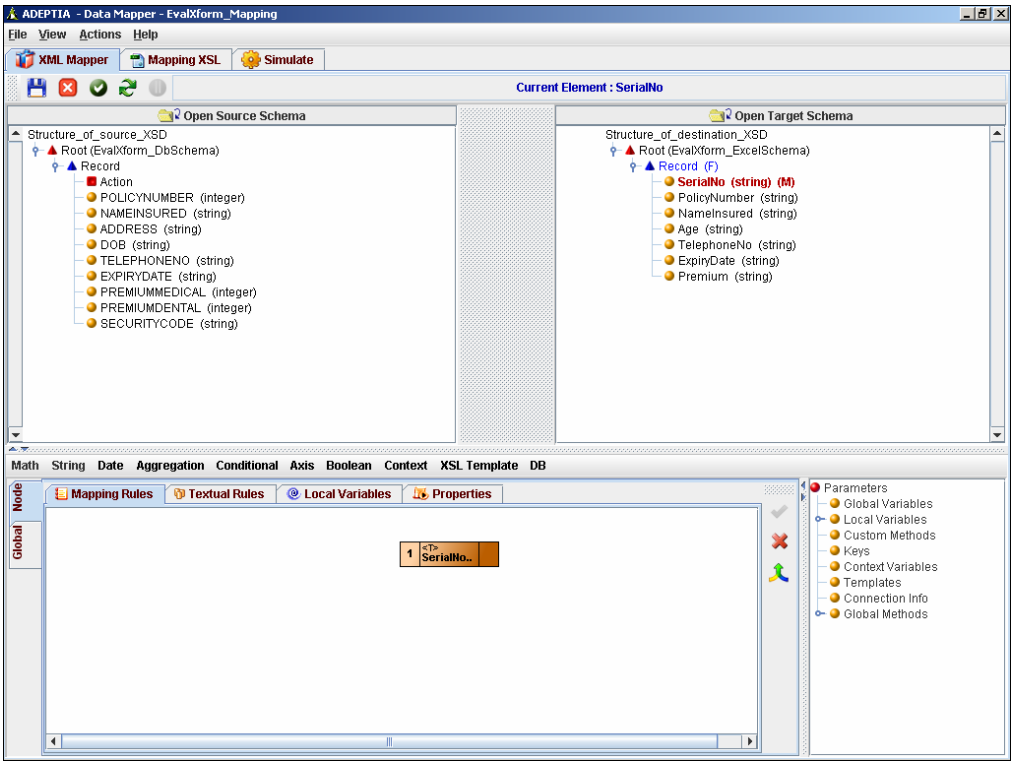



Figure 3.26: Select Target Node

14. Click the **Aggregation** mapping function and select the **Position** sub-function. A node for **Position** function is displayed in the Mapping Graph Area (see Figure 3.27).

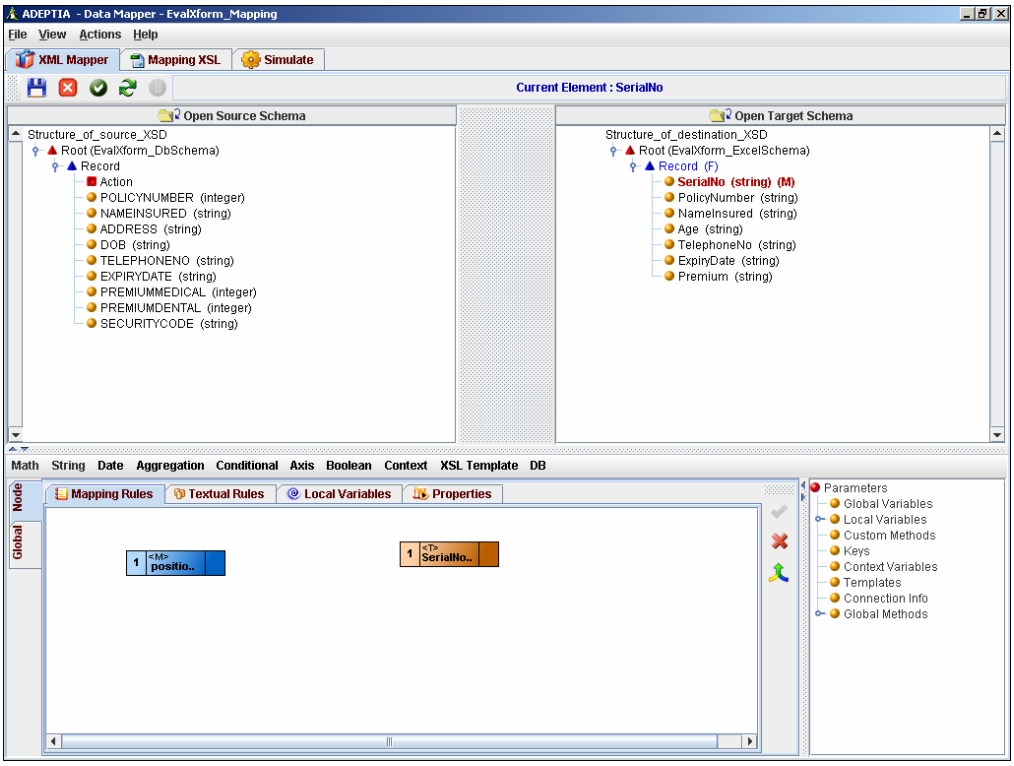



Figure 3.27: Select Position Function

15. Click the **Position** node in the Mapping Graph Area and drag the mouse pointer from the *Position* node to the *SerialNo* node. A line is displayed between *Position* node and *SerialNo* node (see Figure 3.28).

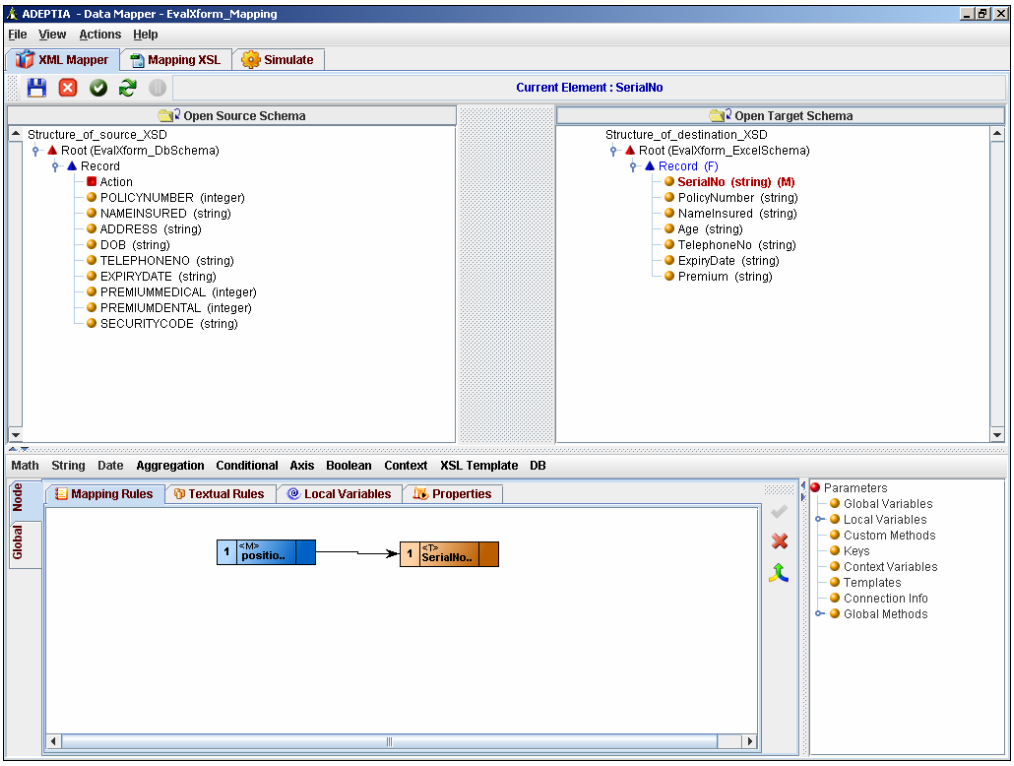



Figure 3.28: Map Position Function to Serial No.

16. Click the **Apply Mapping** ( ) button to apply the mapping.

17. Click *PolicyNumber* in the Source Panel and drag the mouse pointer from the Source Panel to the *PolicyNumber* node in the Target Panel (see Figure 3.29).

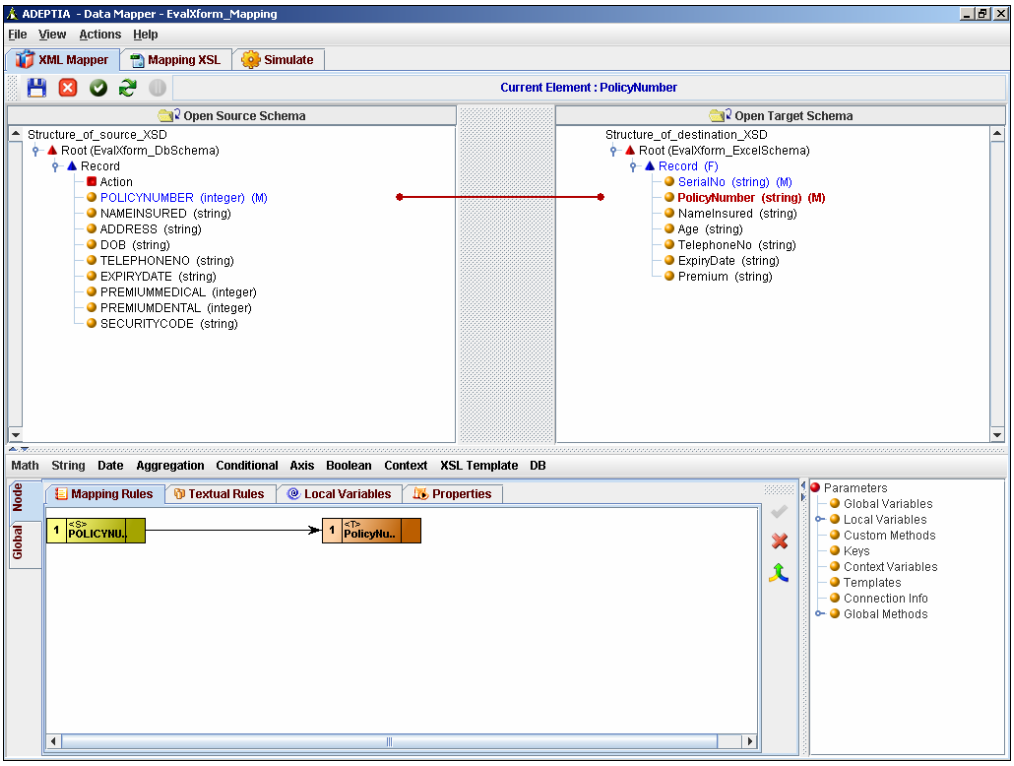



Figure 3.29: Map Policy Numbers

| | |
|---|---|
|  | The character **(M)** is suffixed next to the source and target elements to imply that they have been mapped. |

18. To concatenate *NameInsured* and *Address* of the source schema, select the **NameInsured** node from the Target Panel. The selected node is shown in the Mapping Graph Area (see Figure 3.30).

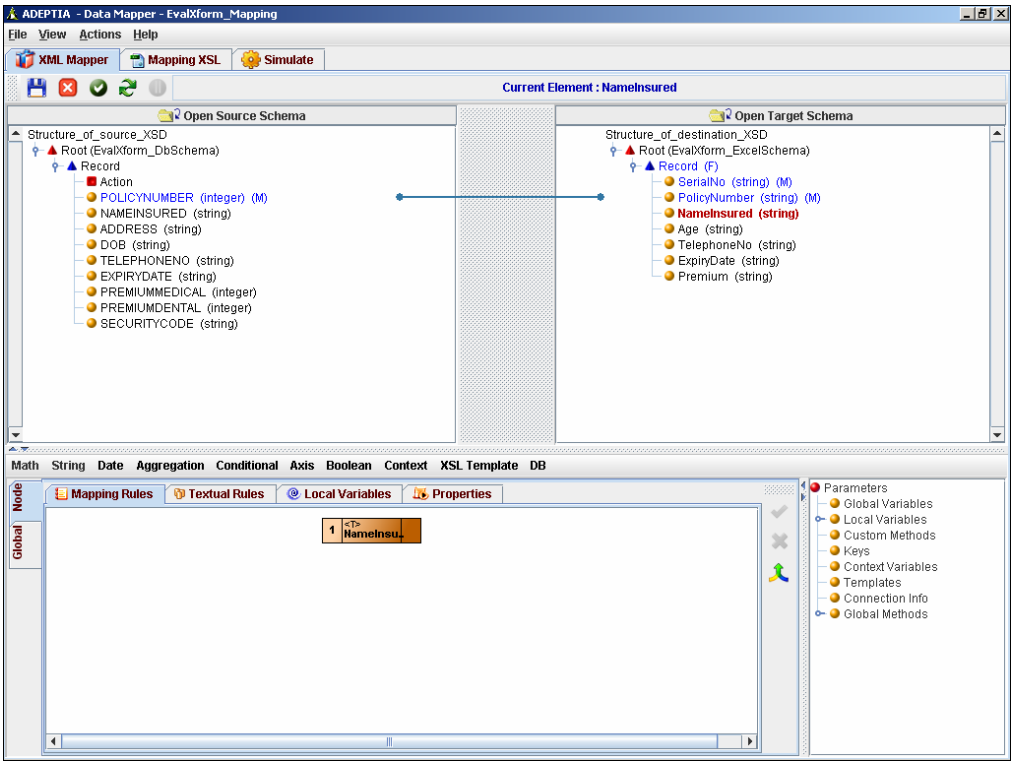



Figure 3.30: Select Name Insured

19. Double-click the **NameInsured** node in the Source Panel. The *NameInsured* node is shown in the Mapping Graph Area. Similarly, double-click the **Address** node in the Source Panel (see Figure 3.31).

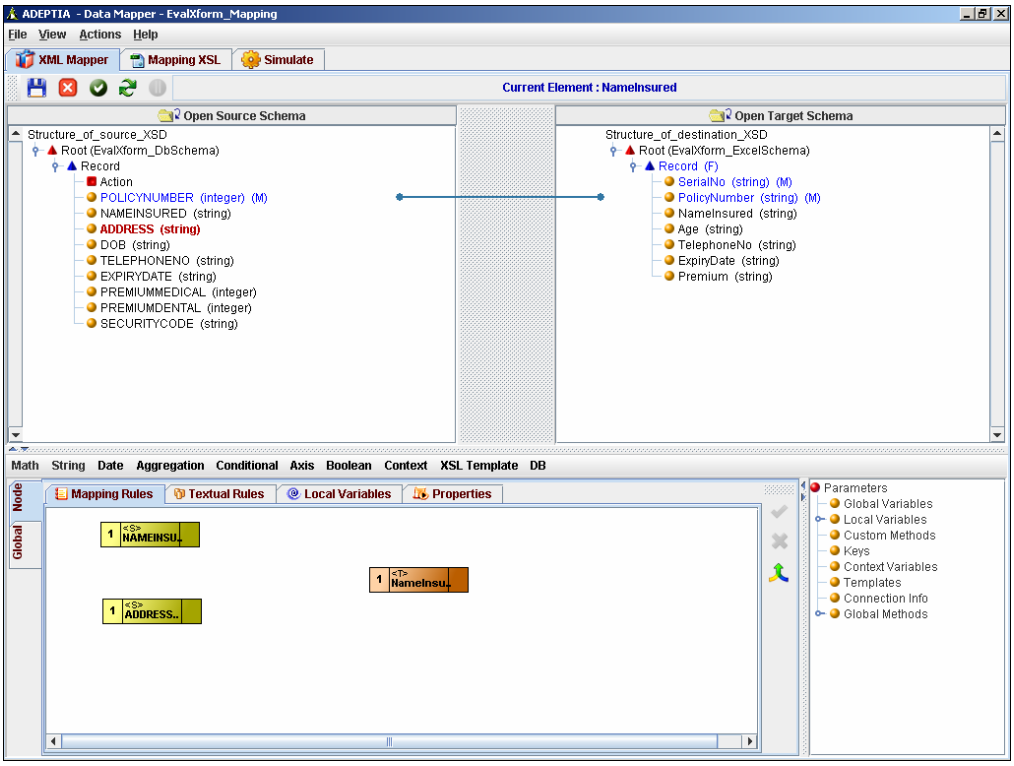



Figure 3.31: Select Source Nodes

20. Click the **String** mapping function and select **Concat** sub-function. The *Concat* node is shown in the Mapping Graph Area.
21. Create a link from the output of the *NameInsured* node to the first input of the *Concat* node.
22. Create a link from the output of the *Address* Node to the second input of the *Concat* node.
23. Create a link from the output of the *Concat* function node to input of the *NameInsured* node (see Figure 3.32).

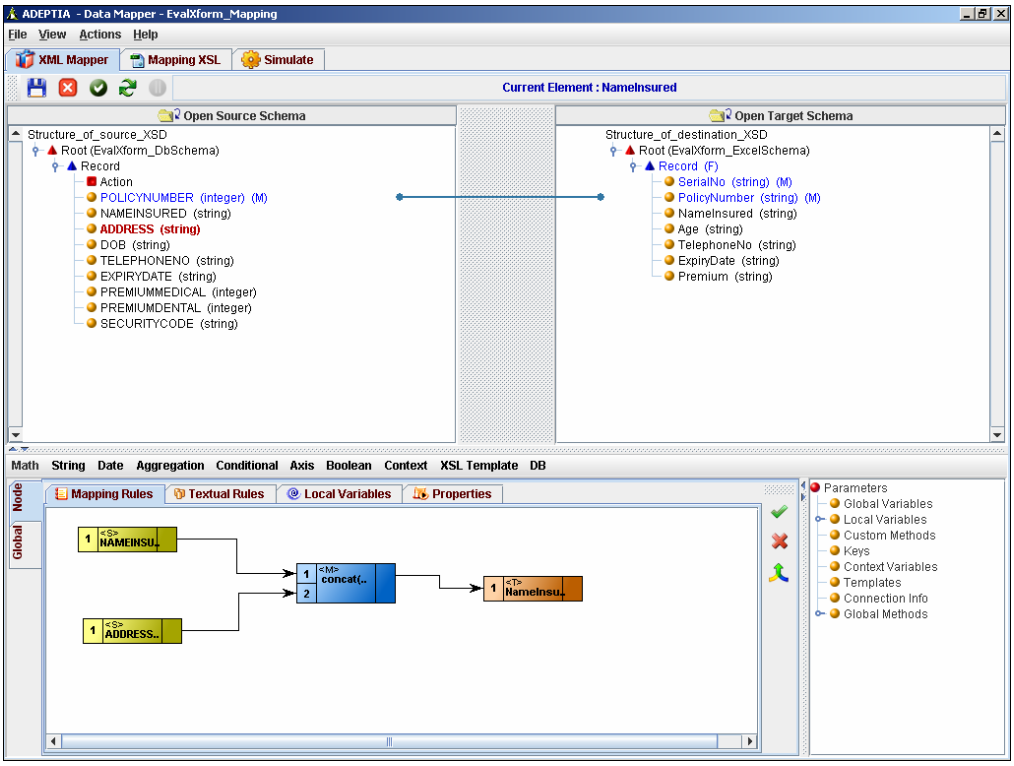



Figure 3.32: Create Links

24. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 3.33).

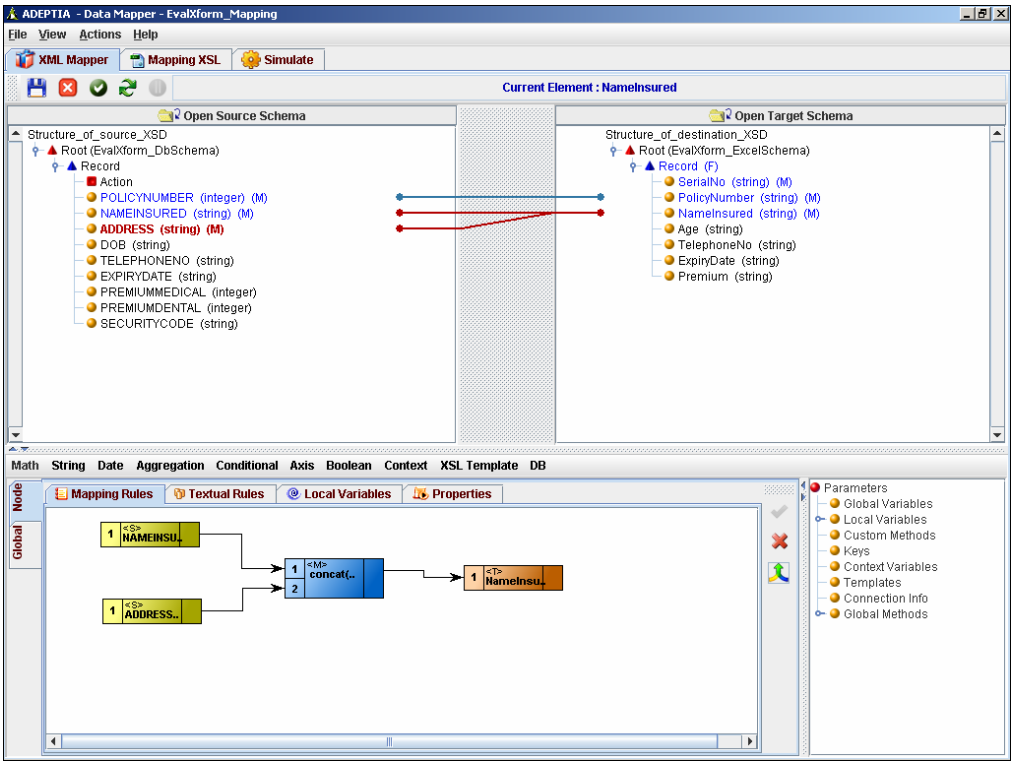



Figure 3.33: Apply Mapping

25. To calculate the Age based on the *DOB* (Date of Birth) field of the source schema, custom method is used. A custom method is used to call a Java method, which calculates the current age, based on *DOB*.

26. To define a Custom Method, click the **Global** tab in the Mapping Graph Area.

27. Click the **Custom Methods** tab. The Custom Methods Panel is displayed (see Figure 3.34).

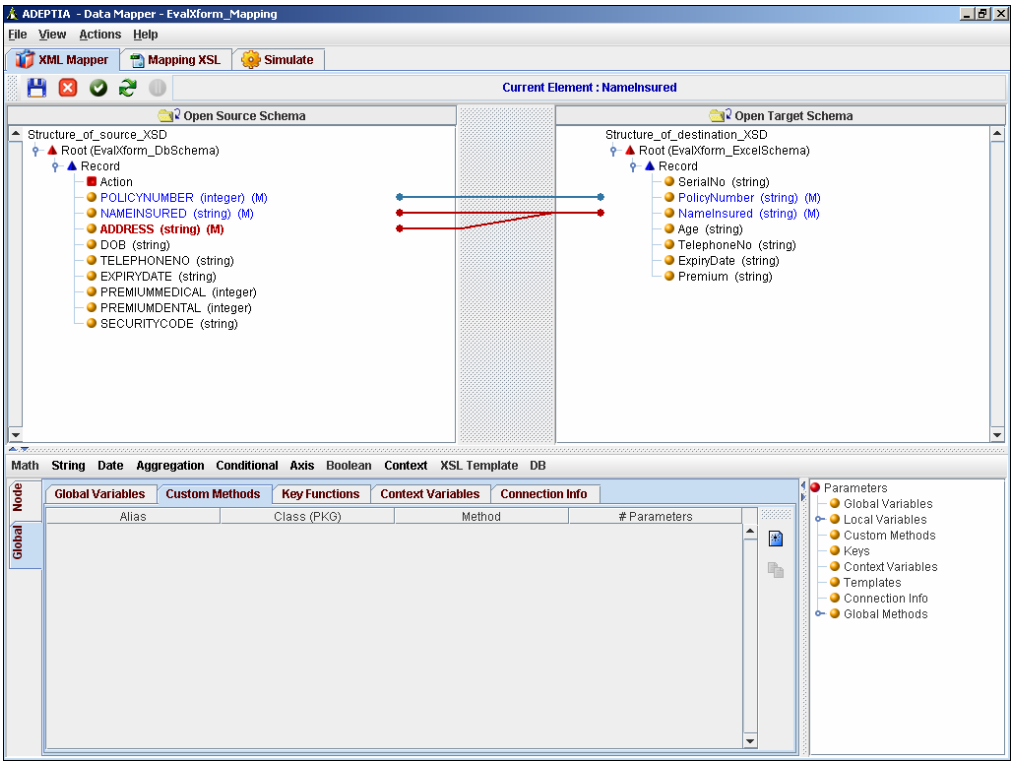



Figure 3.34: Custom Methods Pane

28. Click the **Add Method** ( ) button to add a Custom Method. A row is inserted with following columns (see Figure 3.35):

- Alias
- Class (PKG)
- Method
- # Parameters

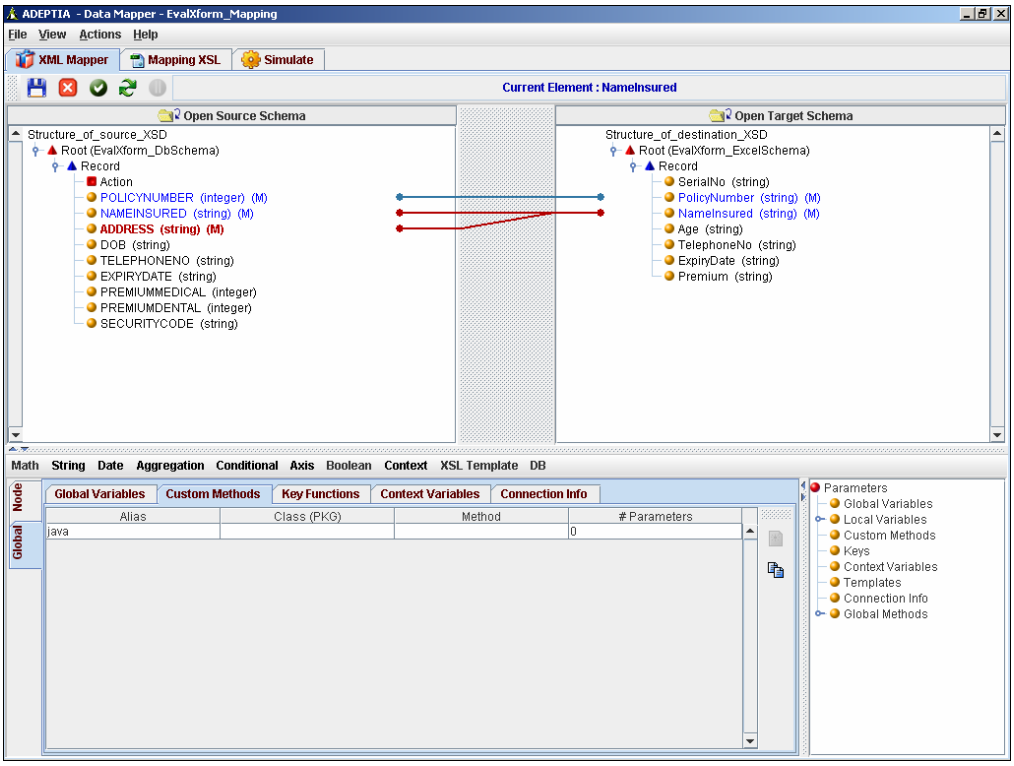



Figure 3.35: Add Custom Method

29. *Alias* column automatically gets populated with value 'java'.
30. Click the *Alias* field, press [**Tab**] or [**Enter**] key to go to the *Class (PKG)* field
31. Enter the fully qualified name of the Java class (*CustomMethodCall*) in the *Class (PKG)* field.
32. Press [**Tab**] or [**Enter**] key to go to the *Method* field and enter the name of the method (*getAge*) in the *Method* column.

| | |
|---|---|
|  | The Java Method specified here is stored in ***../serverkernel/CustomClasess*** folder. |

33. Press [**Tab**] or [**Enter**] key to go to the *#Parameters* field and enter the number of arguments (*1*) taken by Method in the *# Parameter* column.
34. Press [**Tab**] or [**Enter**] key to return to the *Alias* field. This will save the added Custom Method (see Figure 3.36).

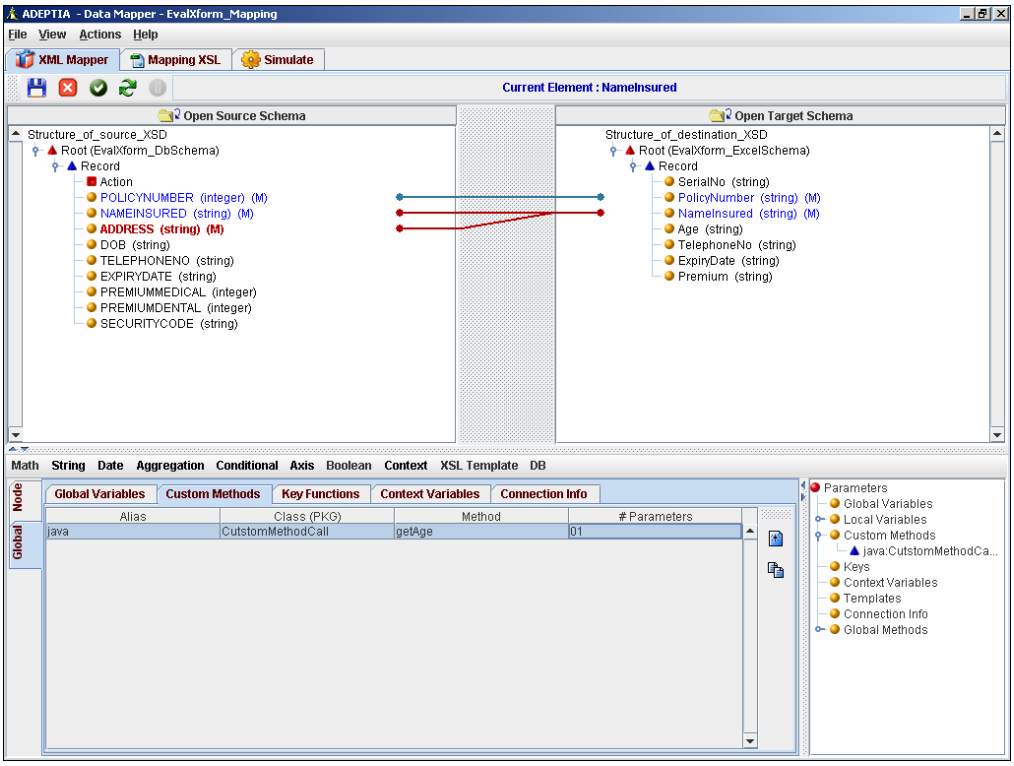



Figure 3.36: Save Custom Method

| | |
|---|---|
|  | The saved custom method is displayed under *Custom Methods* in the Parameters Panel. |

35. Click the **Node** tab to map the declared Custom Method to the *Age* node. The Mapping Graph Area is displayed.
36. Click the node **Age** in the Target Panel. The *Age* node is displayed in the Mapping Graph Area.
37. Double-click the **DOB** node in the Source Panel. The *DOB* node is displayed in the Mapping Graph Area.
38. Expand the *Custom Methods* tree in the Parameters Panel. Double-click the defined Custom Method. The selected Custom Method node is displayed in the Mapping Graph Area (see Figure 3.37).

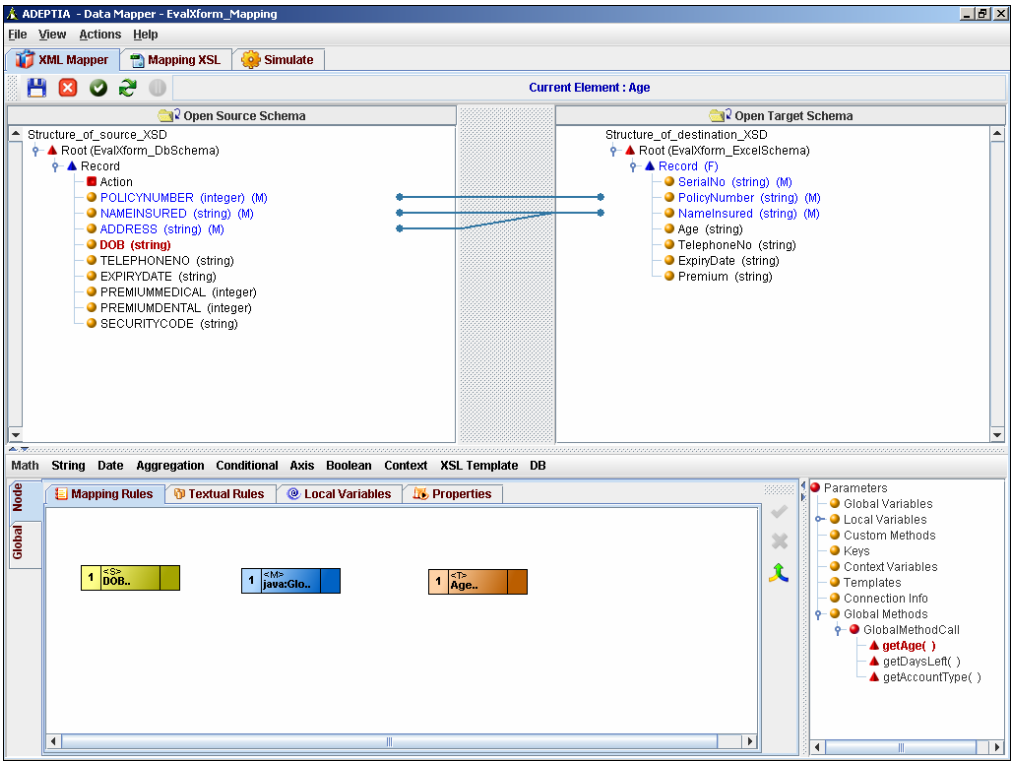



Figure 3.37: Select Custom Method

39. Create a link from the output of the *DOB* node to the input of the *Custom Method* node.

40. Create a link from the *Custom Method* node to the input of the *Age* node.

41. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 3.38).

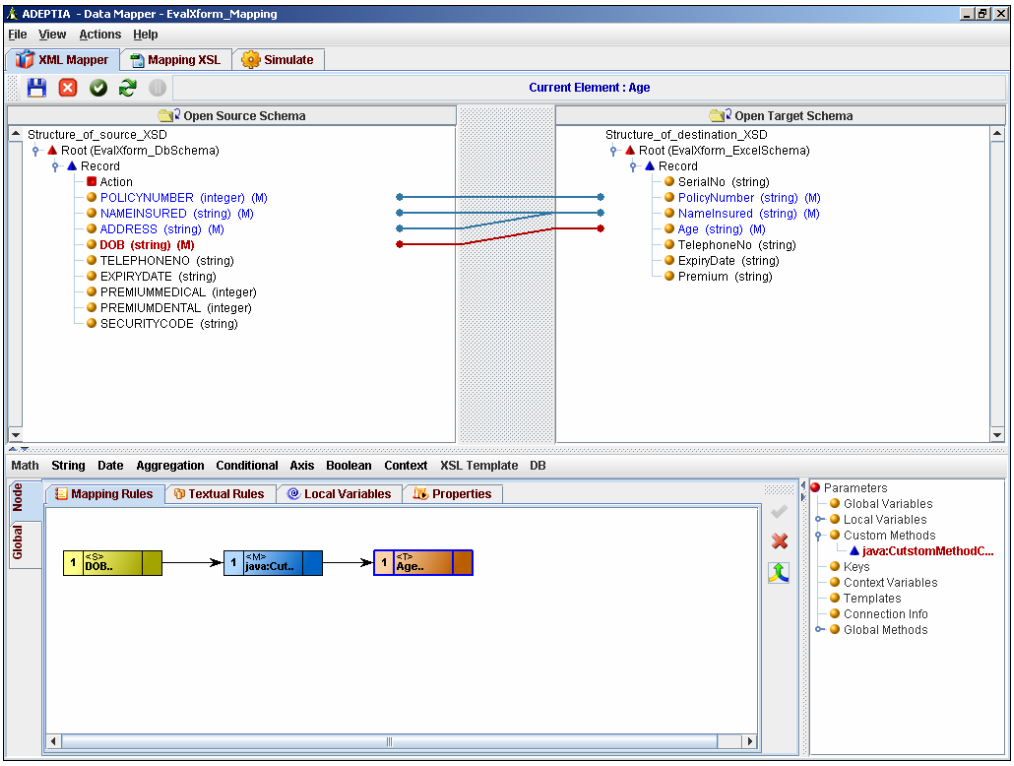



Figure 3.38: Map Custom Method

42. Click the **TelephoneNo** node in the Source Panel and drag the mouse pointer from the Source Panel to the **TelephoneNo** node in the Target Panel.

43. Similarly, map **ExpiryDate** node of the Source Panel with the **ExpiryDate** node of the Target Panel.

44. To add the values of *PremiumMedical* and *PremiumDental*, the **Add** function is used.

45. To use the **Add** function, click **Premium** node in the Target Panel. The *Premium* node is displayed in the Mapping Graph Area.

46. Double-click the **PremiumMedical** node in the Source Panel. The *PremiumMedical* node is displayed in the Mapping Graph Area. Similarly, double-click the **PremiumDental** node in the Source Panel (see Figure 3.39).

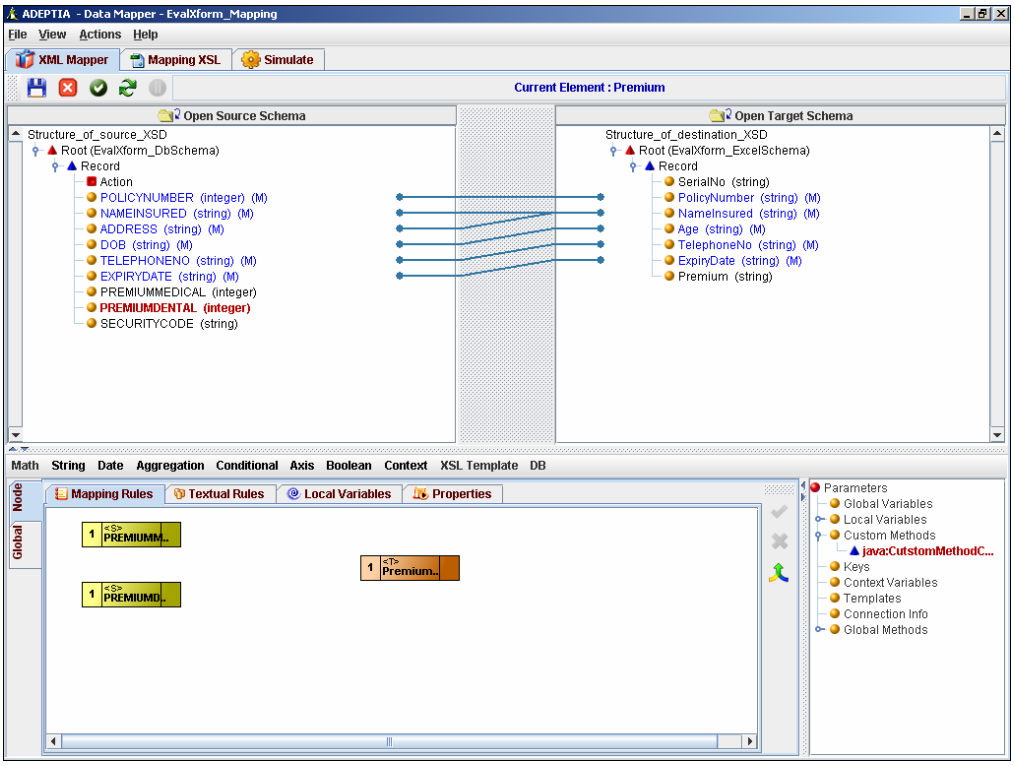



Figure 3.39: Select Source Nodes

47. Click the **Math** mapping function and select **Add** sub-function. The *Add* function node is shown in the Mapping Graph Area.

48. Create a link from the output of the *PremiumMedical* node to the first input of the *Add* function node.

49. Create a link from the output of the *PremiumDental* node to the second input of the *Add* function node.

50. Create a link from the output of the *Add* function node to the input of the *Premium* node.

51. Click the **Apply Mapping** button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 3.40).

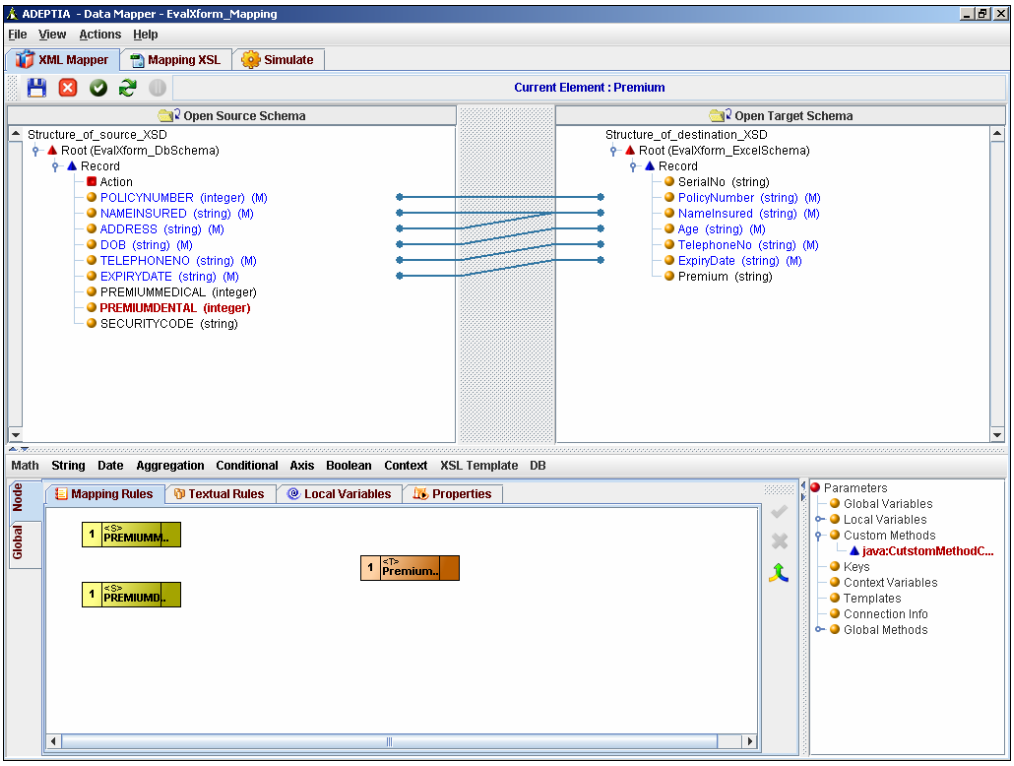



Figure 3.40: Map Premium Nodes

52. To filter the record based on the value of *PremiumMedical* field of source schema, **IFF Condition** is used. In this Process Flow, records in which the value of the *PremiumMedical* is less than US$ 30 are filtered and not passed to the target file.

53. To use **IFF condition**, click the **Record** Node in the Target Panel. The *Record* node is displayed in the Mapping Graph Area.

54. Double-click the **PremiumMedical** node in the Source Panel. The *PremiumMedical* node is displayed in the Mapping Graph Area.

55. To add the constant value (*30*), right-click the blank space in the Mapping Graph Area and select the **Constant** option. A *Constant* node is displayed in the Mapping Graph area.

56. Double-click the *Constan*t node. The Input dialog box is displayed (see Figure 3.41).

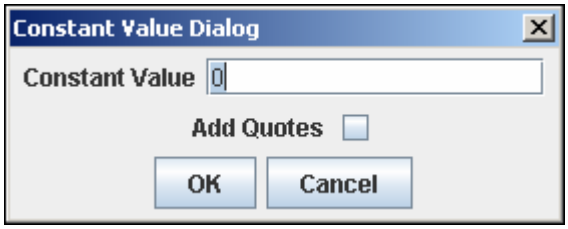



Figure 3.41: Input Dialog Box

57. Enter the required constant value (30) in the *Enter the Value* field and click the **OK** button. The entered value is displayed in the *Constant* node (see Figure 3.42).

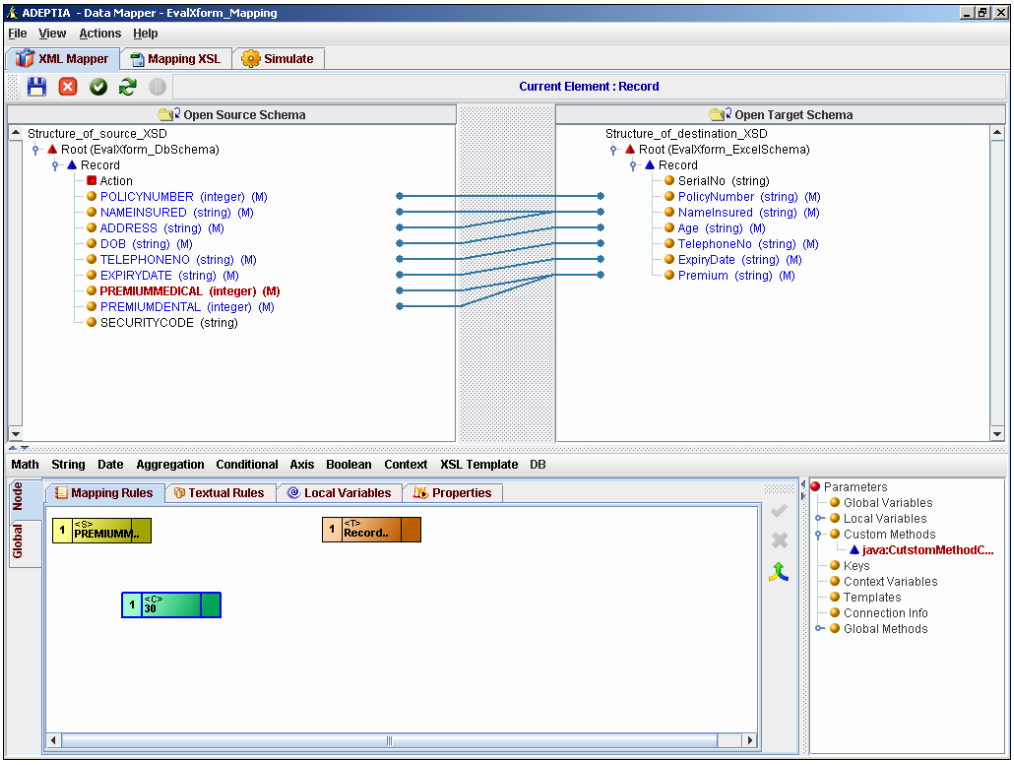



Figure 3.42: Constant Value Entered

58. Click the **Boolean** mapping function and select **Greater Than** (**>**) sub-function. The *Greater Than* function node is displayed in the Mapping Graph Area.

59. Create a link from the output of the *PremiumMedical* node to the first input of the *Greater Than* node.

60. Create another link from the *Constant* node to the second input of the *Greater Than* node (see Figure 3.43).

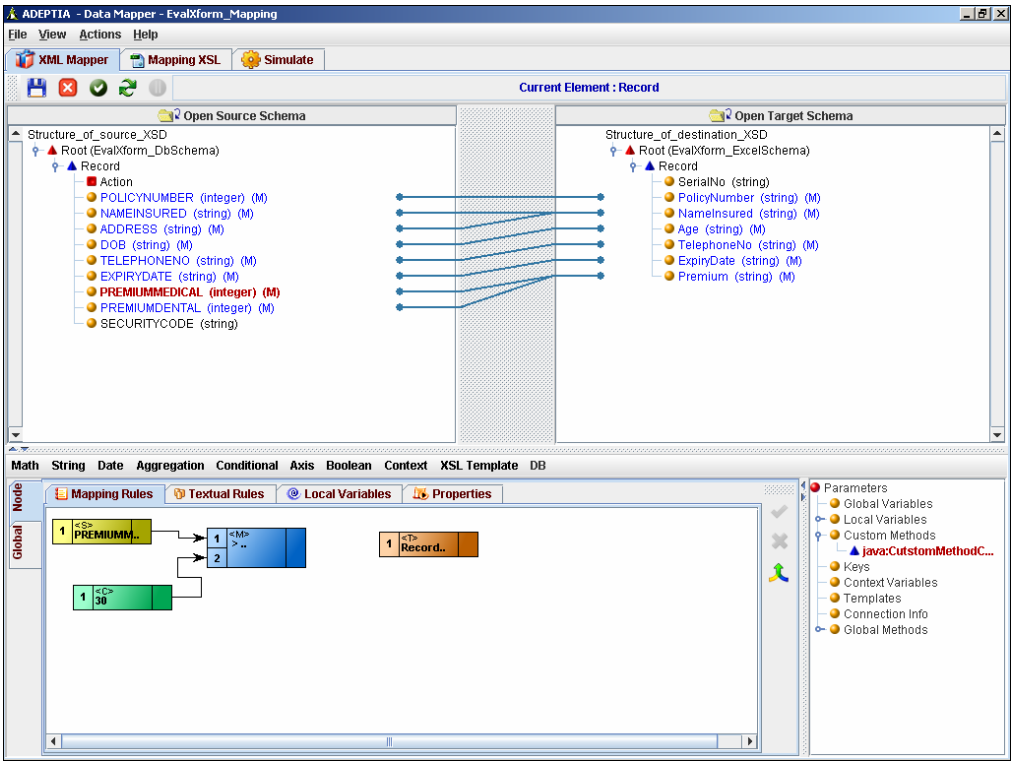



Figure 3.43: Create Links

61. Create another *Constant* node with the value ('*OK*').

| | |
|---|---|
| | While adding the value **OK**, check the *Add Quote* checkbox in *Constant Value* dialog box. |

62. Click the **Conditional** mapping function and select **IF CONDITION > For Filtering Records** sub-function. The *IFF Condition* node is displayed in the Mapping Graph Area (see Figure 3.44).

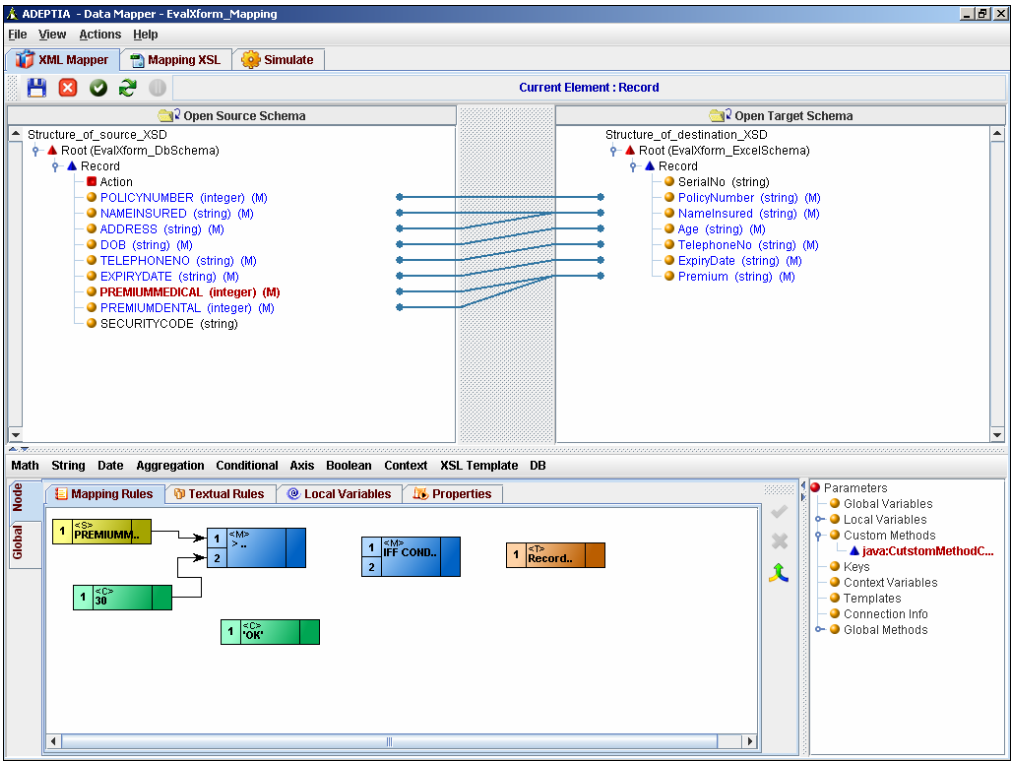



Figure 3.44: Select IFF Condition

63. Create a link from the output of the *Greater Than* function node to the first input of the *IFF Condition* node.

64. Create a link from the output of the *Constant* node (**'*OK*'**) to the second input of the *IFF Condition* node.

65. Create a link from the output of the *IFF Condition* node to the input of the *Record* node.

66. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 3.45).

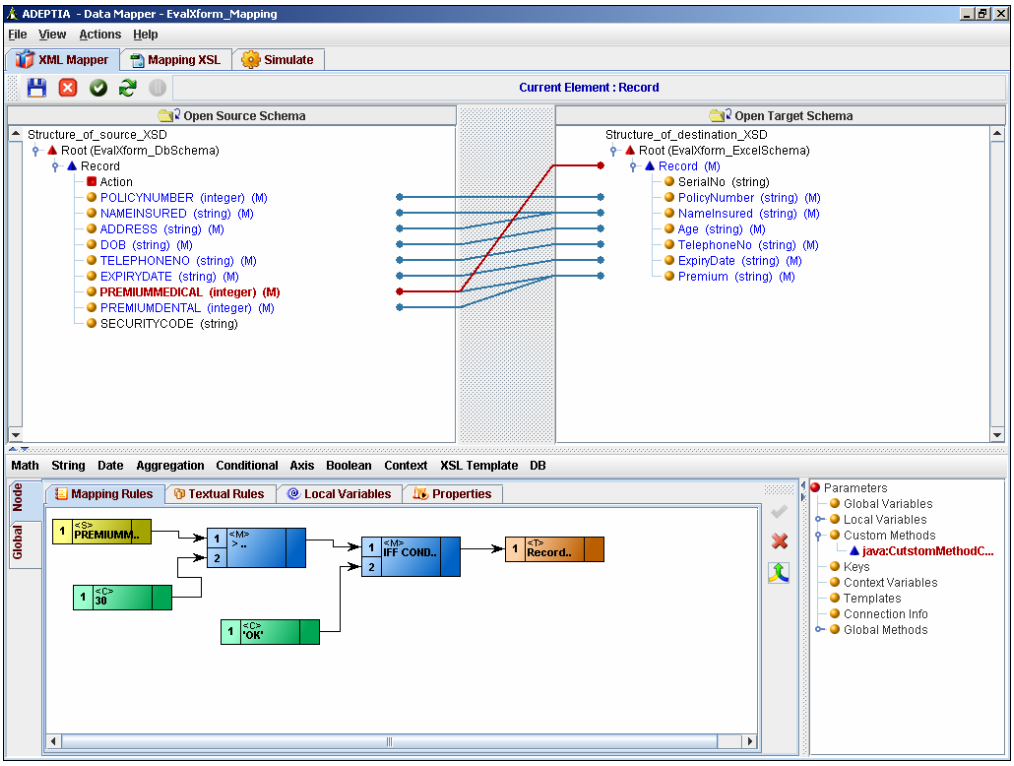



Figure 3.45: Map IFF Condition with Target Element

67. Save the mapping by clicking the **File** menu and selecting **Save**. A dialog box is displayed confirming that the mapping activity has been saved successfully.

| | |
|---|---|
| | Alternately, you can save the mapping activity by clicking **Save** ( ) button on the Tool Bar. |

68. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping (refer to Figure 7.18).

69. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

70. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.

71. Exit the Data Mapper applet by clicking **File** menu and selecting **Exit**.

# CREATING PROCESS FLOW

## (EvalXform_ProcessFlow)

A Process Flow is the set of activities arranged in a sequence to perform a specific task(s). It is created by combining various activities such as Source, Target, Schema or Transformer activities in a logical sequence. Process Designer is used to create a Process Flow. Process Designer has list of activities created. You only need to arrange them in a logical sequence and connect them with BPMN Flows.

***Steps to create Data Transformation Process Flow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Click the **New** link. The Create Process Flow screen is displayed (refer to Figure 7.28).
4. Enter the name and the description of the new Process Flow in the *Name* and *Description* fields respectively.
5. Select the logging level from the *Logging Level* drop-down list. There are four levels of logging. These are described in Table 7.9.
6. Select repository file retention from the *Repository File Retention* option. During execution Process Flow creates a temporary repository file to store the intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand sometime these repository files can be helpful in case of failure of Process Flow execution. For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files and target formatted data. By default repository files are being stored in the repository folder of Adeptia Suite. You can also choose an option to delete them or to archive them in a different location. There are four options for the Repository File Retention. These are described in Table 7.10.
7. Click the **Process Designer** button to open Process Designer. The Process Designer Screen is displayed (refer to Figure 7.29).
8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

| | |
|---|---|
|  | Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar. |

9. Click **[+] Activities** in Activities Panel, to expand the list of services and then click **[+] Source**. All the items in the **Source** category are displayed.
10. Click **[+] Database Source**. A list of existing Database Source activities is displayed.
11. Select **EvalXform_DBSource** and drag it to the Graph Canvas Area (see Figure 3.46).

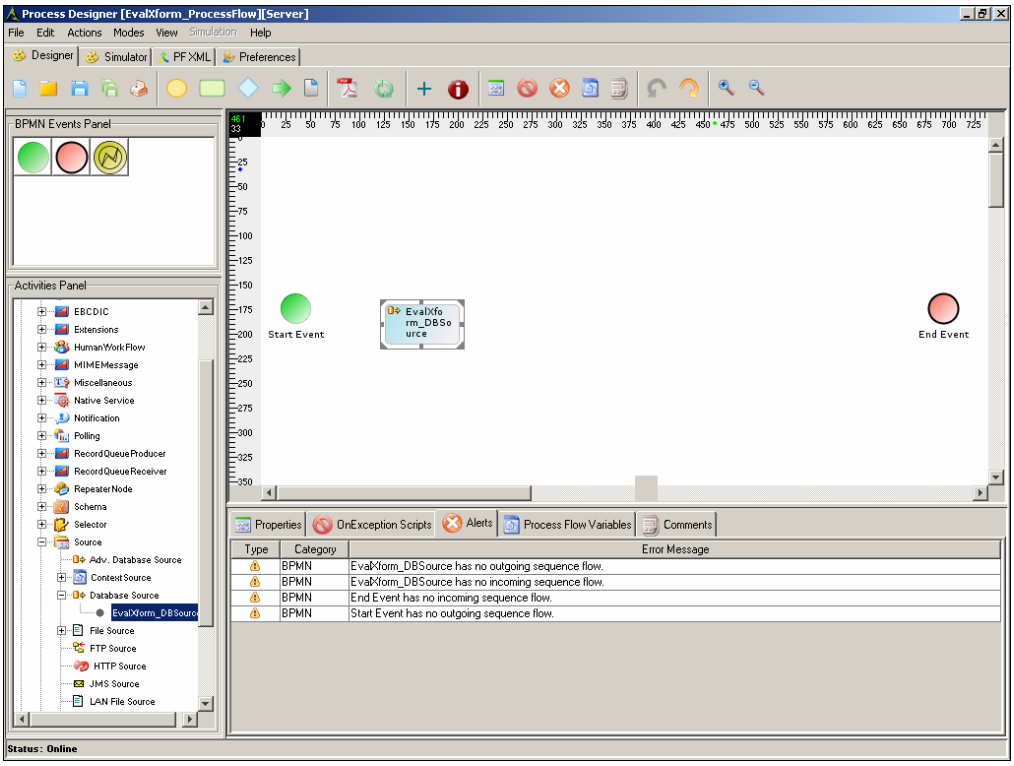



Figure 3.46: Drag Database Source to Graph Canvas Area

12. Similarly, click **[+] DataTransform** and then **[+] Data Transform**. Select **EvalXform_Mapping** activity and drag it to the Graph Canvas Area.
13. Click **[+] Schema** and then **[+] Excel Schema**. Select **EvalXform_ExcelSchema** activity and drag it to the Graph Canvas Area.
14. In Graph Canvas Area, right-click **EvalXform_ExcelSchema** and select **View Properties**. Properties of *EvalXform_ExcelSchema* are shown in the bottom pane (see Figure 3.47).

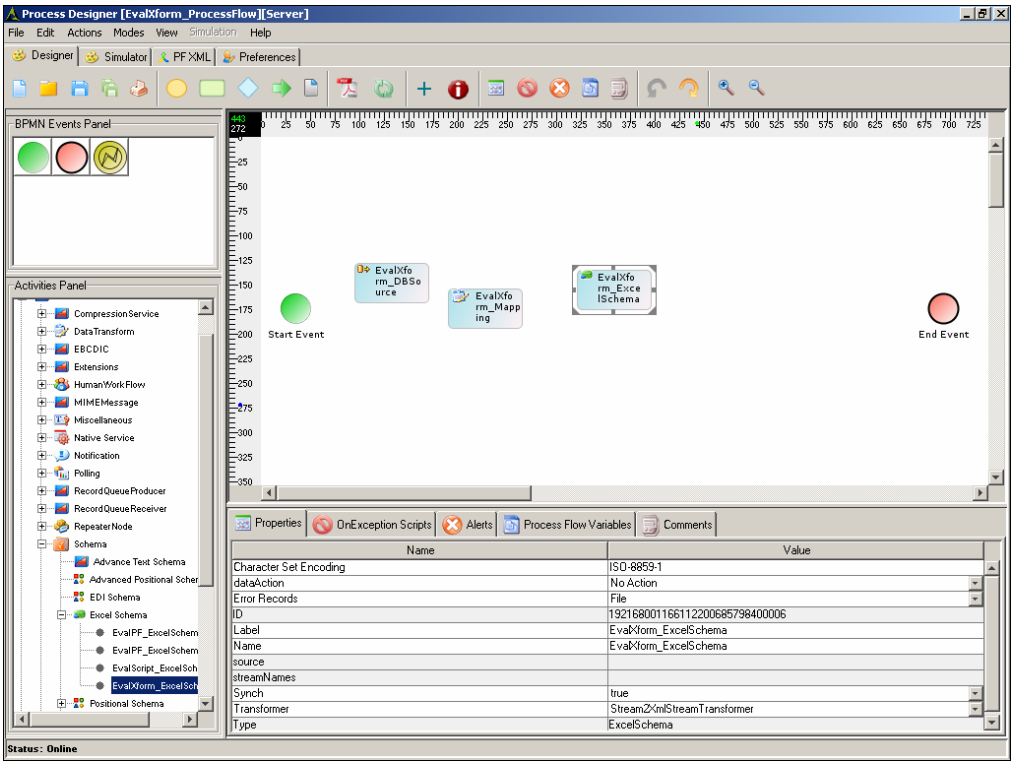



Figure 3.47: View Properties of *EvalXFormExcelSchema* Activity

15. The value of Transformer property is set to ***Stream2xmlStream Transformer***. Click the value box and change it to ***XmlStream2StreamTransformer***. When schema is used at target end, its transformer type must be **XmlStream2Stream Transformer**.
16. Click **[+] Target** and then **[+] File Target**. Select **EvalXform_FileTarget** activity and drag it to the Graph Canvas Area.
17. Click **[+] Native Service** and then **[+] Native Call**. Select ***EvalXform_NativeCall*** activity and drag it to the Graph Canvas Area.
18. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.

19. Click the **Sequence Flow** (⟶) icon. The Sequence flow is selected.
20. To connect *Start Event* with *EvalXform_DBSource,* drag mouse pointer from *Start* Event to *EvalXform_DBSource* (see Figure 3.48).

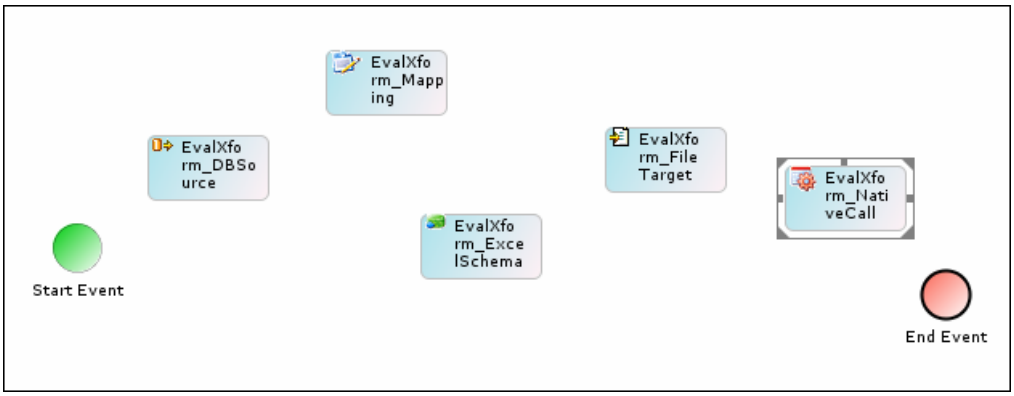



Figure 3.48: Connect Start Event to Database Source

21. Similarly, connect all other activities as shown in Figure 3.49

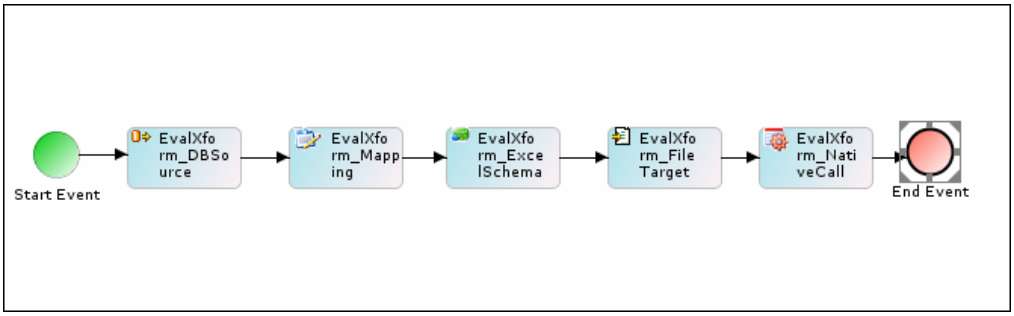



Figure 3.49: Connect all Activities

22. Save the Process Flow by clicking the **File** menu and selecting **Save Process Flow to Server**. A dialog box is displayed confirming that the *EvalXform_ProcessFlow* has been saved successfully. If the *comments* property is enabled, then clicking **Save Process Flow to Server** will display a screen where you need to enter comments related to creating the process flow (refer to Figure 7.37).
23. Enter the comments in the *Specify Comments for process flow customer* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

24. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.
25. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

# 4 SCRIPTED SERVICE PROCESS FLOW

This section describes the Scripted Service Process Flow.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This sample Process Flow demonstrates the use of Scripted Service and generation of Dynamic File Name. Scripted Service is used to call Java code to perform the specific task. In this Process Flow a ZIP file, which contains two different text files, is used as source. Scripted Service calls a Java code, which extracts both the file and concatenates them into single text file. This file is further converted into Excel file and then compressed into a ZIP file. Current date is appended with the name of the ZIP file, which is then saved in the specified target directory.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- File Source
- Scripted Service
- Text Schema
- Mapping
- Excel Schema
- Compression
- Put-Context-Var
- File Target

## DESCRIPTION

This sample Process Flow can be outlined as below:

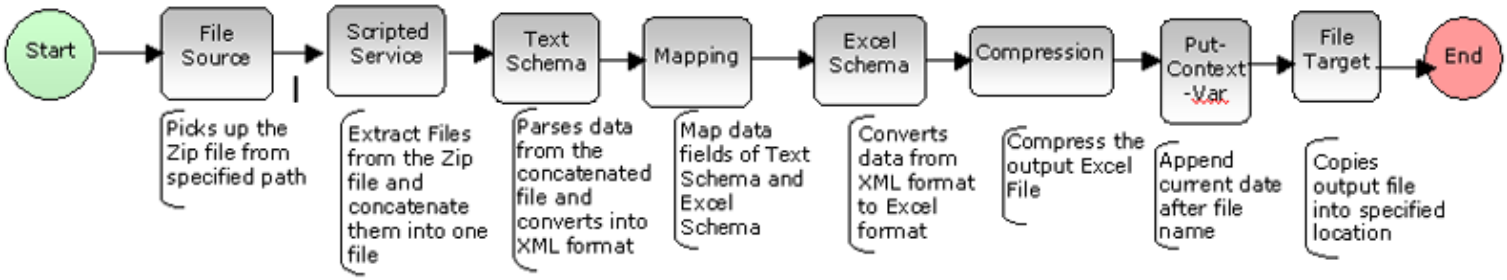


Figure 4.1: Flow Chart to Show Scripted Service Process Flow

## File Source (EvalScript_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow, a Zip files that contains two files, is specified as source.

## Scripted Service (EvalScript_ScriptedService)

Scripted Service is used to run any java code. You may want to perform some specific task, which may not be done by Adeptia Suite. In this case you can write your java code using scripted service. In this Process Flow, scripted service is used to extract files from a zip file and then concatenate them into a single text file.

## Text Schema (EvalScript_TextSchema)

Text Schema is used to parse the data received from the file source and convert it into intermediate XML format.

## Mapping (EvalScript_Mapping)

Mapping is used to map the data fields of Text Schema to the data fields of the Excel Schema. In this Process Flow simple one to one mapping is used.

## Excel Schema (EvalScript_ExcelSchema)

Excel Schema is used to convert data from intermediate XML format to Excel format.

## Compression

Compression is used to zip the output file. Compression is a Process Designer feature and used while creating the Process Flow. You do not have to create any such activity.

## Put-Context-Variable

Put-Context-Variable is used to append current date with name of output Excel file. Put-Context-Var is a Process Designer feature and used while creating the Process Flow. You do not have to create any such activity.

## File Target (EvalScript_FileTarget)

File Target is used to specify the name of the target zip file and the path, where the target zip file is to be saved.

## USAGE SCENARIO

This Process Flow can be used, whenever you want to run any Java code to perform specific task.

## DATA DESCRIPTION

Data used in this Process Flow contains record of employees of different departments of a company. At source end, a zip file is used which contains two different text files. Both the files are extracted from the zip file, concatenated and converted into Excel format.

The structure of text file used as source is outlined in the table below.

Table 4.1: Structure of Text File used as Source

| Field Name | Description | Data Type |
|---|---|---|
| Name | Name of the employee | String |
| Address | Address of the employee | String |
| EmailID | Email address of the employee | String |
| PhoneNo | Contact Number of the employee | Number |
| DateOfBirth | Date of Birth of the employee | Date |
| Department | Department of the employee | String |
| Salary | Salary of the employee | Number |
| DateOfJoining | Date of joining of the employee | Date |
| Designation | Designation of the employee | String |
| Age | Age of the employee | Number |

Name of the fields of the target file are same.

## EXECUTING AND MONITORING

This section describes the execution of sample Process Flow and its monitoring.

***Steps to execute the Process Flow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed.
3. Select the radio button adjacent to *EvalScript_ProcessFlow* Process Flow and then click **Execute** link. The View Process Flow Log screen is displayed (refer to Figure 3.2).
4. Click the link **Process Flow Logs** to view the status of the Process Flow execution. The Process Flow Logs is displayed (see Figure 4.2).

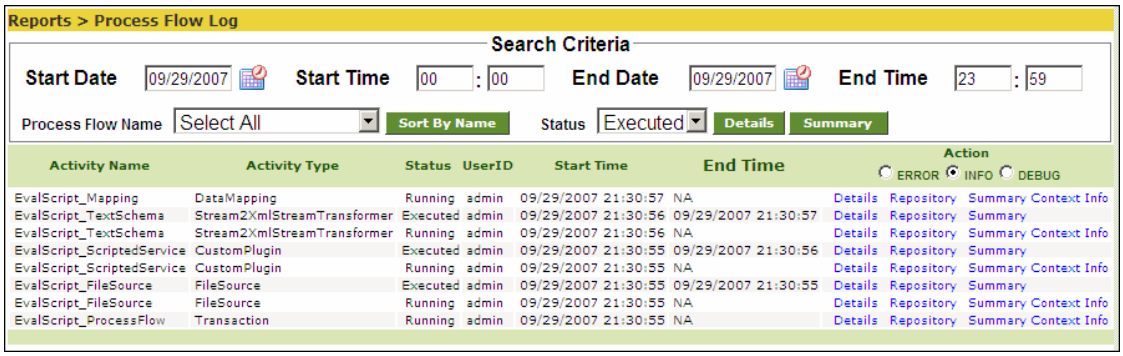



Figure 4.2: Searched Process Flows

To view the summary of all instances of the process flow execution, click the **Summary** button.

5. Click **Details** under the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 4.3).

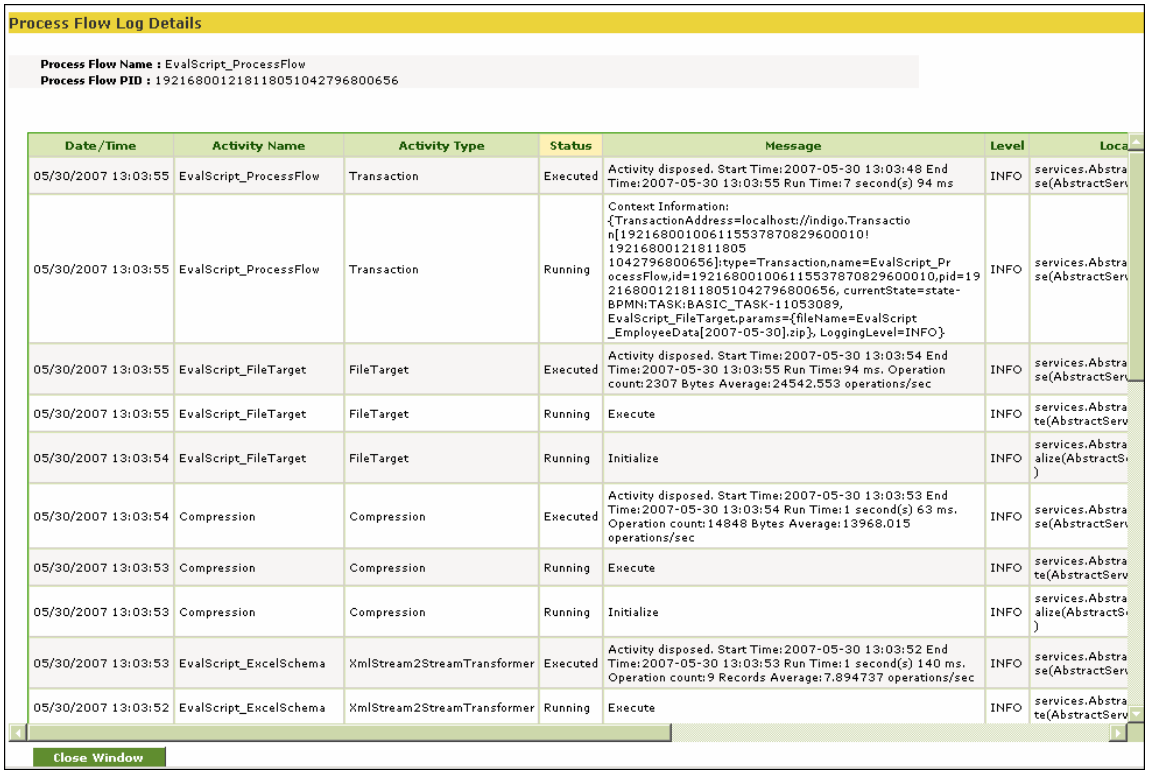



Figure 4.3: View Process Flow Log Details

# EDITING ACTIVITIES

The activities used in this sample Process Flow are pre-created. This section describes how to edit these activities.

## Editing File Source (EvalScript_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow, a Zip file (EvalScript_EmployeeData.zip) that contains two text files, is specified as source. This file is stored in **`../../Solutions/Demo/EvalScript/Source`** directory.

***Steps to edit the File Source:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **File**. The Manage File Source screen is displayed with the list of existing File Source activities (see Figure 4.4).

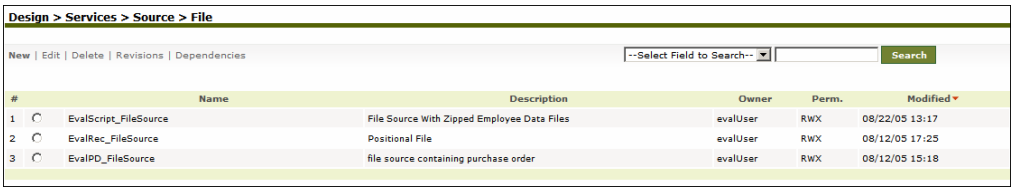



Figure 4.4: Manage File Source

4. Select the radio button adjacent to *EvalScript_FileSource* activity and then click **Edit** link. This displays the Edit *EvalScript_FileSource* activity screen, with the properties of the activity displayed in their respective fields (see Figure 4.5).

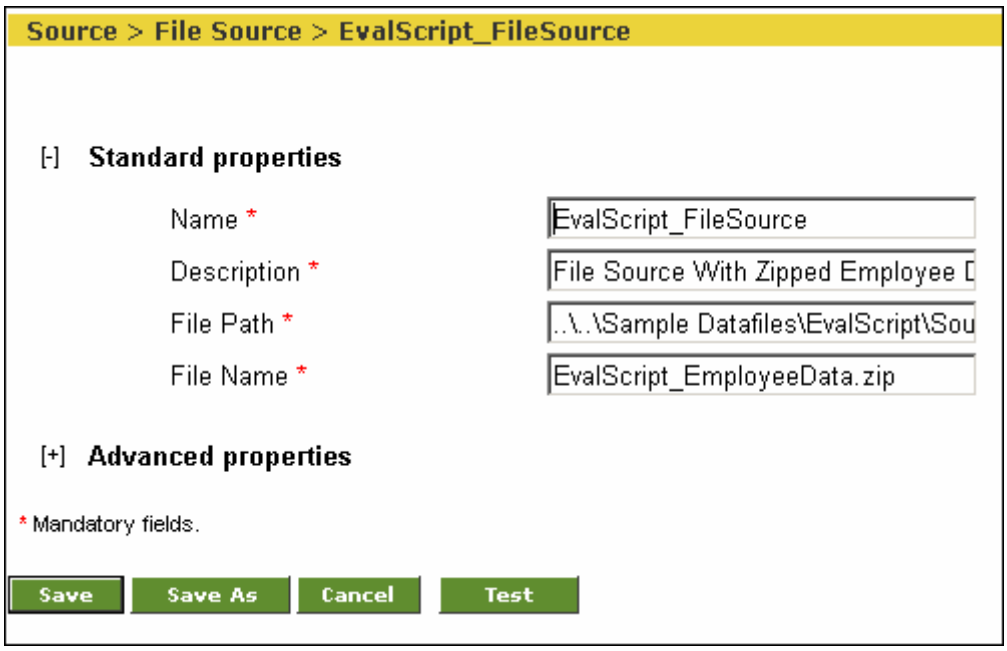



Figure 4.5: Edit *EvalScript_FileSource* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 4.2: Details of Fields on Edit File Source Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Source |
| Description | Description of the File Source |
| File Path | Source Path of the source file.<br>For example:<br>**../../Solutions/Demo/EvalScript/** |
| File Name | Name of the source file. For example: **EvalScript_EmployeeData.Zip** |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Source Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file source activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

8. Click **OK** to save the comments. This displays a screen confirming that the file source activity has been updated successfully.

| | |
|---|---|
|  | You can verify the file source activity at design time. For this, click **Test Connection**. This verifies the values in the *File Path* and *Filename* fields and checks whether the file actually exists in the specified location. |

## Editing Scripted Service (EvalScript_ScriptedService)

Scripted Service is used to call any Java Code to perform specific task. In this Process Flow, Scripted Service is used to extract two files from a Zip file and then concatenate them into single file. The Java Code used in the scripted service is displayed in Figure 4.6:

```
import com.adeptia.indigo.services;
import java.io.InputStream;
import java.util.zip.ZipEntry;
import java.util.zip.ZipFile;
import java.util.zip.ZipInputStream;
import java.io.BufferedInputStream;

byte[] data=new byte[1];

//creating ZipInputStream from the stream passed from file source
ZipInputStream zin = new ZipInputStream(new
BufferedInputStream(inputStream));

ZipEntry entry;

//extracting each zip entry from ZipInputStream and writing to output stream
while((entry = zin.getNextEntry()) != null) {

 //reading from the employee data files
    while ((count = zin.read(data)) != -1) {

//writing data into the output stream
        service.write(data ,"default");

    }

}
```

Figure 4.6: Sample JAVA Code

***Steps to edit the Scripted Service:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Extensions** to expand the tree, and then click **Custom Plugin**. The Manage Custom Plugin screen is displayed with the list of existing custom plugin activities (see Figure 4.7).

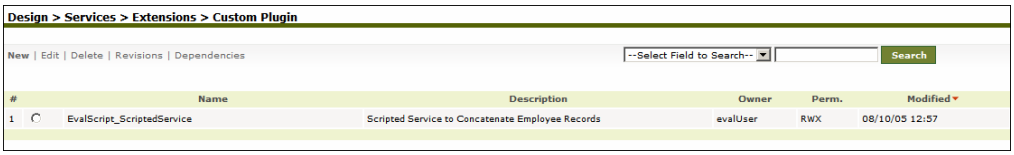



Figure 4.7: Manage Custom Plugins

4. Select the radio button adjacent to *EvalScript_ScriptedService* activity and then click **Edit** link. This displays the Edit *EvalScript_ScriptedService* activity screen, with the properties of the activity displayed in their respective fields (see Figure 4.8).

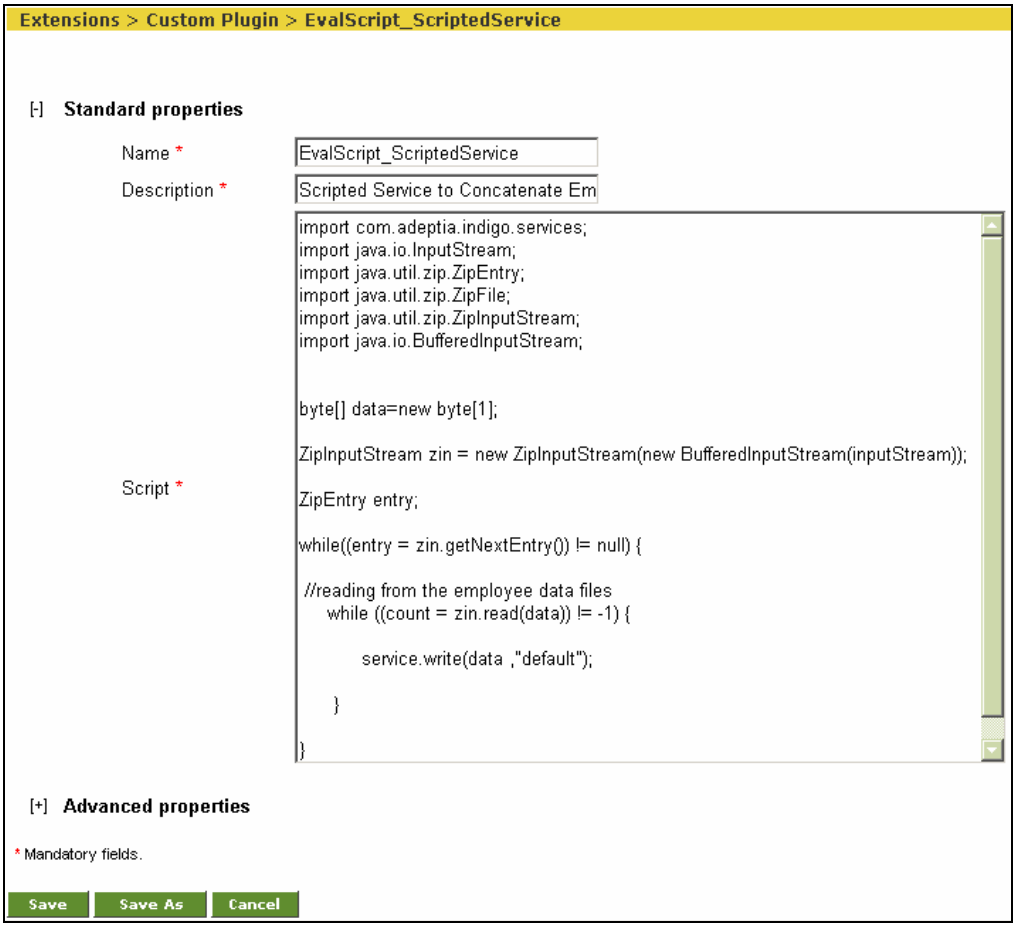



Figure 4.8: Edit *EvalScript_ScriptedService* Activity

The detailed description of fields on this screen is explicated in the table below.

Table 4.3: Details of Fields on Edit Custom Plugin Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Scripted Service |
| Description | Description of the Scripted Service |
| Script | Java Code that you want to run to perform the specific task |

5. Make the necessary changes.

6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the custom plugin activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the custom plugin activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the custom plugin activity has been updated successfully.

## Editing Text Schema (EvalScript_TextSchema)

Text Schema describes the structure of a text file. Text Schema activity is used to define how a text file is read or written in a predefined format. To create a Text Schema activity, you need to specify the record separator, field separator and the field names of the text file.

***Steps to edit the Text Schema:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed with the list of existing Text Schema activities (refer to Figure 7.12)
4. Select the radio button adjacent to *EvalScript_TextSchema* activity and then click **Edit** link. This displays the Edit *EvalScript_TextSchema* activity screen, with the properties of the activity displayed in their respected fields (see Figure 4.9).

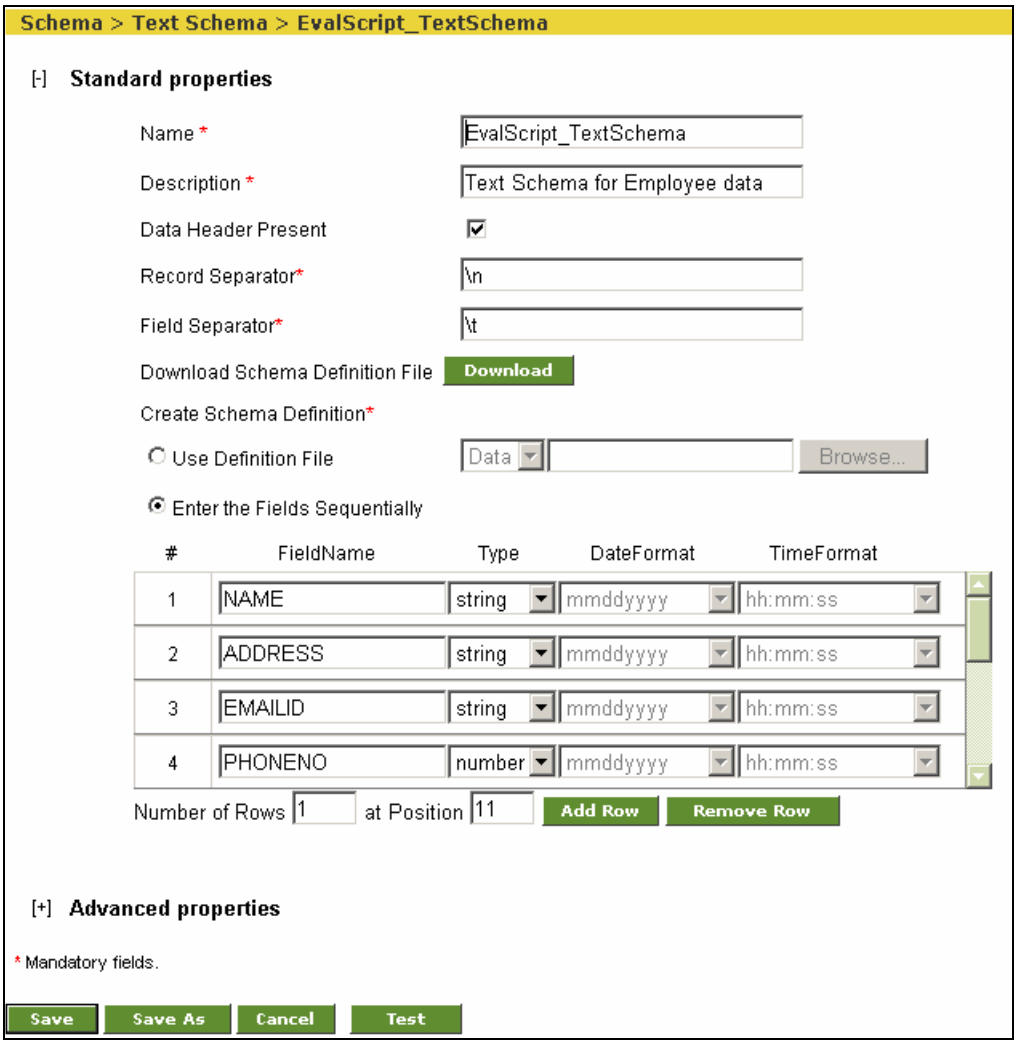



Figure 4.9: Edit *EvalScript_TextSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 4.4: Details of Fields on Edit Text Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Text Schema activity |
| Description | Description of the Text Schema activity |
| Data Header Present | Data Header usually contains the titles of the fields in a text file. If data header is present in the text file, check the **Data Header Present** checkbox |
| Record Separator | Character or set of characters that are used to mark the end of a record. For Example **\n** for **New Line**. |
| Field Separator | Character or set of characters that are used to separate fields. For example **\t** for **TAB** |
| Download Schema Definition File | To download a schema definition file, click **Download** button. Else, you can create a new schema definition. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File<br>▪ Enter the Fields Sequentially |

| | |
|---|---|
| | Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |
| Field Name | Name of the Fields |
| Data Type | There are three data types: |
| | String String can be used for any type of data.<br>Number Contains numbers<br>Date Contains Date and Time |
| Quotes Handling On | Suppose a character (say $) is specified as Field Separator in a record. Now any $ character in data field of that record (Chocolate$20$perpack) is considered as Field Separator, even though it is part of the data field. In the above example the $ after 20 is also considered as Field Separator, whereas it is data. To avoid this situation put those fields within the double quotes i.e. (Chocolate$"20$perpack") and check **Quotes Handling On** checkbox. Now characters within double quotes are considered as one Field even though there is a $ sign.<br>This option is available in *Advanced Properties* of Text Schema. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Text Schema activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the text schema activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the text schema activity has been updated successfully.

## Testing Text Schema

## (EvalScript_TextSchema)

You can verify the text schema activity at design time.

***Steps to verify schema activity***

1. Click **Test** button on the Edit Text Schema screen. The Test Schema screen is displayed (refer to Figure 7.14).
2. Select the type of schema to test, from the *Type* drop-down list. By default, *Source* is selected.

3. Enter the full path (with file name) of the source file in the *Source File Name* field.
4. Enter the full path of the XML target file, where it will be generated, in the *Target File Name* field.
5. Enter the full path of the XML file where errors will be stored, in the *Error File Name* field.
6. Click **Submit** button. This tests the validity of the text schema.

## Editing Mapping (EvalScript_Mapping)

Mapping is used to map data fields of source Schema with the data fields of target Schema. In this sample Process Flow simple one to one mapping is used.

***Steps to edit the Mapping activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed The Manage Data Mapping screen is displayed with the list of existing mapping activities (refer to Figure 7.15).
4. Select the radio button adjacent to *EvalScript_Mapping* activity and then click **Edit** link. This displays the Edit *EvalScript_Mapping* activity screen, with the name and description of the activity displayed in their respective fields (see Figure 4.10).

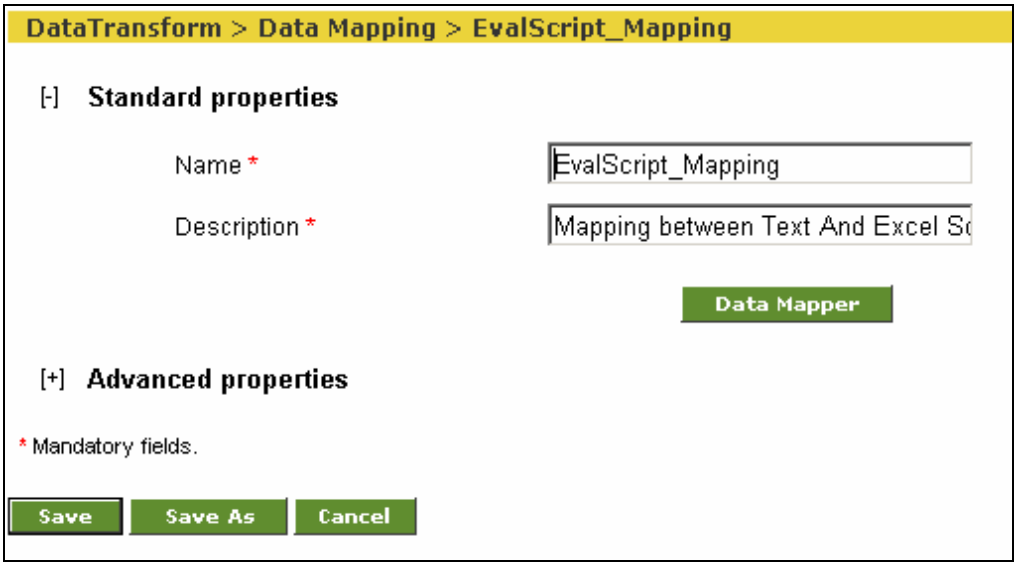



Figure 4.10: Edit *EvalScript_Mapping* Activity

5. Click the **Data Mapper** button. The Data Mapper applet is displayed showing mapping between the data fields of source and target schema (see Figure 4.11).

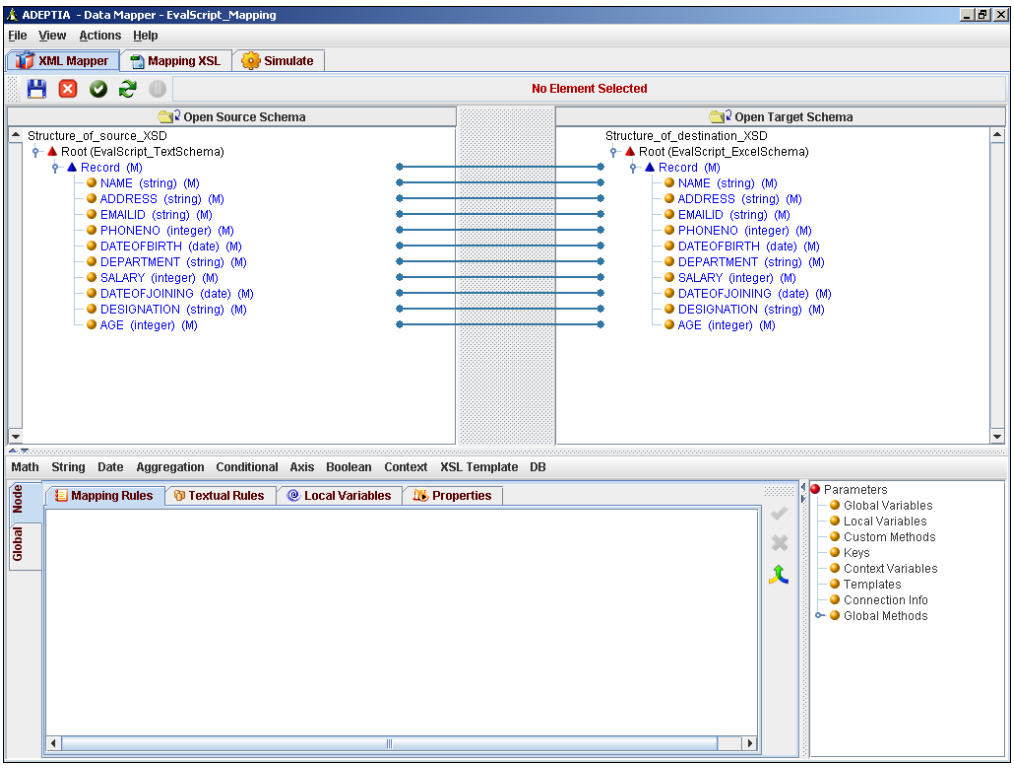



Figure 4.11: *EvalScript_Mapping* Activity in Data Mapper

6. Make the necessary changes to the mapping between the source and target schemas.

7. Once you have made the required changes, save the mapping by clicking **File** menu and selecting **Save**. A dialog box is displayed confirming that the Mapping activity has been saved successfully.

| | |
|---|---|
|  | Alternately, you can save the mapping activity by clicking **Save** (  ) button on the Tool Bar. |

8. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping (refer to Figure 7.18).

9. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

10. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.

11. Exit the Data Mapper applet by clicking **File** menu and selecting **Exit**.

## Editing Excel Schema (EvalScript_ExcelSchema)

Excel Schema defines the structure of Excel file. Excel Schema defines how the data can be read from an excel sheet or how it can be written into an excel sheet. In this sample Process Flow, Excel schema is used at the target end. At target end Excel Schema converts the data from an intermediate XML format into Excel format.

***Steps to edit the Excel Schema activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Excel**. The Manage Excel Schema screen is displayed with the list of existing Excel Schemas (refer to Figure 3.15).
4. Select the radio button adjacent to *EvalScript_ExcelSchema* activity and then click **Edit** link. This displays the Edit *EvalScript_ExcelSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 4.12).

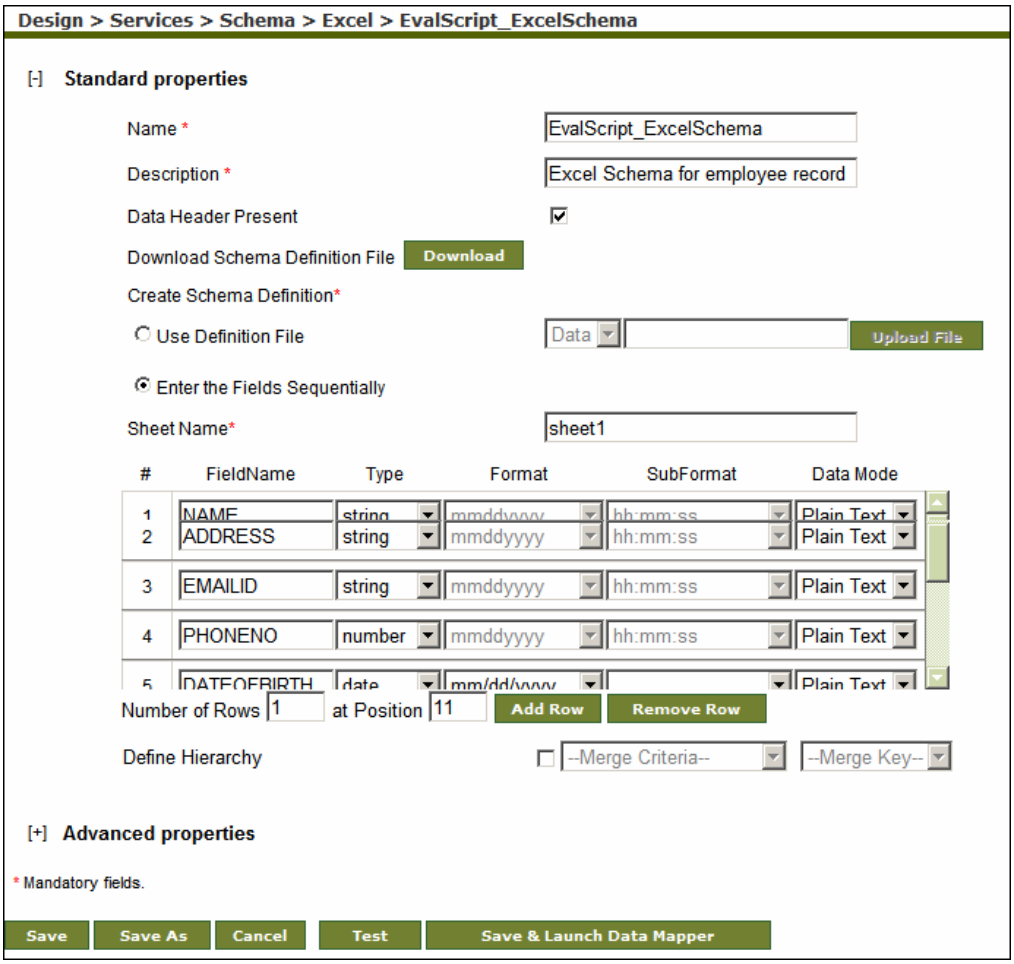



Figure 4.12: Edit *EvalScript_ExcelSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 4.5: Details of Fields on Edit Excel Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Excel Schema |
| Description | Description of the Excel Schema |
| Data Header Present | Data Header usually contains the titles of the fields in a file. If you enable this option, Field Names of Excel Schema are written as column's name in target excel file |
| Sheet Name | Name specified here becomes the Book Name of the excel sheet. |
| Download Schema Definition File | To download an existing schema definition file, click **Download** button. Else, you can create a new schema definition. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File<br>▪ Enter the Field Sequentially<br><br>Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |
| Field Name | Name of the Fields |

| | | |
|---|---|---|
| Data Type | There are three data types: | |
| | String | String can be used for any type of data. |
| | Number | Contains numbers |
| | Date | Contains Date and Time |
| | Currency | Contains Currency Value |
| Format | If data type is *Date*, select the data format from *Format* drop-down list. | |
| SubFormat | If data type is *Date*, select time format from *SubFormat* drop-down list. | |
| Data Mode | If data is in Plain Text, select *Plain Text* option. If data is encrypted, then select *Encrypted* option. | |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Excel Schema has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the excel schema (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the excel schema has been updated successfully.

## Editing File Target

## (EvalScript_FileTarget)

File target is used to specify the name of the target file and path, where the target file is saved. In this sample Process Flow, target file is saved in **`../../Solutions/Demo/EvalScript/`** directory**.**

***Steps to edit the File Target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **File**. The Manage File Target screen is displayed with the list of existing File Target activities (refer to Figure 3.17).
4. Select the radio button adjacent to *EvalScript_FileTarget* activity and then click **Edit** link. This displays the Edit *EvalScript_FileTarget* activity screen, with the properties of the activity displayed in their respective fields (see Figure 4.13).

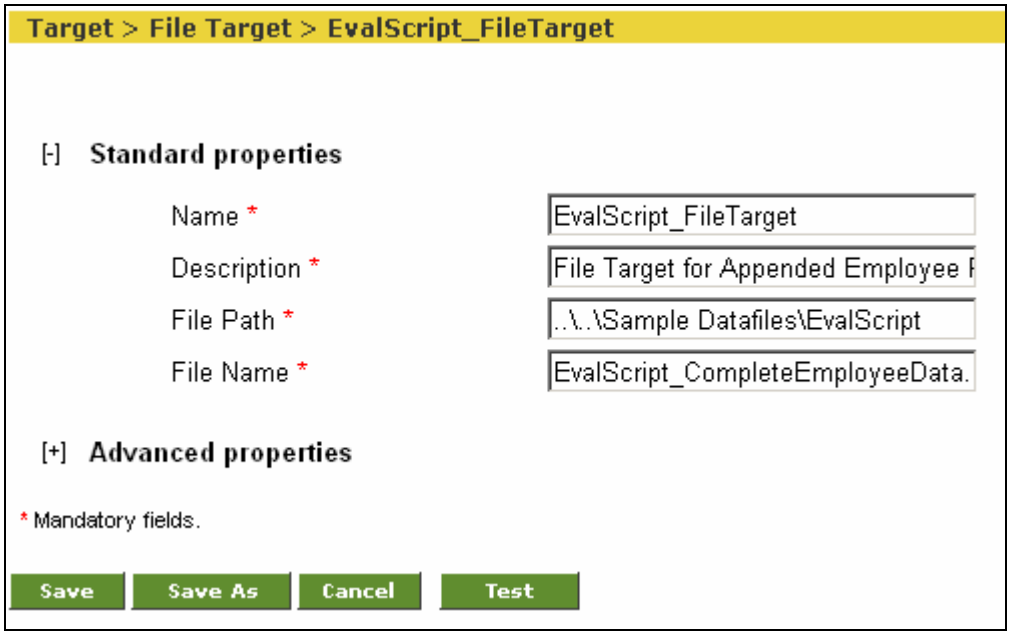



Figure 4.13: Edit *EvalScript_FileTarget* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.6 Details of Fields on Edit File Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Target |
| Description | Description of the File Target |
| File Path | Target Path and the target file.<br>For example:<br>**../../Solutions/Demo/EvalScript/** |
| File Name | Name of target file. For example:<br>**EvalScript_CompleteEmployeeData.zip** |

5. Make the necessary changes.
6. Once you have the made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file target activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

8. Click **OK** to save the comments. This displays a screen confirming that the file target activity has been updated successfully.

| | |
|---|---|
|  | You can verify the file target activity at design time. For this, click **Test Connection**. This verifies the values in the *Path* field and checks whether the file actually exists in the specified location. |

# CREATING PROCESS FLOW

## (EvalScript_ProcessFlow)

Process Flow is the set of activities arranged in a sequence to perform a specific task(s). Process Flow is created by combining various activities such as Source, Target, Schema or Transformer activities in a logical sequence. Process Designer is used to create a Process Flow. Process Designer has list of activities created. You only need to arrange them in a logical sequence and connect them with BPMN Flows.

***Steps to create EvalScript_ProcessFlow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Click the **New** link. The Create Process Flow screen is displayed (refer to Figure 7.28).
4. Enter the name and the description of the new Process Flow in the *Name* and *Description* fields respectively.
5. Select the logging level from the *Logging Level* drop-down list. There are four level of logging. They are described in Table 7.9.
6. Select repository file retention from the *Repository File Retention* option. During execution Process Flow creates a temporary repository file to store the intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand sometime these repository files can be helpful in case of failure of Process Flow execution. For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files and target formatted data. By default repository files are being stored in the repository folder of the Adeptia Suite. You can also choose an option to delete them or to archive them in a different location. There are four options for the Repository File Retention. They are described in Table 7.10.
7. Click the **Process Designer** button to open Process Designer. The Process Designer Screen is displayed (refer to Figure 7.29).
8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar.

9. Click **[+] Activities** in Repository View, to expand the list of services and then click **[+] Source**. All the items in the **Source** category are displayed.
10. Click **[+] File Source**. A list of existing File Source activities is displayed.
11. Select **EvalScript_FileSource** and drag it to the Graph Canvas Area (see Figure 4.14).

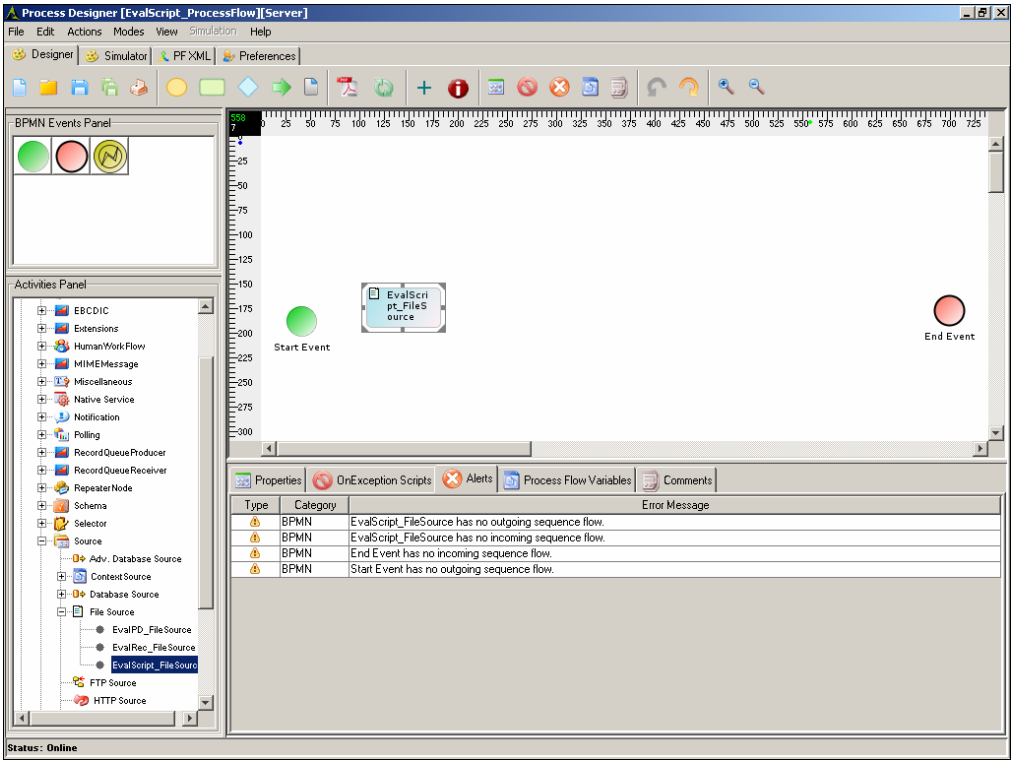



Figure 4.14: Drag File Source to Graph Canvas Area

12. Similarly, click **[+] Extensions** and then **[+] Custom Plugin**. Select **EvalScript_ScriptedService** activity and drag it to the Graph Canvas Area.
13. Click **[+] Schema** and then **[+] Text Schema**. Select **EvalSript_TextSchema** activity and drag it to the Graph Canvas Area.
14. Click **[+] DataTransform** and then **[+] Data Mapping**. Select **EvalScript_Mapping** activity and drag it to the Graph Canvas Area.
15. Click **[+] Schema** and then **[+] Excel Schema**. Select **EvalScript_ExcelSchema** activity and drag it to the Graph Canvas Area.
16. Right-click **EvalScript_ExcelSchema** in the Repository Viewl, and select **View Server Properties**. Properties of *EvalScript*_ExcelSchema are shown in the bottom pane (see Figure 4.15).

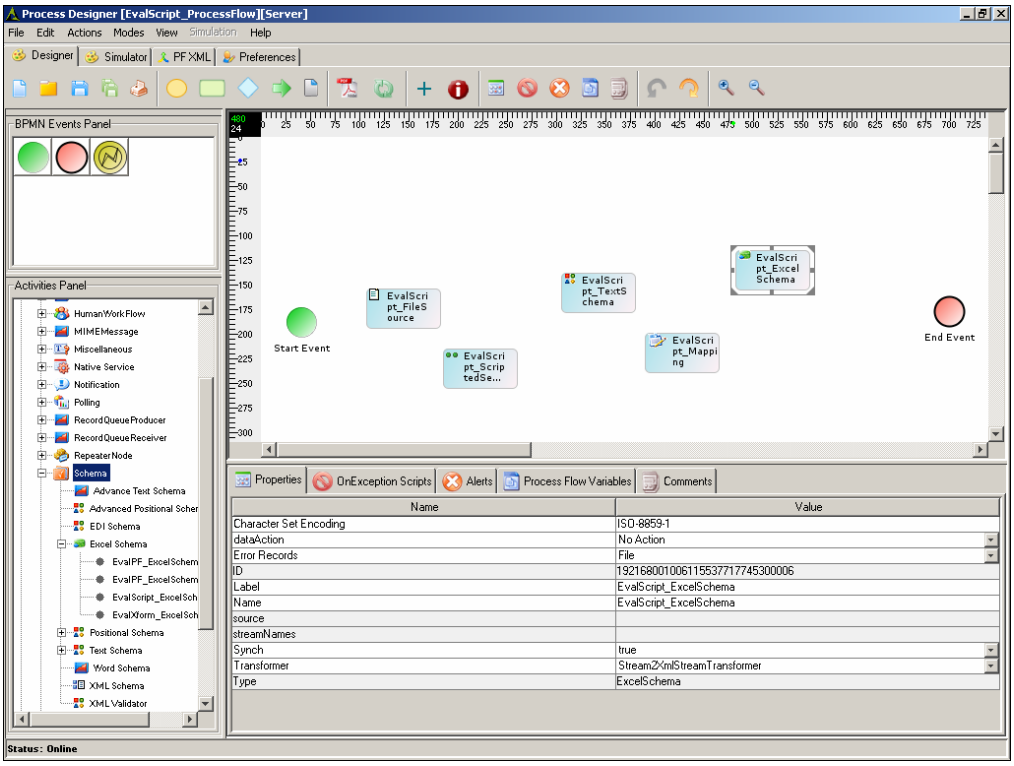



Figure 4.15: View Properties of *EvalScript_ExcelSchema* Activity

17. The value of Transformer property is set to ***Stream2xmlStream Transformer***. Click the value box and change it to ***XmlStream2StreamTransformer***. When schema is used at target end, its transformer type must be **XmlStream2Stream Transformer**.
18. Click **[+] Compression Service** to expand it. Select **Compression** and drag it to the Graph Canvas Area.
19. Click **[+] Action** in Repository View, to expand the list of Actions.

| | |
|---|---|
|  | An Action name is always unique. |

20. Select **Put-Context-Var** and drag it to the Graph Canvas Area (see Figure 4.16).

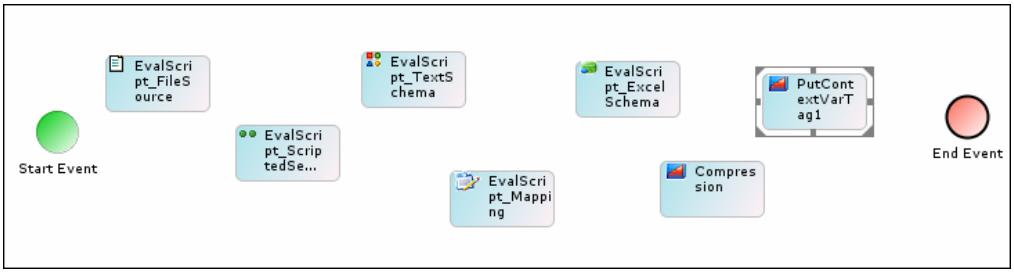



Figure 4.16: Drag Put-Context Var to Graph Canvas Area

21. Click **[+] Target** and then **[+] File Target**. Select **EvalScript_FileTarget** activity and drag it to the Graph Canvas Area.
22. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
23. Click the **Sequence Flow** (⟶) icon from the Palette. The Sequence flow is selected.
24. Drag mouse pointer from *Start Event* to *EvalScript_FileSource* to connect *Start Event* with *EvalScript_FileSource* (see Figure 4.17).

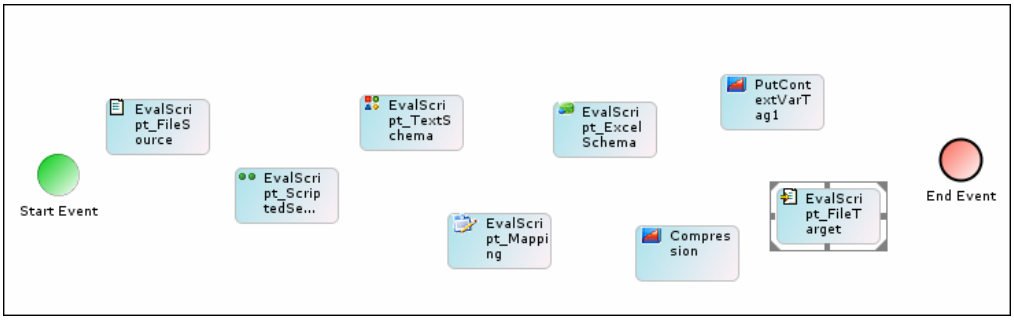



Figure 4.17: Connect Start Event to File Source

25. Similarly, connect all other activities as shown in Figure 4.18.

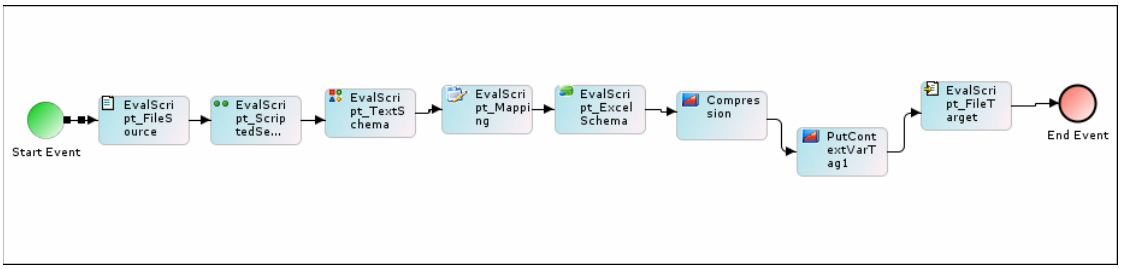



Figure 4.18: Connect all Activities

26. To add current date after the target file name, right-click **Put-Context-Var** and select **View Properties**. Properties of *Put-Context-Var* are shown in the Properties Panel (see Figure 4.19).

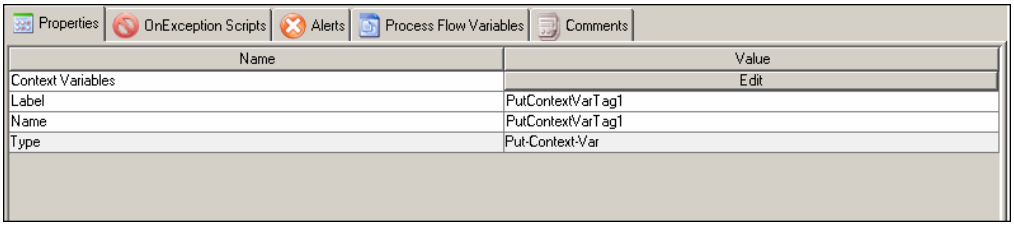



Figure 4.19: View Put-Context Var Properties

27. Click **Edit** from the value field of the *context Variable* properties. The *Edit Context Variable* screen is displayed (see Figure 4.20).

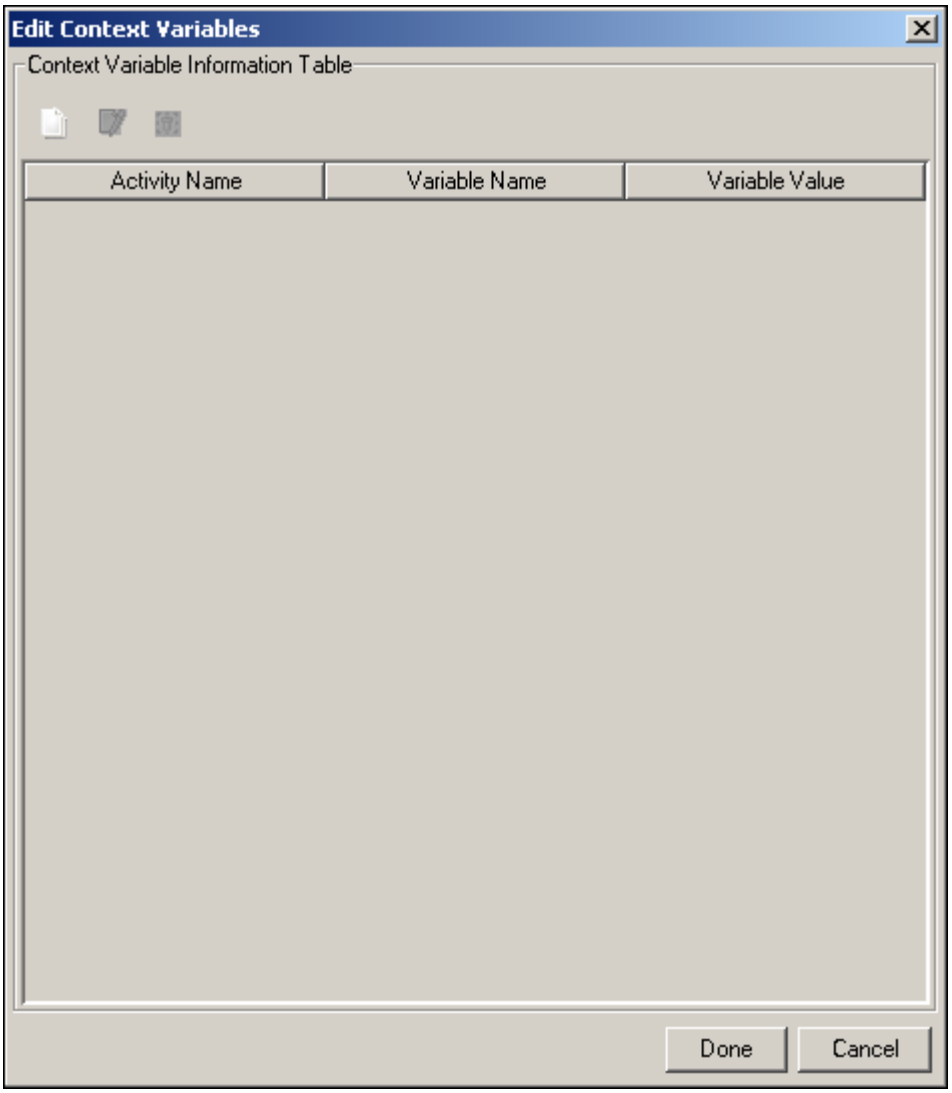



Figure 4.20: Edit Context Variable

28. Click **New variable** (  ) button. The *Context Variable Information* dialog box is displayed (see Figure 4.21).

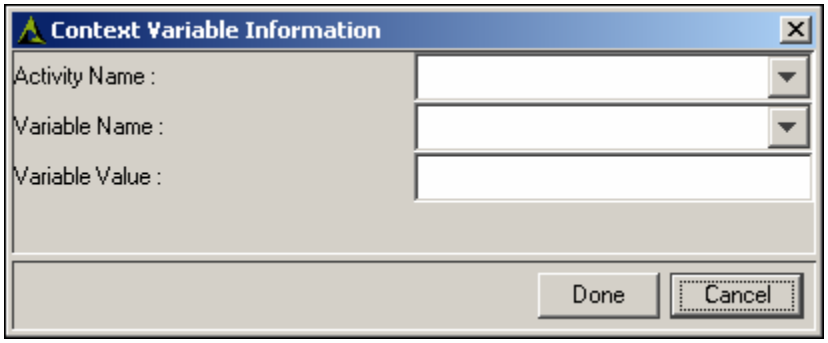



Figure 4.21: Add Context Variable

29. In the *Activity* drop-down list select the *EvalScript_FileTarget* activity.
30. Enter *filename* in the *Key* box.
31. Enter the name of the target file with date and time format in the *Value* box.

    For example: *EvalScript_CompleteEmployeeData[%%yyyy-mm-dd%%].zip*

If the target file is saved on 2005-02-05, name of the file will be EvalScript_EmployeeData[2005-02-05].zip.

32. Click **Done** twice to return to Graph Canvas Area.

Figure 6.18 shows only the control flow of the Process Flow. Control Flows only shows the order in which activities of a Process Flow is executed. It does not show the flow of data. For example as you can see in the figure 6.18, control flow from Compression goes to Put-Context-Var. But data is not passed from Compression to Put-Context-Var. Data is directly passed from Compression to EvalScript_FileTarget activity. To create the data flow, you need to create data stream between Compression and EvalScript_FileTarget.

33. To create data stream, right-click **Compression** and select **Multiple Stream**. The Multiple Stream dialog box is displayed (see Figure 4.22).

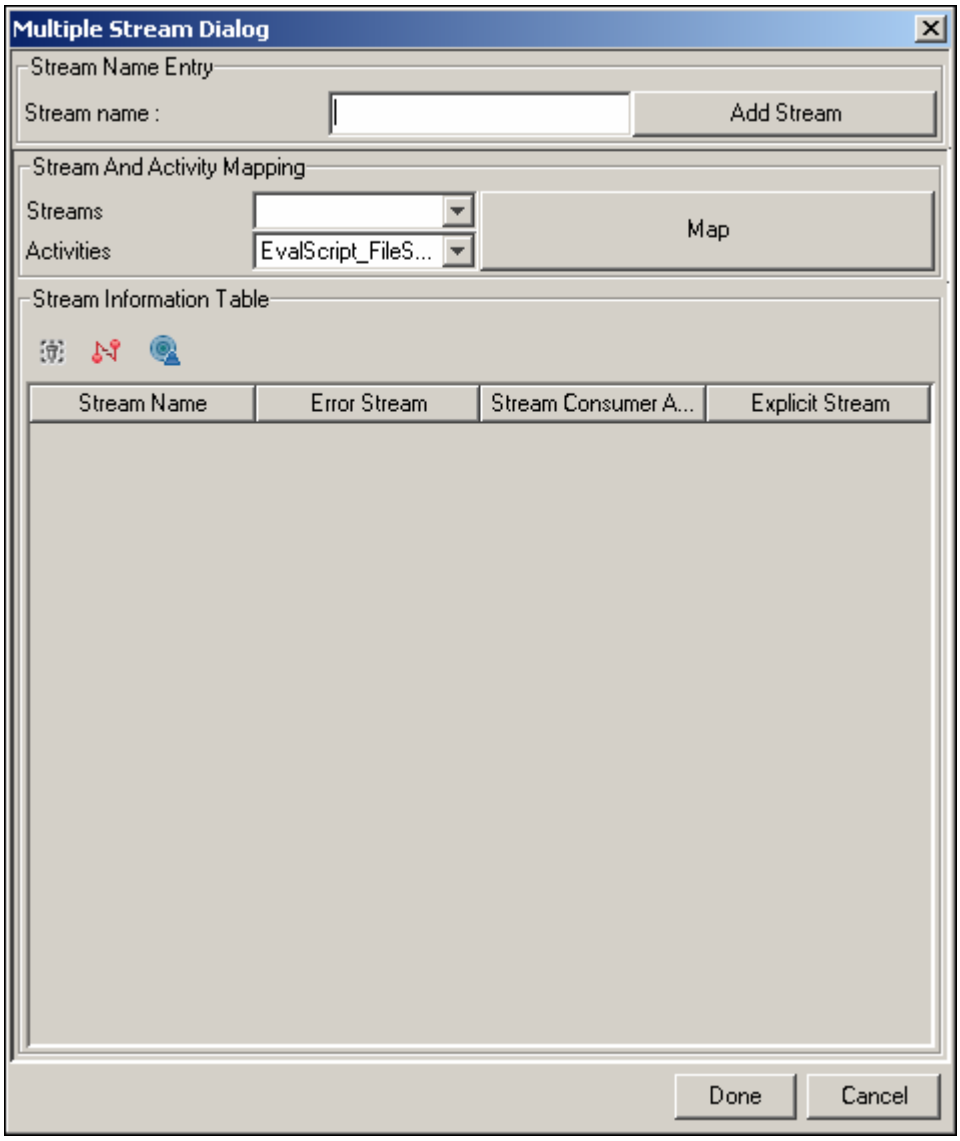



Figure 4.22: Multiple Stream Dialog Box

34. Select **EvalScript_FileTarget** from the *Activities* drop-down list and then click the **Default Stream** ( ) button.
35. Click the **Map** button. A stream between *Compression* and *EvalScript_FileTarget* is created (see Figure 4.23).

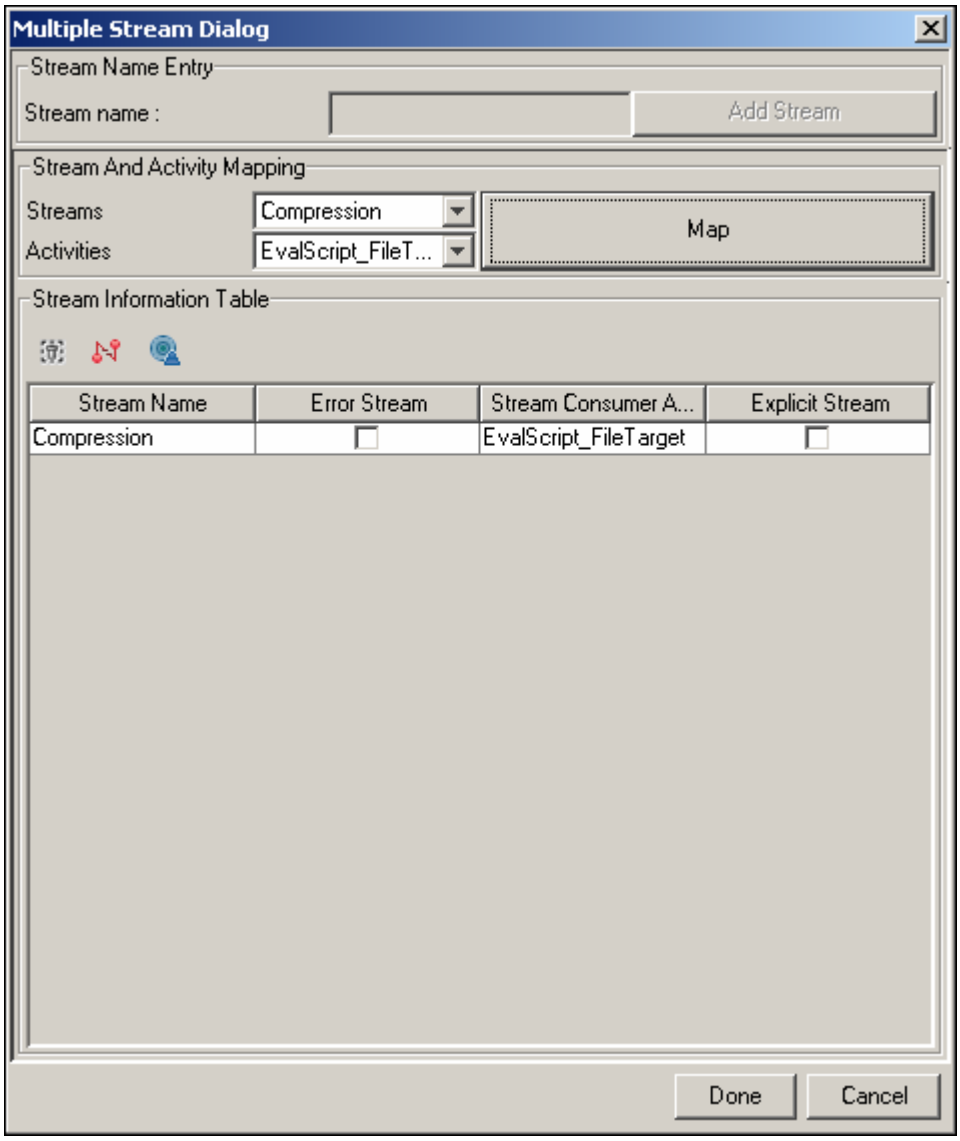



Figure 4.23: Define Data Stream

36. Check the *Explicit Stream* checkbox.

37. Click **Done** button to close the Multiple Stream Dialog box. Data stream created is shown in the Graph Canvas area (see Figure 4.24).

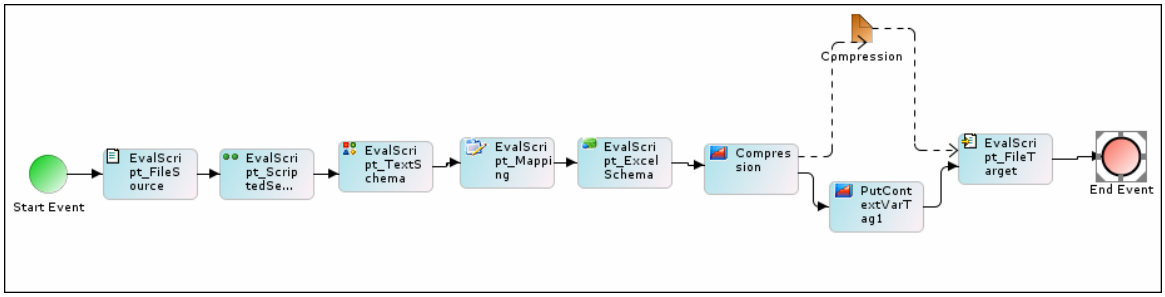



Figure 4.24: Data Stream Created

38. Save the Process Flow by clicking **File** menu and selecting **Save Process Flow to Server**. A dialog box is displayed confirming that the *EvalScript_ProcessFlow* has been saved successfully. If the *comments* property is enabled, then clicking **Save Process Flow to Server** will display a screen where you need to enter comments related to creating the process flow (refer to Figure 7.37).
39. Enter the comments in the *Specify Comments for process flow* field.

The comment should be at least 1 character in length.

40. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.
41. Click the **Done** button to close the dialog box.
42. Exit the Process Designer by clicking **File** menu and selecting **Exit**.

# 5 PROCESS DESIGNER PROCESS FLOW

This section describes the Process Designer Process Flow.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This Process Flow demonstrates the use of different features of Process Designer. In this Process Flow, a purchase order, which is in text format, is used as source. Values of the purchase order (i.e. PO Number, Amount, Company Name, and Item) are separated by comma (,). A copy of the purchase order is saved into a specified folder. Another copy of the purchase order is sent to either of the Manager or Director of the company through email. If the purchase amount is less than US$ 50000, the purchase order is sent to the Manager for approval. If the purchase amount is greater than US$ 50000, the purchase order is sent to the Director for approval.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- File Source
- Context Target
- Context Source
- Repeater
- File target
- Decision Node
- Mail Target

# DESCRIPTION

This sample Process Flow can be outlined as below:

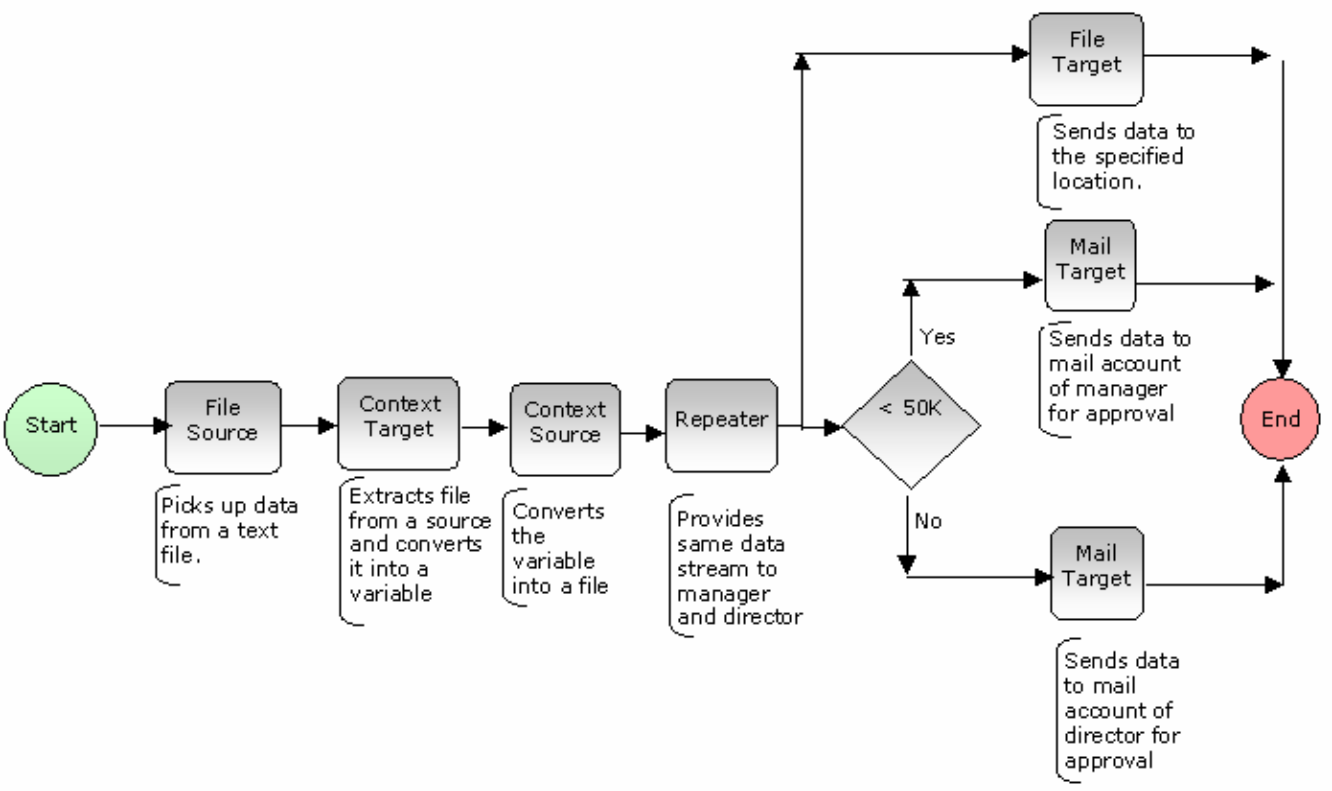



Figure 5.1: Flow Chart to show Process Flow

## File Source (EvalPD_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow, a text file containing purchase order is specified as source.

## Context Target

Context Target is used to put the data of the text file into the context of the Process Flow. From the context of the Process Flow, decision node picks up the amount and decides whether to send the data to Manager or to the Director.

## Context Source

Context Source is used to take the data from the context of the Process Flow and to send it to the decision node.

### Repeater

Repeater node is used to send the same stream to more than one activity. In this Process Flow repeater node is used to send stream, coming from Context Source, to the File Target activity and Mail Targets.

Context Target, Context Source and Repeater Node are Process Designer features and you do need to create any such activities.

### File Target (EvalPD_FileTarget)

File Target is used to specify target Text file name with full path, where the target Text file is to be saved. In this Process Flow, file target is used to save a copy of the purchase order in the specified location for future use.

### Mail Targets (EvalPD_MailTargetLessThan50k & EvalPD_MailTargetGreaterThan50k)

Mail Target is used to send the data to the specified email address. In this Process Flow two Mail Targets are used to send purchase order to either the manager or the director, for approval, depending upon value of purchase amount.

## USAGE SCENARIO

This Process Flow can be used, whenever you want to process a purchase order based on the value of purchase amount.

## DATA DESCRIPTION

Data used in this Process Flow is a purchase order in a text file format. This text file contains following information separated by comma (,).

The structure of Text File used as Source is displayed in the table below.

Table 5.1: Structure of Text File used as Source

| Field Name | Description | Data Type |
|---|---|---|
| PO# | Purchase Order number | String |
| PO_AMOUNT | Purchase Amount | Number |
| COMPANY | Name of the company | String |
| ITEM | Name of the item to be purchased. | String |

| | |
|---|---|
| | The Target file also has the same fields as shown in Table 8.1. |

## PREREQUISITES

Mail Target activities must be edited before executing the Process Flow. These activities are outlined as:

- EvalPD_MailTargetLessThan50k
- EvalPD_MailTargetGreaterThan50k

| | |
|---|---|
| | To know, how to edit these activities refer to section **Editing Activities**. |

## EXECUTION AND MONITORING

This section describes the execution of sample Process Flow and monitoring Process Flow execution.

***Steps to execute the Process Flow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Select the radio button adjacent to *EvalPD_ProcessFlow* Process Flow and then click **Execute** link. The View Process Flow Log screen is displayed (refer to Figure 3.2).
4. Click the link **Process Flow Logs** link to view the status of the Process Flow execution. The Process Flow Log is displayed (see Figure 5.2).

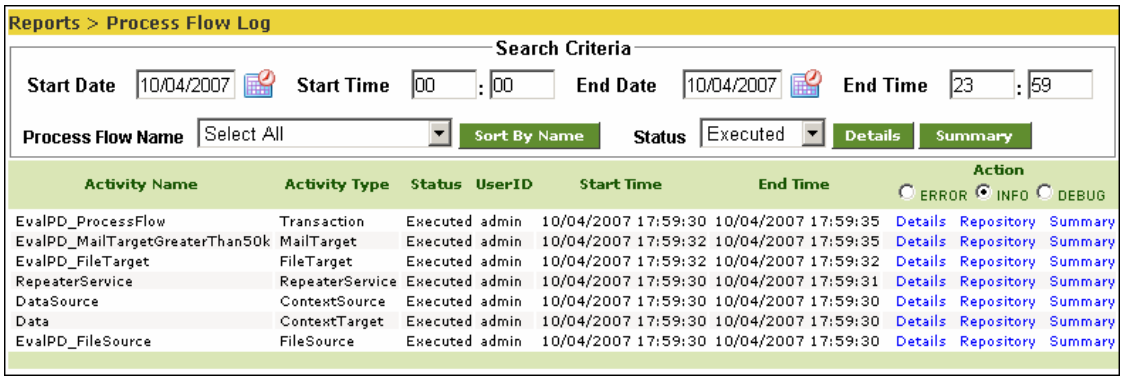
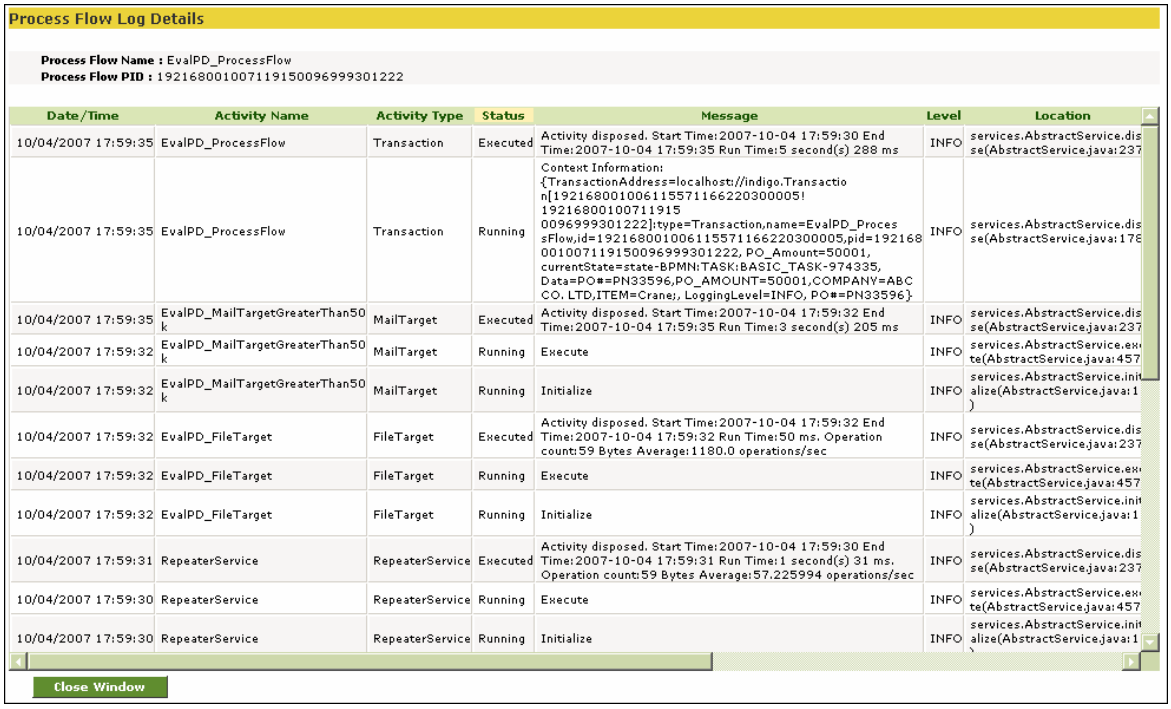



Figure 5.2: View Searched Process Flows

| | |
|---|---|
| | To view the summary of all instances of the process flow execution, click the **Summary** button. |

5. Click **Details** under the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 5.3).



Process Flow Log Details

**Process Flow Name :** EvalPD_ProcessFlow
**Process Flow PID :** 19216800100711915009699930122 2

| Date/Time | Activity Name | Activity Type | Status | Message | Level | Location |
|---|---|---|---|---|---|---|
| 10/04/2007 17:59:35 | EvalPD_ProcessFlow | Transaction | Executed | Activity disposed. Start Time:2007-10-04 17:59:30 End Time:2007-10-04 17:59:35 Run Time:5 second(s) 288 ms | INFO | services.AbstractService.dis se(AbstractService.java:237 |
| 10/04/2007 17:59:35 | EvalPD_ProcessFlow | Transaction | Running | Context Information: {TransactionAddress=localhost://indigo.Transactio n[19216800100611557116622030000 5!1921680010071191500969993012 22]:type=Transaction,name=EvalPD_Proces sFlow,id=19216800100611557116622030000 5,pid=19216800100711915009699930122 2, PO_Amount=50001, currentState=state-BPMN:TASK:BASIC_TASK-974335, Data=PO#=PN33596,PO_AMOUNT=50001,COMPANY=ABC CO. LTD,ITEM=Crane;, LoggingLevel=INFO, PO#=PN33596} | INFO | services.AbstractService.dis se(AbstractService.java:178 |
| 10/04/2007 17:59:35 | EvalPD_MailTargetGreaterThan50 k | MailTarget | Executed | Activity disposed. Start Time:2007-10-04 17:59:32 End Time:2007-10-04 17:59:35 Run Time:3 second(s) 205 ms | INFO | services.AbstractService.dis se(AbstractService.java:237 |
| 10/04/2007 17:59:32 | EvalPD_MailTargetGreaterThan50 k | MailTarget | Running | Execute | INFO | services.AbstractService.exe te(AbstractService.java:457 |
| 10/04/2007 17:59:32 | EvalPD_MailTargetGreaterThan50 k | MailTarget | Running | Initialize | INFO | services.AbstractService.init alize(AbstractService.java:1 ) |
| 10/04/2007 17:59:32 | EvalPD_FileTarget | FileTarget | Executed | Activity disposed. Start Time:2007-10-04 17:59:32 End Time:2007-10-04 17:59:32 Run Time:50 ms. Operation count:59 Bytes Average:1180.0 operations/sec | INFO | services.AbstractService.dis se(AbstractService.java:237 |
| 10/04/2007 17:59:32 | EvalPD_FileTarget | FileTarget | Running | Execute | INFO | services.AbstractService.exe te(AbstractService.java:457 |
| 10/04/2007 17:59:32 | EvalPD_FileTarget | FileTarget | Running | Initialize | INFO | services.AbstractService.init alize(AbstractService.java:1 ) |
| 10/04/2007 17:59:31 | RepeaterService | RepeaterService | Executed | Activity disposed. Start Time:2007-10-04 17:59:30 End Time:2007-10-04 17:59:31 Run Time:1 second(s) 31 ms. Operation count:59 Bytes Average:57.225994 operations/sec | INFO | services.AbstractService.dis se(AbstractService.java:237 |
| 10/04/2007 17:59:30 | RepeaterService | RepeaterService | Running | Execute | INFO | services.AbstractService.exe te(AbstractService.java:457 |
| 10/04/2007 17:59:30 | RepeaterService | RepeaterService | Running | Initialize | INFO | services.AbstractService.init alize(AbstractService.java:1 ) |

Close Window

Figure 5.3: View Process Flow Log Details

# EDITING ACTIVITIES

Activities used in this sample Process Flow are pre-created. This section describes how to edit these activities.

## File Source

## (EvalPD_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow a text file **(EvalPD_FileSource.txt)** is specified as source. This file is stored in **../../Solutions/Demo/EvalPD/** directory.

***Steps to edit the File Source:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.

3. Click **[+] Source** to expand the tree, and then click **File**. The Manage File Source screen is displayed with the list of existing File Source activities (refer to Figure 4.4).

4. Select the radio button adjacent to *EvalPD_FileSource* activity and then click **Edit** link. This displays the Edit *EvalPD_FileSource* activity screen, with the properties of the activity displayed in their respective fields (see Figure 5.4).

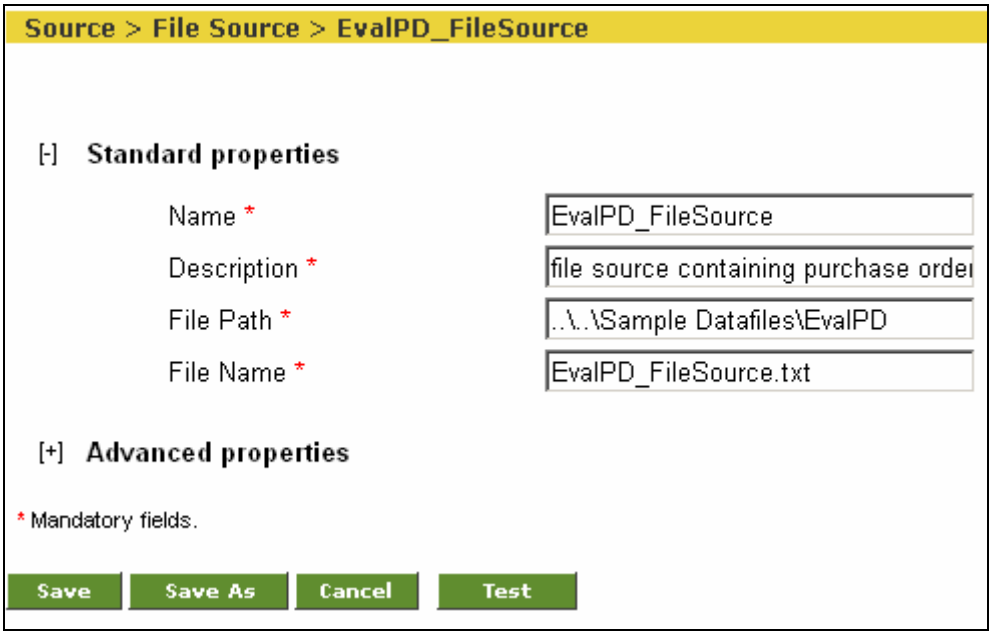



Figure 5.4: Edit *EvalPD_FileSource* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 5.2: Details of Fields on Edit File Source Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Source |
| Description | Description of the File Source |
| File Path | Path of the source file.<br>For example:<br>**../../Solutions/Demo/EvalPD/** |
| File Name | Name of the source file. For example : **EvalPD_FileSource.txt** |

5. Make the necessary changes.

6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Source Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file source activity (refer to Figure 7.10).

7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the file source activity has been updated successfully.

| | |
|---|---|
|  | You can verify the file source activity at design time. For this, click **Test Connection**. This verifies the values in the *File Path* and *Filename* fields and checks whether the file actually exists in the specified location. |

## Editing File Target (EvalPD_FileTarget)

File target is used to specify the name of the target file and path, where the target file is saved. In this sample Process Flow, target file is saved in **`../../Solutions/Demo/EvalPD/EvalPD_FileTarget.txt`** directory**.**

***Steps to edit the File Target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **File**. The Manage File Target screen is displayed with the list of existing File Target activities (refer to Figure 3.17).
4. Select the radio button adjacent to *EvalPD_FileTarget* activity and then click **Edit** link. This displays the Edit *EvalPD_FileTarget* activity screen, with the properites of the activity displayed in their respective fields (see Figure 5.5).

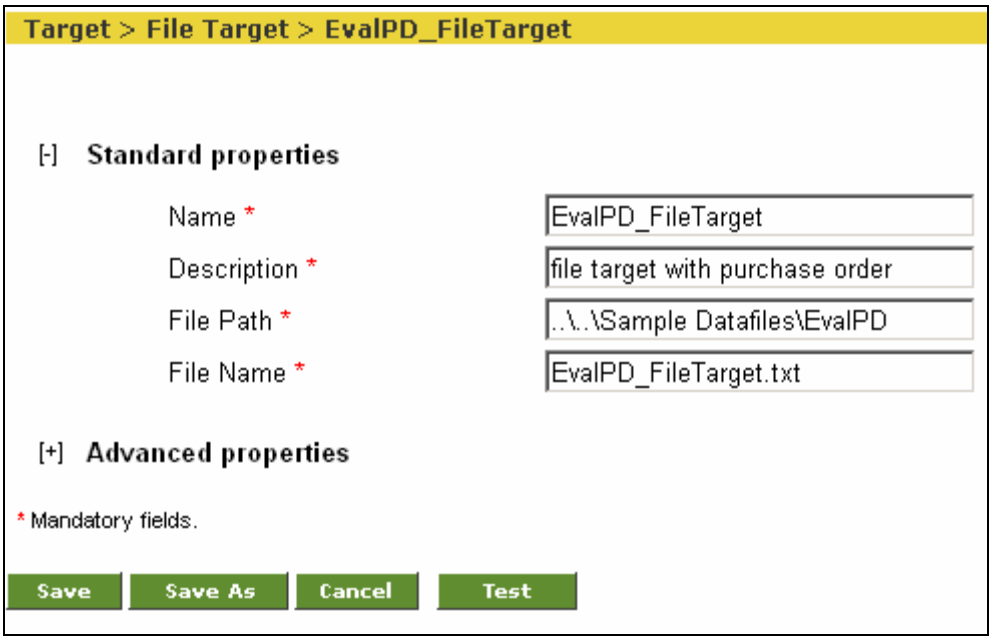



Figure 5.5: Edit *EvalPD_FileTarget* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 5.3: Details of Fields on Edit File Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Target |
| Description | Description of the File Target |
| File Path | Path of the target file.<br>For example:<br>**../../Solutions/Demo/EvalPD/** |
| File Name | Name of the target file. For example: **EvalPD_FileTarget.txt** |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file target activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the file target activity has been updated successfully.

You can verify the file target activity at design time. For this, click **Test Connection**. This verifies the values in the *File Path* and *Filename* fields and checks whether the file actually exists in the specified location.

## Editing Mail Target (EvalPD_MailTargetLessThan50K & EvalPD_MailTargetGreaterThan50K)

Mail Target provides the ability to specify target location that is accessible via Mail. In this Process Flow, mail target is used to send target data either to the manager or the director depending on whether the given condition is true or false. When the purchase amount is less than US$ 50000, EvalPD_MailTargetLessThan50K is executed and mail is sent to the Manager for approval. When the purchase amount is greater than US$ 50000, EvalPD_MailTargetGreaterThan50K is executed and mail is sent to the Director for approval.

***Steps to edit the Mail Target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.

2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.

3. Click **[+] Target** to expand the tree, and then click **Mail**. The Manage Mail Target screen is displayed with a list of existing Mail Target activities (see Figure 5.6).

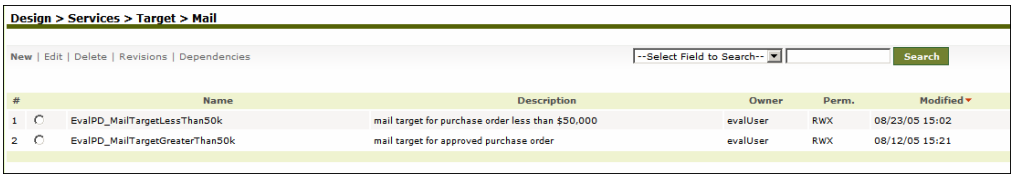



Figure 5.6: Manage Mail Target

4. Select the radio button adjacent to *EvalPD_MailTargetLessThan50K* activity and then click **Edit** link. This displays the Edit *EvalPD_MailTargetLessThan50K* activity screen, with the properties of the activity displayed in their respective fields (see Figure 5.7).

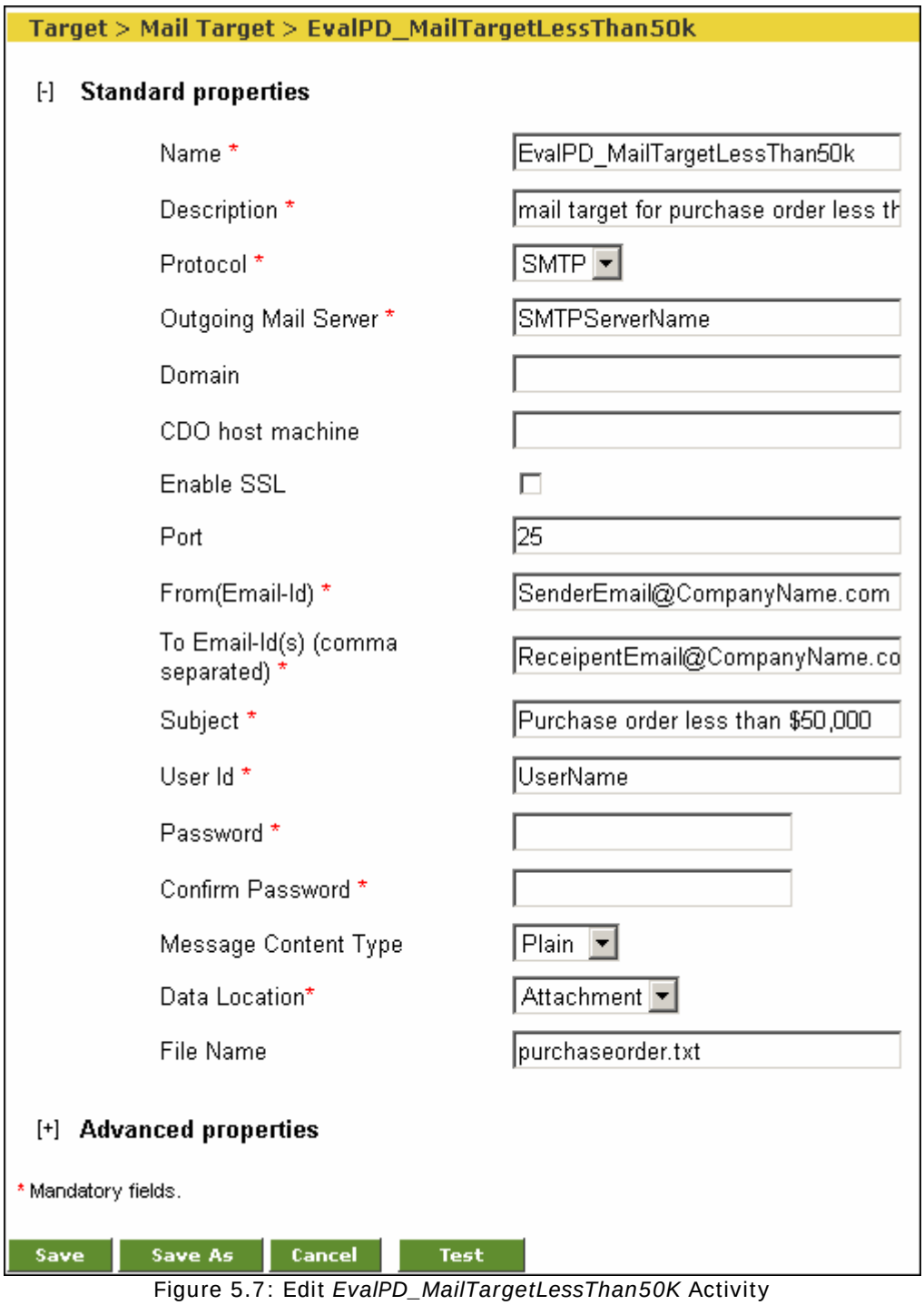



Figure 5.7: Edit *EvalPD_MailTargetLessThan50K* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 5.4: Details of Fields on Edit Mail Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Mail Target. |
| Description | Description of the Mail Target. |
| Outgoing Mail(SMTP) | Name or IP address of the Outgoing Mail or SMTP Server. |

| | |
|---|---|
| Server | |
| From(Email-Id) | Sender's email address |
| To Email-Id(s) (comma separated) | Email Id(s) of the recipient(s) separated by commas |
| Subject | Subject of target email |
| User ID | Username required to access the mailbox |
| Password | Password required to access the mailbox |
| Confirm Password | Re-enter the password for confirmation |
| Data Location | Data Location specifies whether the data is in the email body or is contained in an attached file. |
| File Name | Name of the file that is used as an attachment |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Mail Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the mail target activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the mail target activity has been updated successfully.

You can verify the mail target activity at design time. For this, click **Test Connection**.

Repeat the same steps to edit the *EvalPD_MailTargetGreaterThan50K* activity.

# CREATING PROCESS FLOW

## (EvalPD_ProcessFlow)

Process Flow is the set of activities arranged in a sequence to perform a specific task(s). Process Flow is created by combining various activities such as Source, Target, Schema or Transformer activities in a logical sequence. Process Designer is used to create a Process Flow. Process Designer has list of activities created. You only need to arrange them in a logical sequence and connect them with BPMN Flows.

***Steps to create EvalPD_ProcessFlow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.

2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Click the **New Process Flow** button. The Create Process Flow screen is displayed (refer to Figure 7.28).
4. Enter the name and the description of the new Process Flow in the *Name* and *Description* box respectively.
5. Select the logging level from the *Logging Level* drop-down list. There are four level of logging. They are described in Table 7.9.
6. Select repository file retention from the *Repository File Retention* option. During execution Process Flow creates a temporary repository file to store the intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand sometime these repository files can be helpful in case of failure of Process Flow execution. For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files and target formatted data. By default repository files are being stored in the repository folder of the Adeptia Suite. You can also choose an option to delete them or to archive them in a different location. There are four options for the Repository File Retention. They are described in Table 7.10.
7. Click the **Process Designer** button to open Process Designer. The Process Designer Screen is displayed (refer to Figure 7.29).
8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar.

9. Click **[+] Activities** in Repository Panel, to expand the list of services and then click **[+] Source**. All the items in the **Source** category are displayed.
10. Click the **[+] File Source**. A list of existing File Source activities that is displayed.
11. Select **EvalPD_FileSource** and drag it to the Graph Canvas Area (see Figure 5.8).

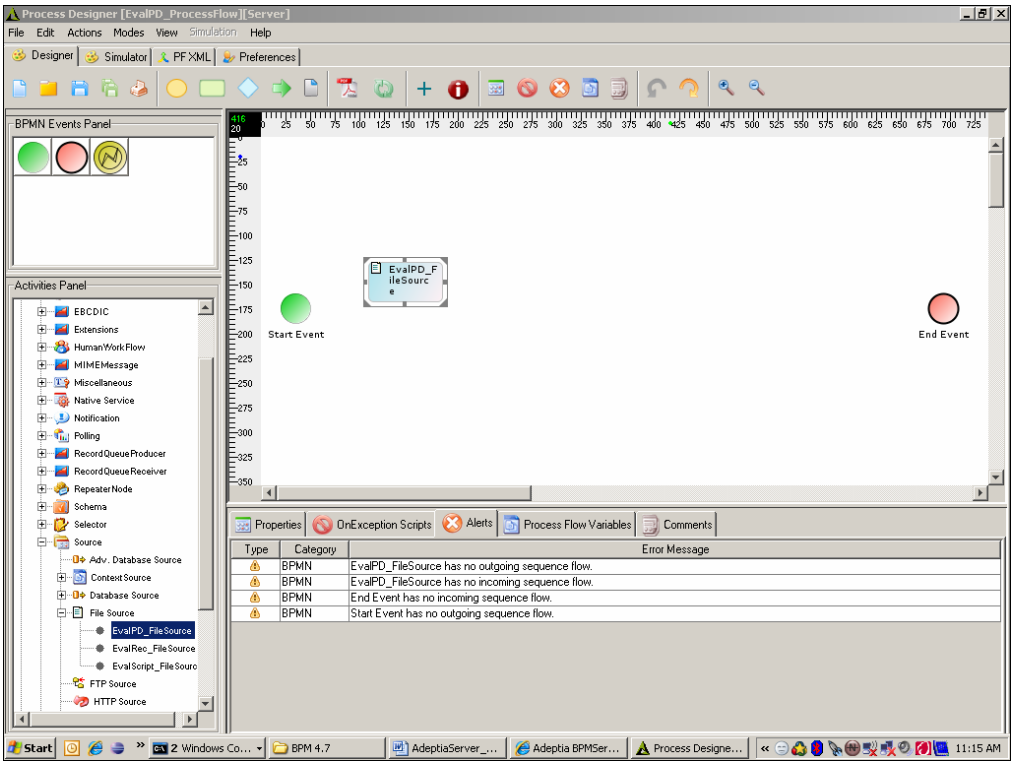



Figure 5.8: Drag File Source Activity to Graph Canvas Area

12. To create Process Flow Variable, click *Process Flow Variables* tab in the bottom pane. The Process Flow Variables panel is displayed in bottom pane. (see Figure 5.9)

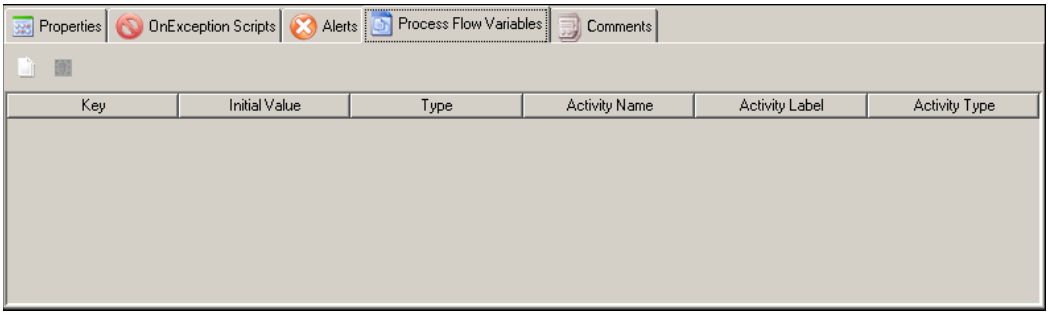



Figure 5.9: Process Flow Variables Panel

13. Click **New Process Flow Variable** ( ) button. The Process flow variable entry dialog box is displayed (see Figure 5.10).

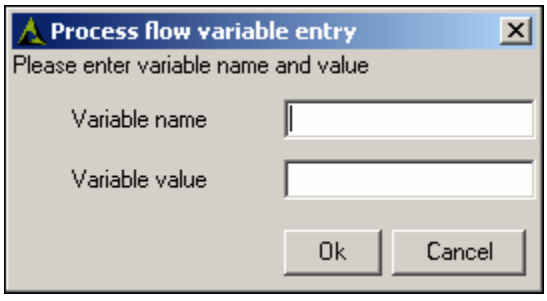


Figure 5.10: Create Process Flow Variable

14. Enter *Data* in the *Variable Name* field and click **Ok**. A process flow variable with name *Data* is created.

15. To create Context Target, click **[+] Target** under Activities list in Repository View, to expand the list of Target activities.

16. Click **[+] Context Target** activity to expand the Context Target activity. The Context Target node is displayed.

17. Drag the Context Target node to the Graph Canvas Area (see Figure 5.11).

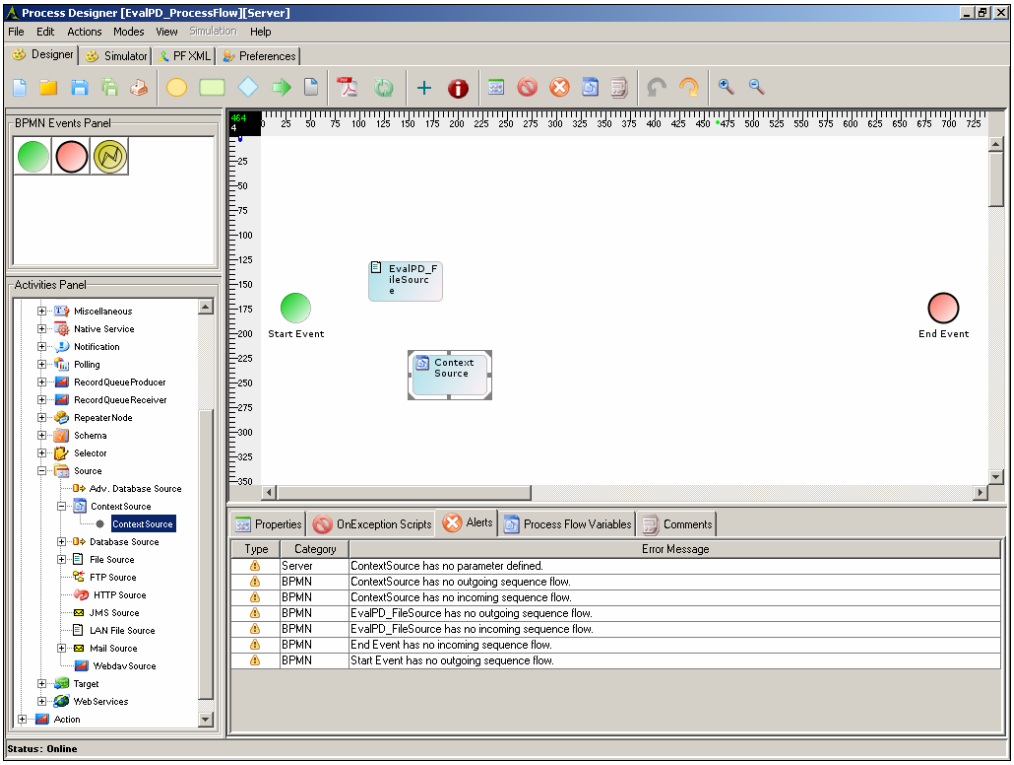



Figure 5.11: Drag Context Source Activity to Graph Canvas Area

18. Right-click the Context Target and select **View Properties**. Properties of the Context Target are displayed in the Bottom Pane (see Figure 5.12).

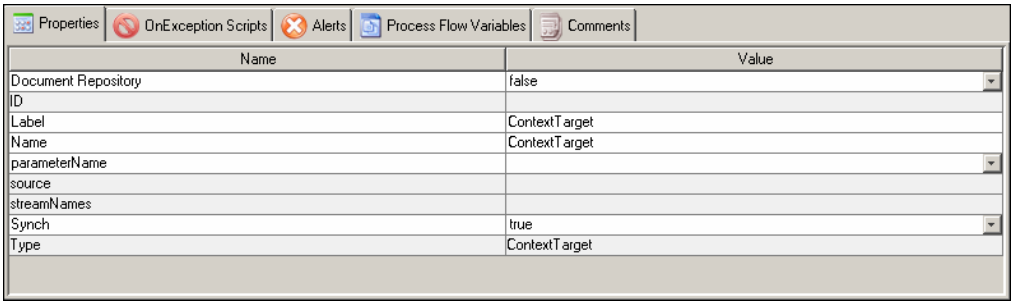



Figure 5.12: View Properties of Context Target Activity

19. Enter the name of the context Target as *Data* in the *Value* field of the *Name* property

20. Change the value of *Label* from *Context Target* to *EvalPD_ContextTarget.*

21. Select *Data* from the *parameterName* drop-down list (see Figure 5.13).

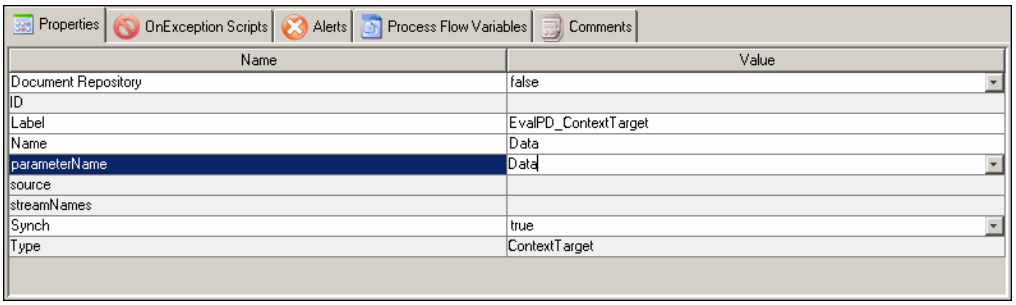



Figure 5.13: Enter Context Target Name

22. Click **[+] Source** and then **[+] Context Source**. Select **Context Source** and drag it to Graph Canvas Area.

23. Right-click the Context Source and select **View Properties**. Properties of the Context Source are displayed in the Bottom Pane (see Figure 5.14).

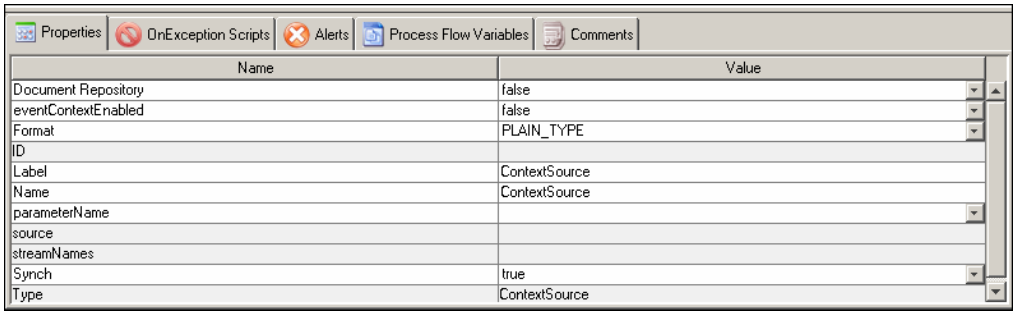



Figure 5.14: View Properties of Context Source activity

24. Enter the name of the context Target as *DataSource* in the *Value* field of the *Name* property

25. Change the value of *Label* from *Context Source* to *EvalPD_ContextSource.*

26. Select *Data* from the *parameterName* drop-down list (see Figure 5.15).

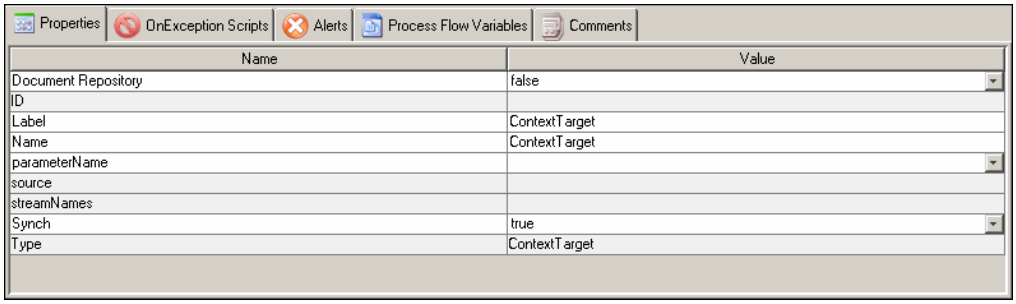



Figure 5.15: Enter Context Source Name

27. Click **[+] Repeater Node** to expand it. Select **Repeater Service** and drag it to the Graph Canvas Area.

28. Click **[+] Target** and then **[+] File Target**. Select **EvalPD_FileTarget** activity and drag it to the Graph Canvas Area.

29. To select a BPMN Gateway, click the **Gateway** (  ) icon in the Palette and drag it to the Graph Canvas Area (see Figure 5.16). In this Process Flow, **Gateway** is used to check the purchase amount and to decide which mail target is to be used.

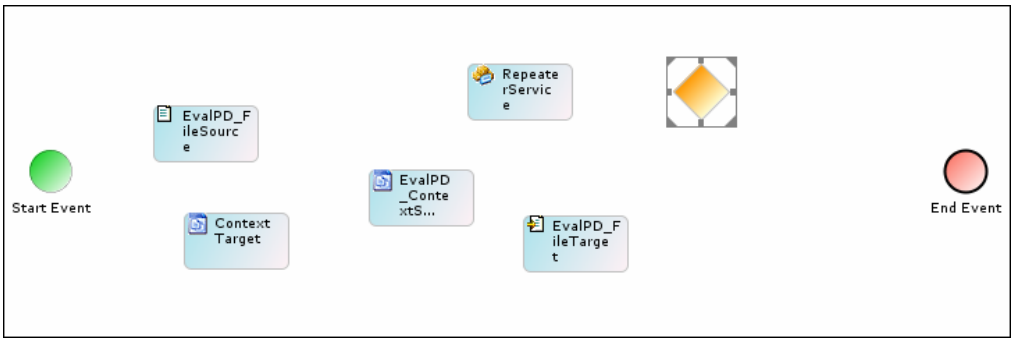



Figure 5.16: Drag BPMN Gateway Element to Graph Canvas Area

30. Click [+] **Action** to expand the list of actions.

An Action name is always unique.

31. Select **Trace** and drag it to the Graph Canvas Area. *Trace* is used to write any message, which is later, logged into the Process Flow Log during the execution of Process Flow (see Figure 5.17).

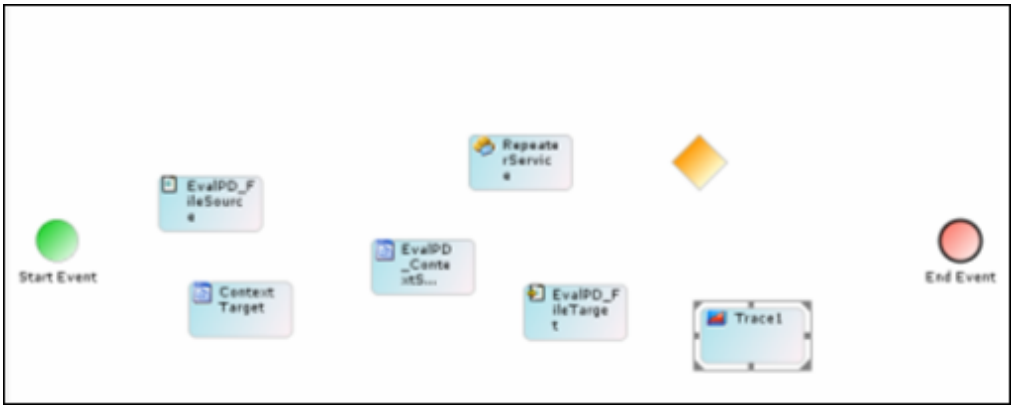



Figure 5.17: Drag Trace Action

32. To enter the message, right-click the **Trace** activity and select **View Properties**. The properties of the trace activity are shown in the Properties Panel of the Bottom Pane (see Figure 5.18).

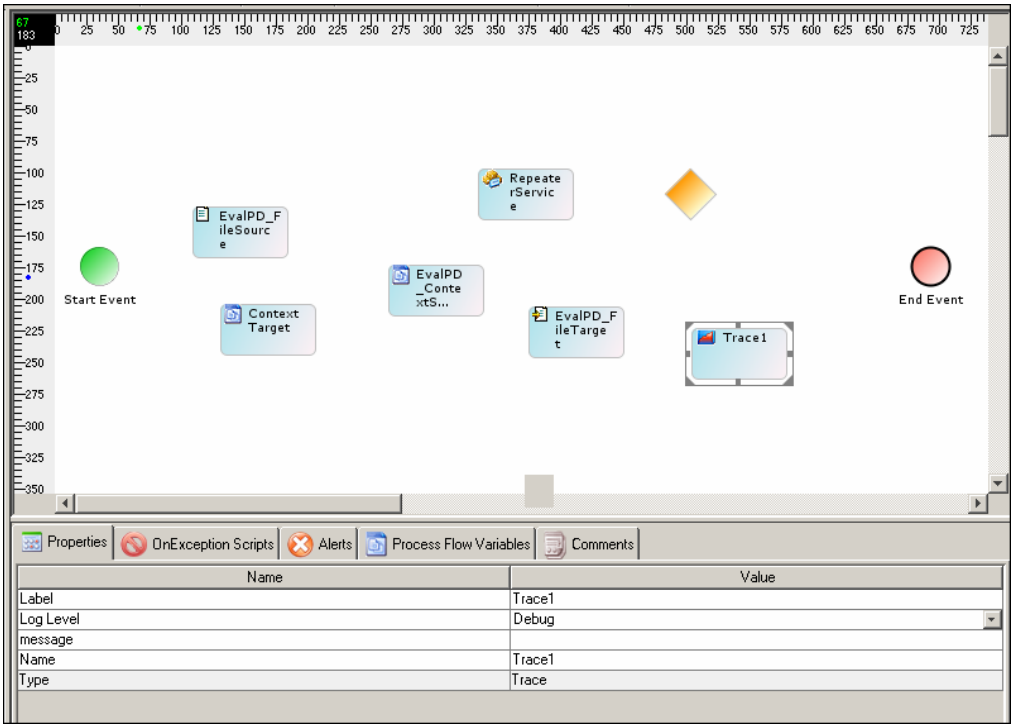



Figure 5.18: View Properties of Trace Action

33. In the value field of the *Message* property enter the following message:

    *Mailing Purchase Order (PO# = $$PO#$$ and Amount = $ $$PO_Amount$$) for approval to Manager*

34. Change the Name and Label of the *Trace* activity to *EvalPD_TraceForPOLessThan50K*.

35. Similarly, again drag **Trace** from the list of actions, change its Name and Label to *EvalPD_TraceForPOGreaterThan50K* and enter the following message in the *Value* field of the Message property (see Figure 5.19).

    *Mailing Purchase Order (PO# = $$PO#$$ and Amount = $ $$PO_Amount$$) for approval to Director*

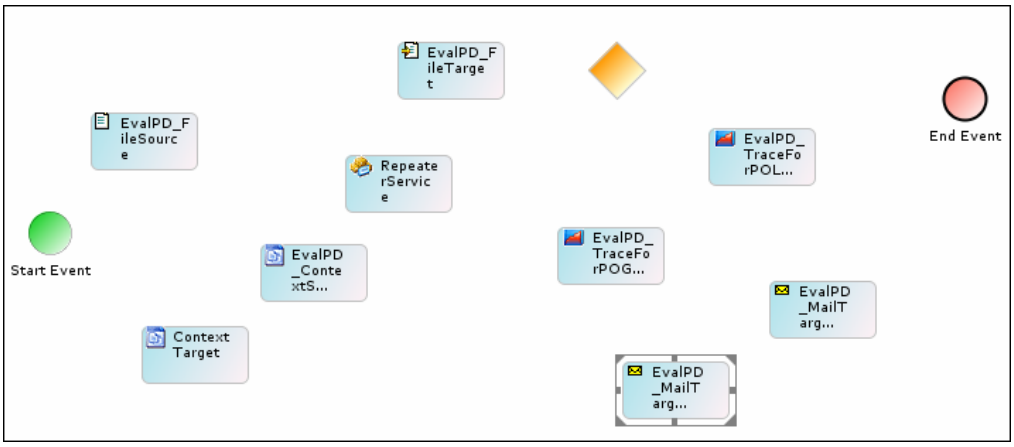



Figure 5.19: Drag another Trace Action

36. Click **[+] Target** and then **[+] Mail Target.** Select **EvalPD_MailTargetLessThan50K** activity and drag it to the Graph Canvas Area.
37. Select **EvalPD_MailTargetGreaterThan50K** activity and drag it to the Graph Canvas Area.
38. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
39. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
40. To connect *Start Event* with *EvalPD_FileSource,* drag mouse pointer from *Start Event* to *EvalPD_FileSource* (see Figure 5.20).

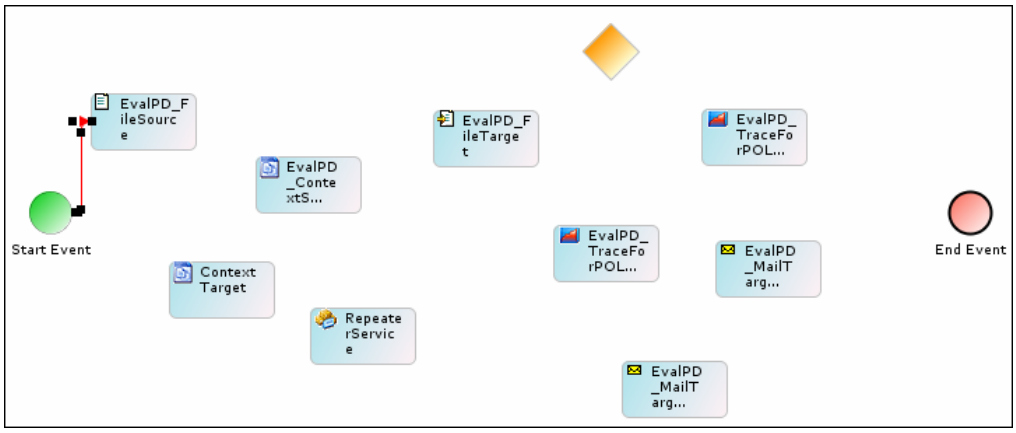



Figure 5.20: Connect Start Event to File Source Activity

41. Similarly, connect all other activities as shown in Figure 5.21.

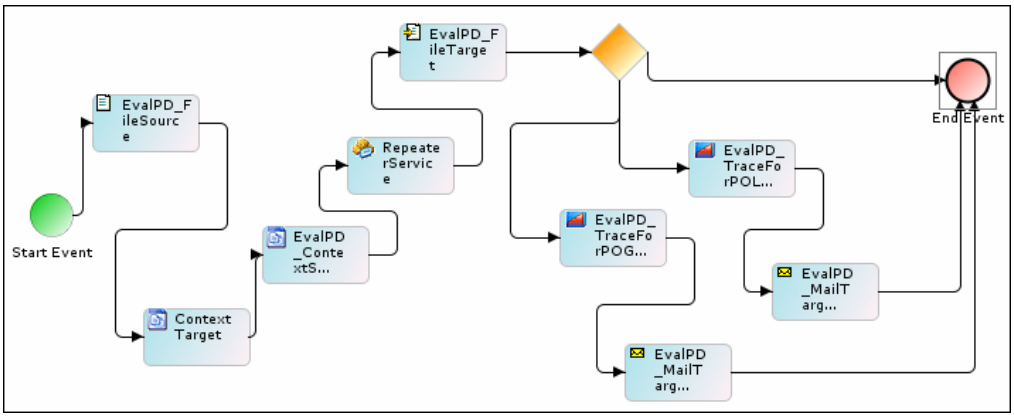



Figure 5.21: Connect all Activities

| | |
|---|---|
|  | Figure 5.21 shows only the control flow of the Process Flow. Control Flow only shows the order in which activities of a Process Flow are executed. It does not show the flow of data. For example as you can see in the Figure 5.21, control flow from Repeater Service goes to EvalPD_FileTarget and further to the Decision Node. But data is not passed from EvalPD_FileTarget to the Decision Node. Data is directly passed from Repeater Service EvalPD_FileTarget and both of the mail target activities (EvalPD_MailTargetLessThan50k and EvalPD_MailTargetGreaterThan50k). To create the data flow, you need to create Multiple Stream from repeater service to EvalPD_FileTarget and Mail Target activities. Here you need to create two streams from the Repeater Service. One Stream goes to EvalPD_FileTarget and the other stream goes to both of the Mail Target activities. |

42. To create data stream, right-click **Repeater Service** and select **Multiple Stream**. The Multiple Stream dialog box is displayed (refer to Figure 4.22).
43. Enter number of streams (2) in the *Enter stream count* field, and then click the **Add Stream** button.
44. Select *1* from the *Streams* drop-down list, and select **EvalPD_FileTarget**. from the *Activities* drop-down list,
45. Click the **Map** button. A stream between *Repeater Service* and *EvalPD_FileTarget* is created (see Figure 5.22).

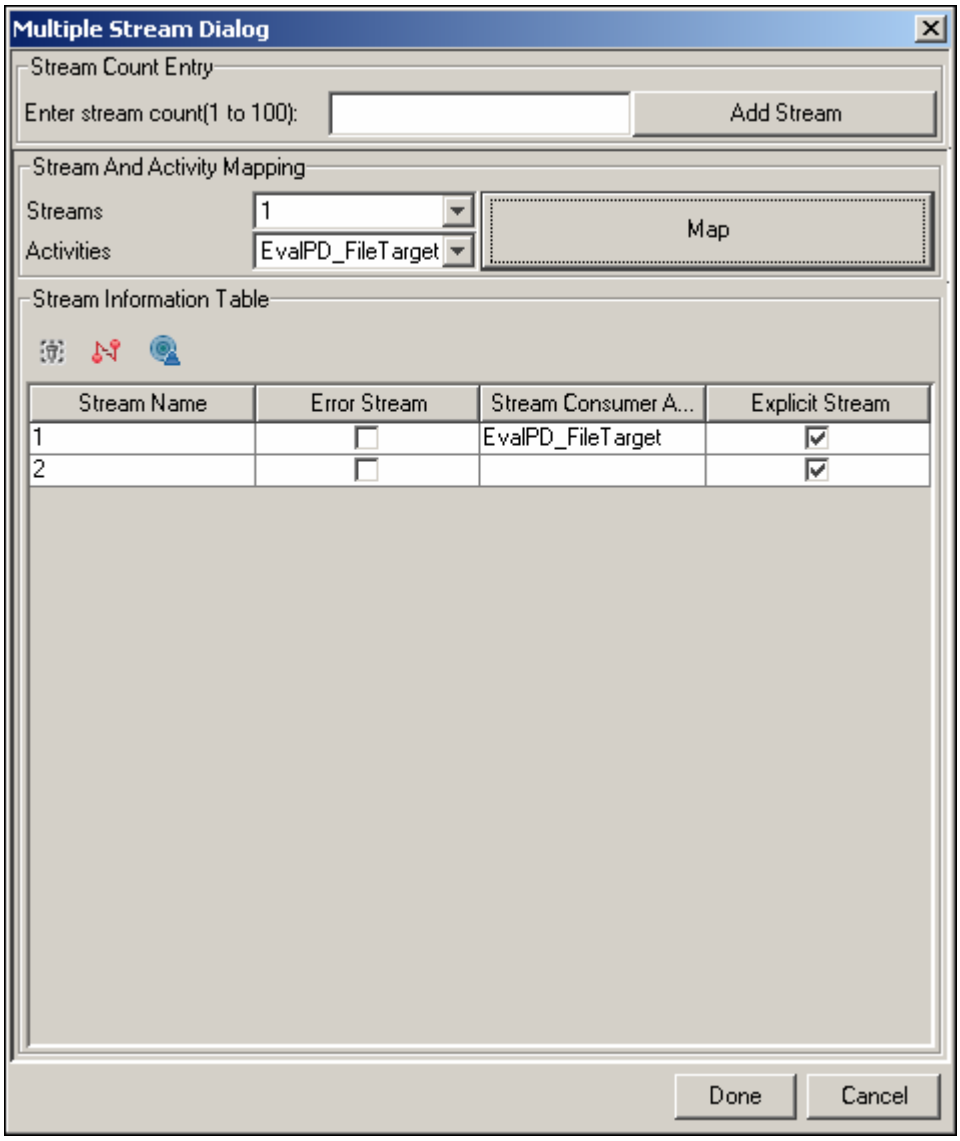



Figure 5.22: Create Stream

46. Ensure that the *Explicit Stream* checkbox is checked.

47. Select *2* from the *Streams* drop-down list, and select **EvalPD_MailTargetLessThan50K** from the *Activities* drop-down list, and click the **Map** button.

48. Select **EvalPD_MailTargetGreaterThan50K** from *Activities* drop-down list, and click the **Map** button. A dialog box is displayed (see Figure 5.23).

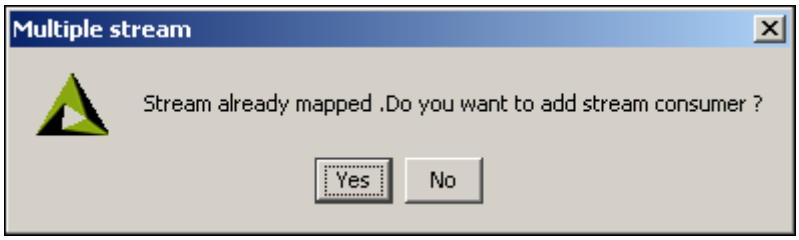



Figure 5.23: Add another Stream

49. Click the **Yes** button to add second mail target activity to the same stream.

50. Click the **Done** button to close the Multiple Stream Dialog box. Data stream created is shown in the Graph Canvas area (see Figure 5.24).

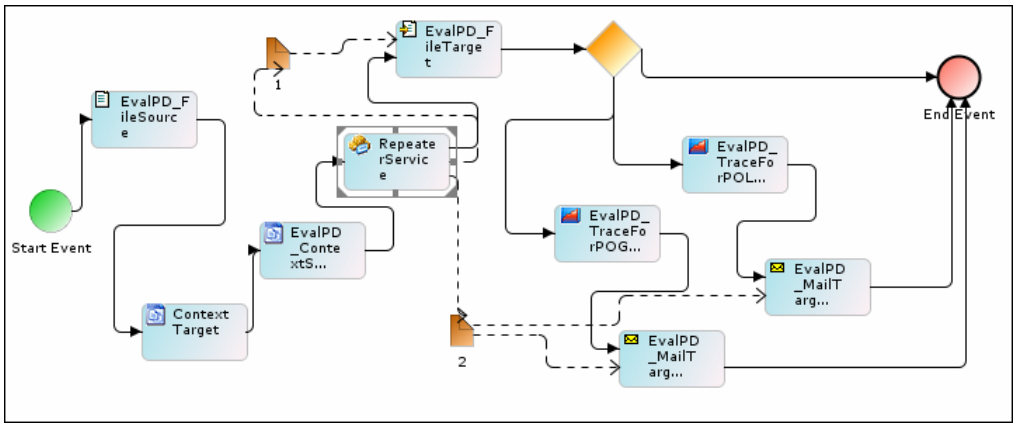



Figure 5.24: Multiple Streams Created

51. To define the decision criteria, right-click the control flow which is connecting Decision Node with *EvalPD_TraceForPOLessThan50K* and select **View Properties**. Properties of the selected control flow are shown in the Properties Panel in the Bottom Pane.

52. Click the **Edit Condition** in the Properties Panel. The Condition Wizard is displayed (refer to Figure 8.30).

53. Select **Java Condition** and click the **Next** button. The Condition Type screen is displayed (refer to Figure 8.31).

54. Enter the sample Java Code to specify the decision criteria and click the **Done** button (see Figure 5.25).

```
// retrieving purchase order from context
String purchaseOrderData = (String) context.get("Data");
// filling purchase order attributes
String [] dataArray = purchaseOrderData.split(",");
int amount = 0;
for( int i = 0 ; i < dataArray.length ; i++ )
{
//extracting  purchase order PO_AMOUNT attribute
        if( dataArray[i].indexOf("PO_AMOUNT")!= -1)
        {
        String [] attributeArray = dataArray[i].split("=");
        context.put("PO_Amount", attributeArray[1] );
```

```
        amount = Integer.parseInt(attributeArray[1]);
        }else   if( dataArray[i].indexOf("PO#")!= -1)
        {
//extracting  purchase order PO# attribute
        String [] attributeArray = dataArray[i].split("=");
        context.put("PO#", attributeArray[1] );
        }
}
// checking condition
if(amount < 50000){
        return true ;
}else{
        return false;
}
```

Figure 5.25: Sample JAVA Code

55. Similarly, define the following decision criteria on the Control Flow connecting decision node and *EvalPD_TraceForPOGreaterThan50K*. (see Figure 5.26).

```
// retrieving purchase order from context
String purchaseOrderData = (String) context.get("Data");
//splitting order to fill attributes
String [] dataArray = purchaseOrderData.split(",");
int amount = 0;
for( int i = 0 ; i < dataArray.length ; i++ )
{
        //filling amount from PO_AMOUNT attribute
        if( dataArray[i].indexOf("PO_AMOUNT")!= -1){
        String [] attributeArray = dataArray[i].split("=");
        amount = Integer.parseInt(attributeArray[1]);
        }
}
if(amount >= 50000)
return true ;
```

Figure 5.26: Define Decision Criteria

56. To define the Sequence Flow Ordering, right-click the **decision** node and select the **Sequence Flow Ordering** option. The Sequence Flow Ordering dialog box appears (refer to Figure 8.33).

57. Ensure that the control flow connecting decision node and the **EvalPD_TraceForPOLessThan50K** is listed first.

| | |
|---|---|
|  | A default outgoing sequence flow is added for the gateway. While executing a process flow, if none of the specified conditions are met, then the default gateway is executed. If there is no default gateway specified, then an error occurs at the Process Flow design level. |

58. Save the Process Flow by clicking the **File** menu and selecting **Save Process Flow to Server**. A dialog box is displayed confirming that the *EvalPD_ProcessFlow* has been saved successfully. If the *comments* property is enabled, then clicking **Save Process Flow to Server** will display a screen where you need to enter comments related to creating the process flow (refer to Figure 7.37).

59. Enter the comments in the *Specify Comments for process flow customer* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

60. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.
61. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

# 6 PROCESS FLOW TO PROCESS EXCEL DATA

This section describes the sample Process Flow, which is used to process contents of an Excel file attached with an email.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This sample Process Flow demonstrates the use of different features of process designer and complex mapping functions. The Process Flow is triggered on arrival of emails with different subject. Each mail has an excel file attached to it. Subject of the mail specifies the format of the excel file. Based on subject of the mail, Decision Node of the Process Flow decides which schema to be used to parse data of the Excel file. For example, if subject of email is FORMAT1, Schema1 is used. If subject of email is FORMAT2, schema2 is used. After the data is parsed using either of the schemas, mapping rules are applied and data is inserted or updated into one of the two databases. All error records (e.g. duplicate record) are written into a sequential file.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- Mail Event to trigger the Process Flow on arrival of a mail
- Mail Source to receive data from mail event and forward it to Excel Schema
- Excel Schema to convert data of excel file into intermediate XML format
- Mapping to map data fields of Excel schema and Database Schema
- Database Schema to convert data from intermediate XML format to database specific format
- Database Target to insert or update data into the database server
- Process Designer to create Process Flow. Following Process Designer features are used in this Process Flow:
- Decision Node to decide which excel schema to use based on incoming Excel file format
- Multiple Streams to send data from one source to more than one schema
- Put context Var to append date and time stamp with the name of the file at the target end

# DESCRIPTION

This sample Process Flow can be outlined as below:

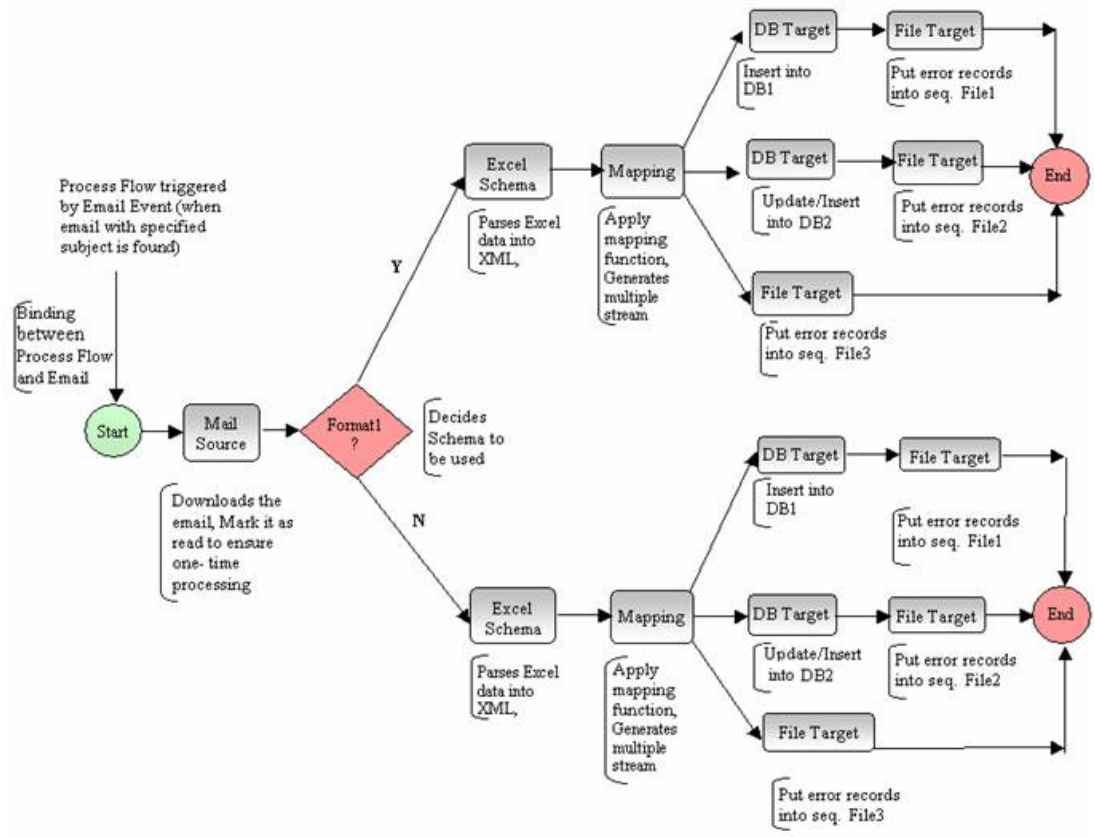



Figure 6.1: Flow Chart of the Process Flow

## Mail Event (EvalPF_MailEvent_Format1 and EvalPF_MailEvent_Format2)

Mail Event is used to trigger Process Flow when a message from specified sender or with specified subject arrives in a mailbox. In this Process Flow two Mail Events are used. The Mail Event **EvalPF_MailEvent_Format1** is used to trigger the Process Flow when a message with subject **Format1** arrives in the mailbox. The Mail Event **EvalPF_MailEvent_Format2** is used to trigger the Process Flow when a message with subject **Format2** arrives in the mailbox.

## Event Registry (EvalPF_EventRegistry_Format1 and EvalPF_EventRegistry_Format2)

Event Registry is used to register the Mail Events with Process Flow. In other words Event Registry is a link between Mail Event and the Process Flow. EvalPF_EventRegistry_Format1 is used to register EvalPF_MailEvent_Format1 with

the Process Flow. EvalPF_EventRegistry_Format2 is used to register EvalPF_MailEvent_Format2 with the Process Flow.

## Process Flow (EvalPF_ProcessFlow)

Process Flow receives data from one of the Mail events, converts it into database format and then inserts the data in one of the two databases. Process Flow does this conversion using following sequence of activities:

1. **Mail Source** (EvalPF_MailSource)

   Mail source is used to receive data from Mail Events and forward it to one of the two Excel schemas based on subject of the mail.

2. **Excel Schema** (EvalPF_ExcelSchema_Format1 and EvalPF_ExcelSchema_Format2)

   Excel Schema is used to parse the data from the excel file attached with mail received from Mail Source. Excel Schema converts the parsed data into intermediate XML format. If subject of mail is Format1, EvalPF_ExcelSchema_Format1 is used to parse the data. If subject of mail is Format2, EvalPF_ExcelSchema_Format2 is used to parse the data.

3. **Mapping** (EvalPF_MappingTranformation_Format1 and EvalPF_MappingTransformation_Format2)

   Mapping is used to map data fields of Excel schema with data fields of Database Schema. Similar to Excel Schema, two different mapping activities are used based on subject of the mail.

4. **Database Schema** (EvalPF_DatabaseSchema_Database1 and EvalPF_DatabaseSchema_Database2)

   At the target end, database Schema is used to convert data from intermediate XML format into database specific format. Database Schema uses Database Driver (EvalPF_DatabaseDriver_SQLServer) and Database Info (EvalPF_DatabaseInfo_SQLServer) to connect to Database Server.

5. **Database Target** (EvalPF_DatabaseTarget_Database1 and EvalPF_DatabaseTarget_Database2)

   Database target is used to specify the database server and name of the database, where the target data is stored. In this Process Flow, SQL Server is used as the database server. Database Target uses Database Schema. EvalPF_DatabaseTarget_Database1 uses EvalPF_DatabaseSchema_Database1 and EvalPF_DatabaseTarget_Database2 uses EvalPF_DatabaseSchema_Database2.

6. **File Target** (EvalPF_FileTarget_ErrorRecord, EvalPF_FileTarget_InsertError and EvalPF_UpdateError)

In this sample process file targets are used to store error records. Sometime it might be possible that format of the excel file received is not compatible with either of the schemas. In this case records of that excel file is saved into the file specified in EvalPF_FileTarget_ErrorRecord). There could be other possibilities of error while inserting or updating the records in database target. Records which cannot be inserted into database server is stored into the file specified in EvalPF_FileTarget_InsertError. Records, which cannot be updated in database server, is stored in the file specified in EvalPF_FileTarget_UpdateError.

## USAGE SCENARIO

This sample Process Flow can be used whenever you wish to process incoming data of different format using different schemas.

## DATA DESCRIPTION

Data used in this Process Flow is the sales report, which is in, excel files. There are two formats of excel files. The formats of the Excel files, which contain source data, are displayed in Table 6.1and Table 6.2 respectively.

Table 6.1: Format 1 of Excel Files used as Source

| Field Name | Description | Data Type |
|---|---|---|
| Account Number | Account Number can be either 99-999999 or AA-999999 formats. Here 99 denotes any numeric and AA denote any alphabets. | String |
| Product Code | Product Code | String |
| Quantity | Quantity of the product specified Product Code Field | Number |
| Unit Price | Unit Price in US $ | Number |

Table 6.2: Format 2 of Excel Files used as Source

| Field Name | Description | Data Type |
|---|---|---|
| Account Number | Account Number can be either 99-999999 or AA-999999 formats. Here 99 denotes any numeric and AA denote any alphabets. | String |
| Total Price | Multiplication of Quantity and Unit Price | Number |
| Product Code | Product Code | String |

The fields of both the databases used as target are displayed in the table below.

Table 6.3: Fields of Databases used as Target

| Field Name | Description | Data Type |
|---|---|---|
| Account Number | Account Number can be either 99-999999 or AA-999999 formats. Here 99 denotes any numeric and AA denote any alphabets. Records with Account Number 99-999999 goes to Database1 and records with Account Number AA-999999 goes to Database2 | String |
| Total Price | Multiplication of Quantity and Unit Price | Number |
| Product Code | Product Code | String |

## PREREQUISITES

- SQL Server, which is used as target, must be running.
- Database table must be created as described in the Data Description section. To create the table in target database, you can run the databasescript.sql file, located in **../../Solutions/Demo/EvalPF** folder.
- Access to Incoming Mail (POP3) Server specified in Mail Event and Mail Source activities.
- Some activities must be edited before executing the Process Flow. These activities are outlined as:
- EvalPF_MailEvent_Format1
- EvalPF_MailEvent_Format2
- EvalPF_MailSource
- EvalPF_DataBaseInfo_SQLServer

To know, how to edit these activities refer to **Editing Activities** section.

## USING ANOTHER DATABASE SERVER

This sample Process Flow is configured with SQL Server as target. If another database server is to be used as target, following activities need to be changed:

- EvalPF_DatabaseDriver_SQLServer
- EvalPF_DatabaseInfo_SQLServer
- EvalPF_DatabaseTarget_Database1
- EvalPF_DatabaseTarget_Database2

To know, how to edit these activities refer to **Editing Activities** section.

## EXECUTION AND MONITORING

This section describes the execution of Sample Process Flow and monitoring its execution. Steps involved to execute this sample Process Flow can be broadly divided as below:

- Activating Mail Events
- Sending mail with subject and attachments specified in Mail Events
- Monitoring Process Flow execution

## Activating Mail Events

By default, Mail Events are in *deactivated* state. They must be activated before executing the sample Process Flow.

***Steps to activate the Mail Events:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Mail**. The Manage Mail Event screen is displayed with the list of existing events (see Figure 6.2).

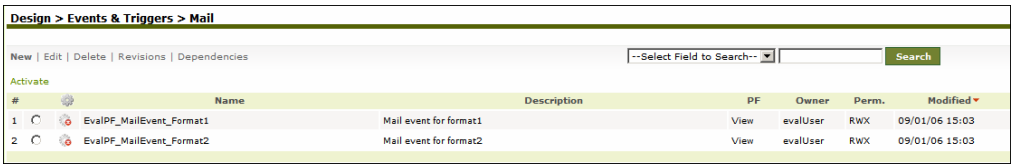



Figure 6.2: Manage Mail Event

3. To activate the Mail Event (*EvalPF_MailEvent_Format1*), Select the radio button adjacent to *EvalPF_MailEvent_Format1* activity and then click **Activate** link. A screen is displayed confirming that the Mail Event activity has been activated successfully.
4. Similarly, activate another Mail Event (*EvalPF_MailEvent_Format2*).

## Sending Mail to execute Process Flow

After Mail Events are activated they start checking specified mailbox for mails. Now to execute the Sample Process Flow, you only need to send the mail with specified subject and attachment to the mailbox. When mail is sent to the mailbox, one of the two Mail Events gets fired and in turn triggers the Process Flow.

## Monitoring Process Flow Execution

During execution of the Process Flow you can view status of the Process Flow as well as status of each activity of the Process Flow.

***Steps to monitor Process Flow execution:***

1. In the Adeptia Suite homepage menu, click **[+] History** to expand the tree. All the items in the **History** category are displayed.
2. Click **Process Flow Log**. The Process Flow Log screen is displayed (refer to Figure 7.4).
3. Select the Process Flow (*EvalPF_ProcessFlow*) from the *Select Process Flow* drop-down list.

4. Click the **Details** button. This displays the list of activities of the selected process flow conforming to the selected criteria (see Figure 6.3).

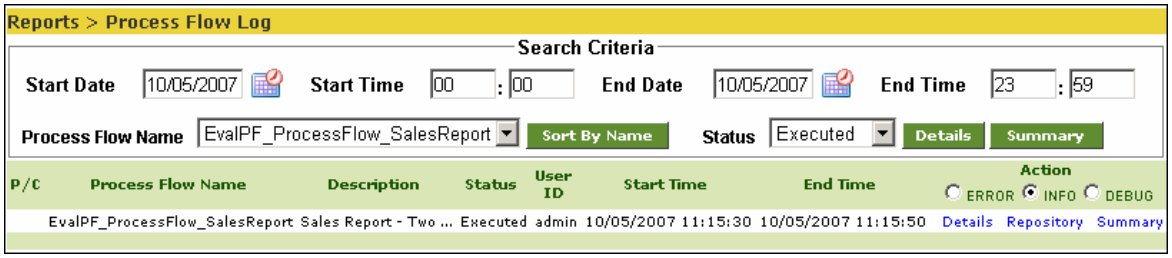
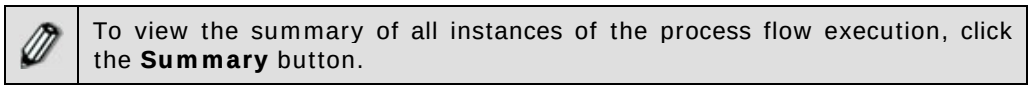



Figure 6.3: Searched Process Flows



To view the summary of all instances of the process flow execution, click the **Summary** button.

5. Click **Details** in the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 6.4).

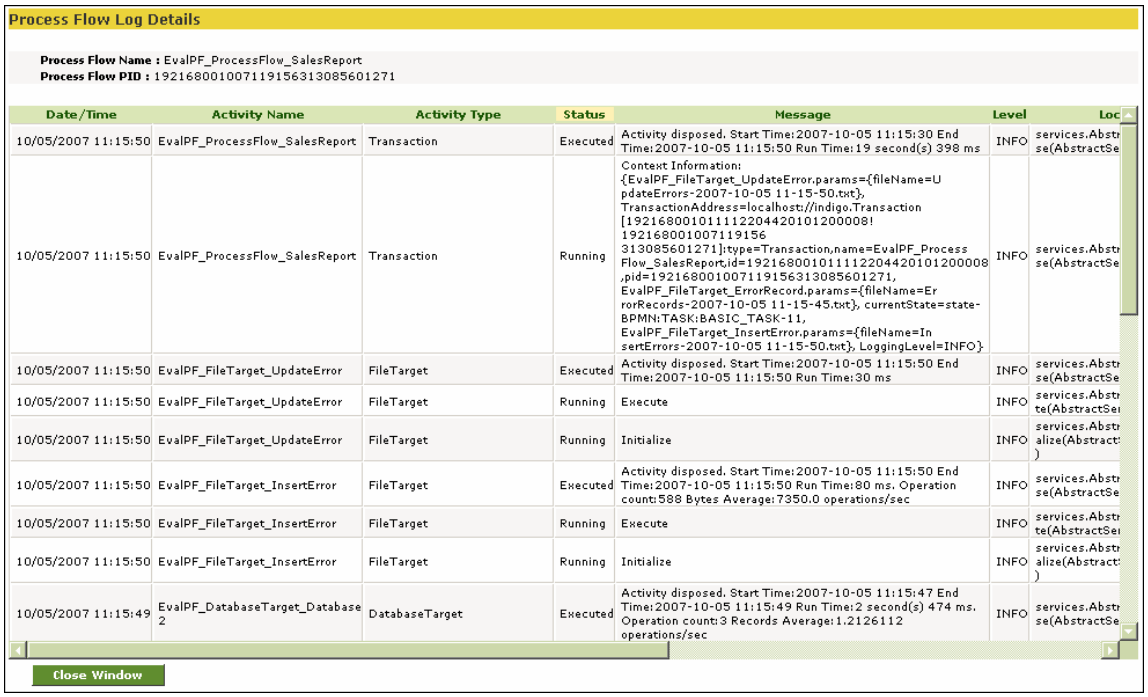



Figure 6.4: View Process Flow Log Details

# EDITING ACTIVITIES

Activities used in this sample Process Flow are pre-created. This section describes how to edit these activities.

## Editing Mail Events

Mail Event is used to trigger the Process Flow, when a mail with specified subject arrives in a mailbox. In this sample Process Flow two Mail Events are used. When a mail with subject **Format1** arrives in the mailbox, *EvalPF_MailEvent_Format1* gets fired and triggers the Process Flow. When a mail with subject **Format2** arrives in the mailbox, *EvalPF_MailEvent_Format2* gets fired and triggers the same Process Flow.

***Steps to edit Mail Events activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Mail**. The Manage Mail Event screen is displayed with the list of existing Mail Events (refer to Figure 6.2).
3. Select the radio button adjacent to Edit *EvalPF_MailEvent_Format1* activity and then click **Edit** link. This displays the Edit *EvalPF_MailEvent_Format1* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.5).

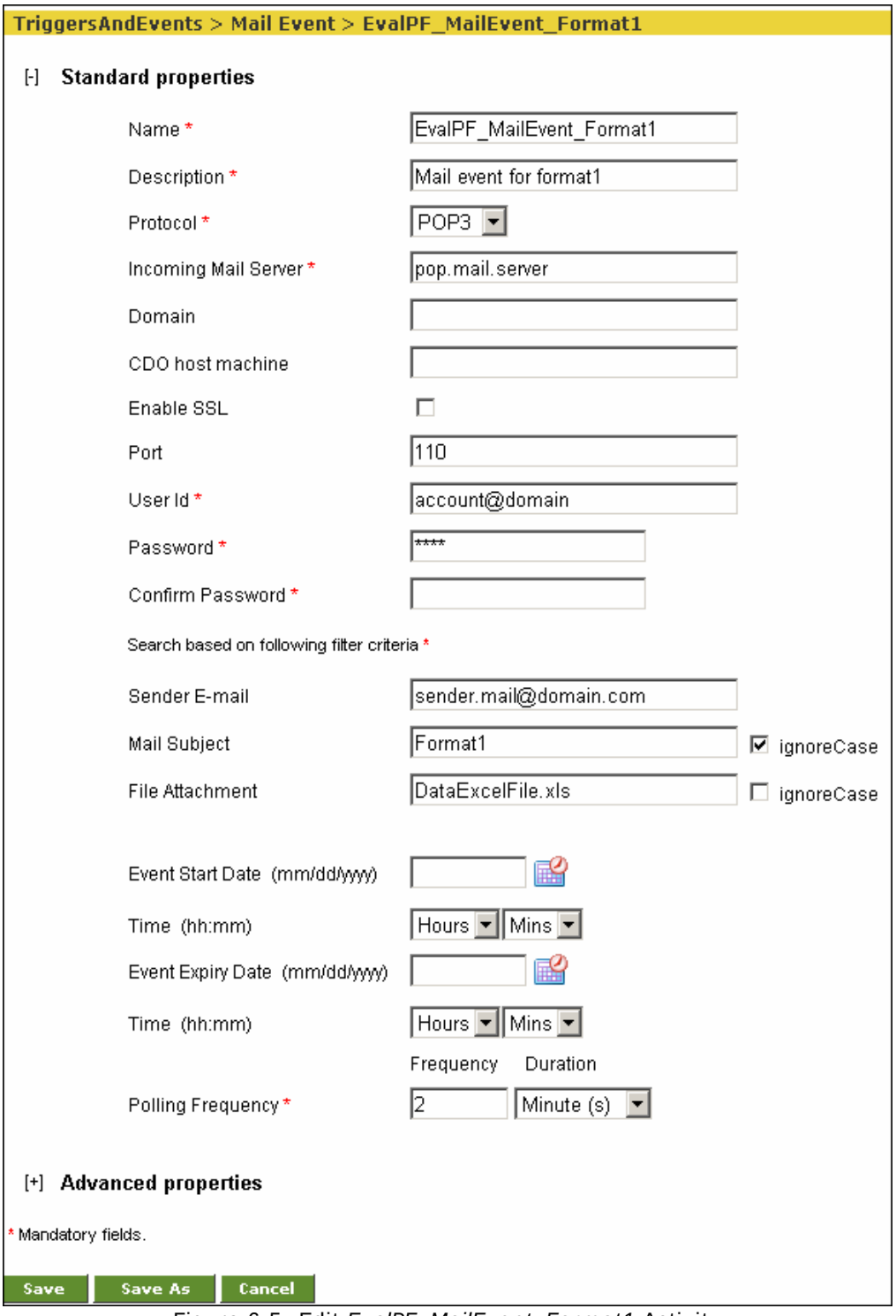



Figure 6.5: Edit *EvalPF_MailEvent_Format1* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.4: Details of Fields on Edit Mail Event Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Mail Event activity |
| Description | Description of the Mail Event activity |
| Protocol | Name of the internet protocol used for retrieving incoming mails. It can be POP3, IMAP4 or MAPI. |
| Incoming Mail Server | Name or IP address of the selected incoming mail server, where Mail Source checks for specified mail. Incase MAPI protocol is selected in the *Protocol* drop-down lists, enter the name of exchange server in the *Incoming Mail Server* field. |
| Domain | Enter the domain name. This field is applicable only when MAPI is selected in the protocol drop-down list. |
| CDO Host Machine | Enter the name of the CDO host machine. CDOConfig.exe is a tool, which comes with the J-Integra for Exchange SDK and is used for configuring CDO. Host where CDO is configured is called CDO host machine. |
| Enable SSL | Check this checkbox, if the specified mail server is a secure mail server. |
| Port | Default Port number of the Incoming Mail Server. You can change this if required |
| User ID | Username required to access the mailbox in which mail with specified subject arrives |
| Password | Password required to access the mailbox |
| Confirm Password | Re-enter the Password |
| Search based on following filter criteria | Select any of the following filter criteria:<br>Sender' Email Address — Email address of the sender<br>Subject — Subject of mail<br>File Attachment — Name of the attached file<br>If you want to ignore the case of subject or the name of file attachment, check the *Ignore Case* check box displayed next to *subject* or *File attachment field.*<br>In this Process Flow Sender's Email Address and Subject is used to specify the mail. |
| Event Start Date (mm/dd/yyyy) | Date from which Mail Events starts checking the specified mailbox. |
| Time (hh:mm) | Time from which Mail Events starts checking the specified mailbox. |
| Event Expiry Date (mm/dd/yyyy) | Date on which Mail event will stop checking for specified mail, |
| Time (hh:mm) | Time on which Mail Events will stop checking the specified mailbox. |
| Polling Frequency | Time interval at which Mail Event checks for specified mail |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Mail Event activity has been

updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the mail event activity (refer to Figure 7.10).

6. Enter the comments in the *Add Comments* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

7. Click **OK** to save the comments. This displays a screen confirming that the mail event activity has been updated successfully.

| | |
|---|---|
| | Repeat the same steps to edit the *EvalPF_MailEvent_Format2* activity. |

## Editing Mail Source (EvalPF_MailSource)

Mail Source is used to receive mails from the specified mail server. In this Process Flow mail source receives mails from Mail Event. Mail, which is used to trigger the Process Flow, is also used as the source of this Process Flow. Mail Event forwards the mails to Mail Source.

***Steps to edit the Mail Source activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **Mail**. The Manage Mail Source screen is displayed with the list of existing Mail Source activities (see Figure 6.6).

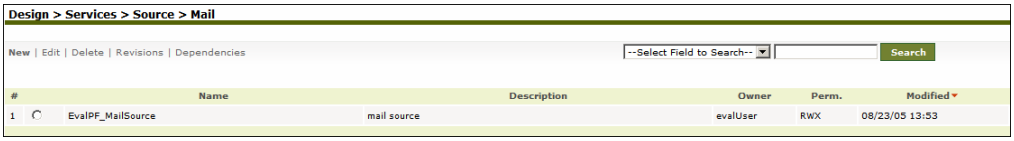



Figure 6.6: Manage Mail Source

4. Select the radio button adjacent to Edit *EvalPF_MailSource* activity and then click **Edit** link. This displays the Edit *EvalPF_MailSource* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.7).

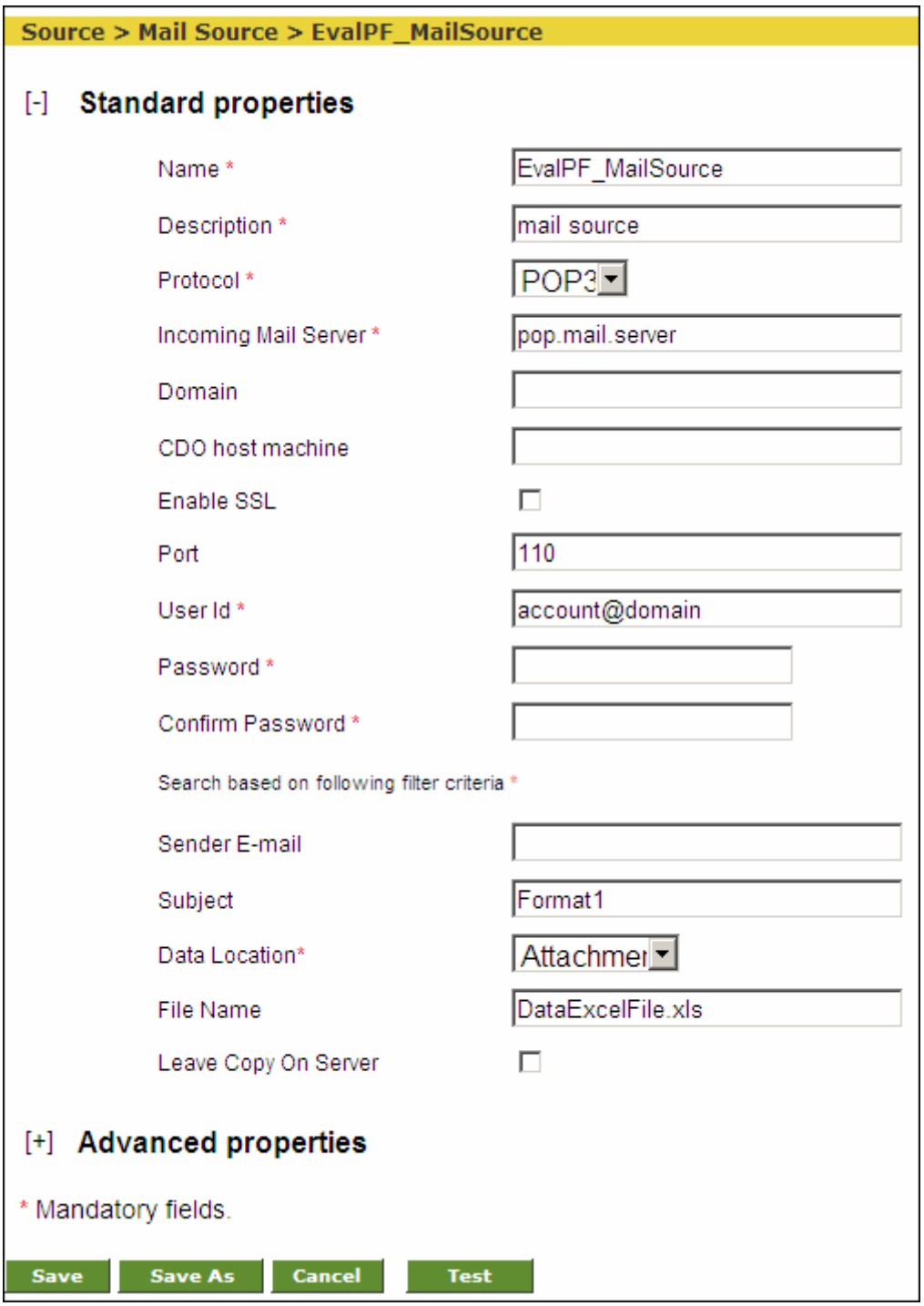



Figure 6.7: Edit *EvalPF_MailSource* Activity

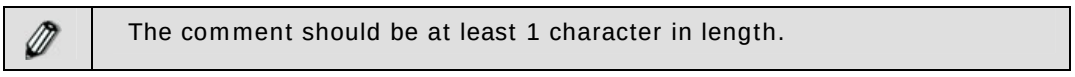
A detailed description of fields on this screen is explicated in the table below.

Table 6.5: Details of Fields on Edit Mail Source Screen

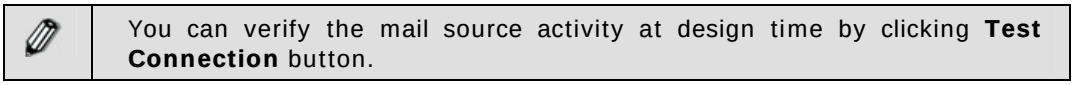
| Field Name | Field Description |
|---|---|
| Name | Name of the Mail Source activity |

| Description | Description of the Mail Source activity |
|---|---|
| Protocol | Name of the internet protocol used for retrieving incoming mails. It can be POP3, IMAP4 or MAPI. |
| Incoming Mail Server | Name or IP address of the selected incoming mail server, where Mail Source checks for specified mail. Incase MAPI protocol is selected in the *Protocol* drop-down lists, enter the name of exchange server in the *Incoming Mail Server* field. |
| Domain | Enter the domain name. This field is applicable only when MAPI is selected in the protocol drop-down list. |
| CDO Host Machine | Enter the name of the CDO host machine. CDOConfig.exe is a tool, which comes with the J-Integra for Exchange SDK and is used for configuring CDO. Host where CDO is configured is called CDO host machine. |
| Enable SSL | Check this checkbox, if the specified mail server is a secure mail server. |
| Port | Default Port number of the Incoming Mail Server. You can change this if required |
| Subject | Subject of the mail. Here you can enter any subject. During execution of process flow, subject will be overwritten with the subject of the mail, received to Mail Event. |
| User ID | Username required to access the mailbox in which mail with specified subject arrives |
| Password | Password required to access the mailbox |
| Confirm Password | Re-enter the Password |
| Data Location | Data location specifies that whether data is in body of the email or in attached file |
| File Name | Name of the attached file, if the data is in attached file. |
| Leave a Copy on Server | If this option is enabled, a copy of mail is left on the server. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Mail Source activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the mail source activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.




8. Click **OK** to save the comments. This displays a screen confirming that the mail source activity has been updated successfully.



## Editing Excel Schema

Excel Schema is used to parse the data from an excel file attached with incoming mail. Excel Schema parses the data from excel file and converts it into intermediate XML format. In this Process Flow two Excel Schemas are used to parse the data. If the subject of the mail is **Format1**, **EvalPF_ExcelSchema_Format1** is used and if the subject of the mail is **Format2, EvalPF_ExcelSchema_Format2** is used.

***Steps to edit the Excel Schema (EvalPF_ExcelSchema_Format1) activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Excel**. The Manage Excel Schema screen is displayed with the list of existing Excel Schemas (refer to Figure 3.15).
4. Select the radio button adjacent to *EvalPF_ExcelSchema_Format1* activity and then click **Edit** link. This displays the Edit *EvalPF_ExcelSchema_Format1* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.8).

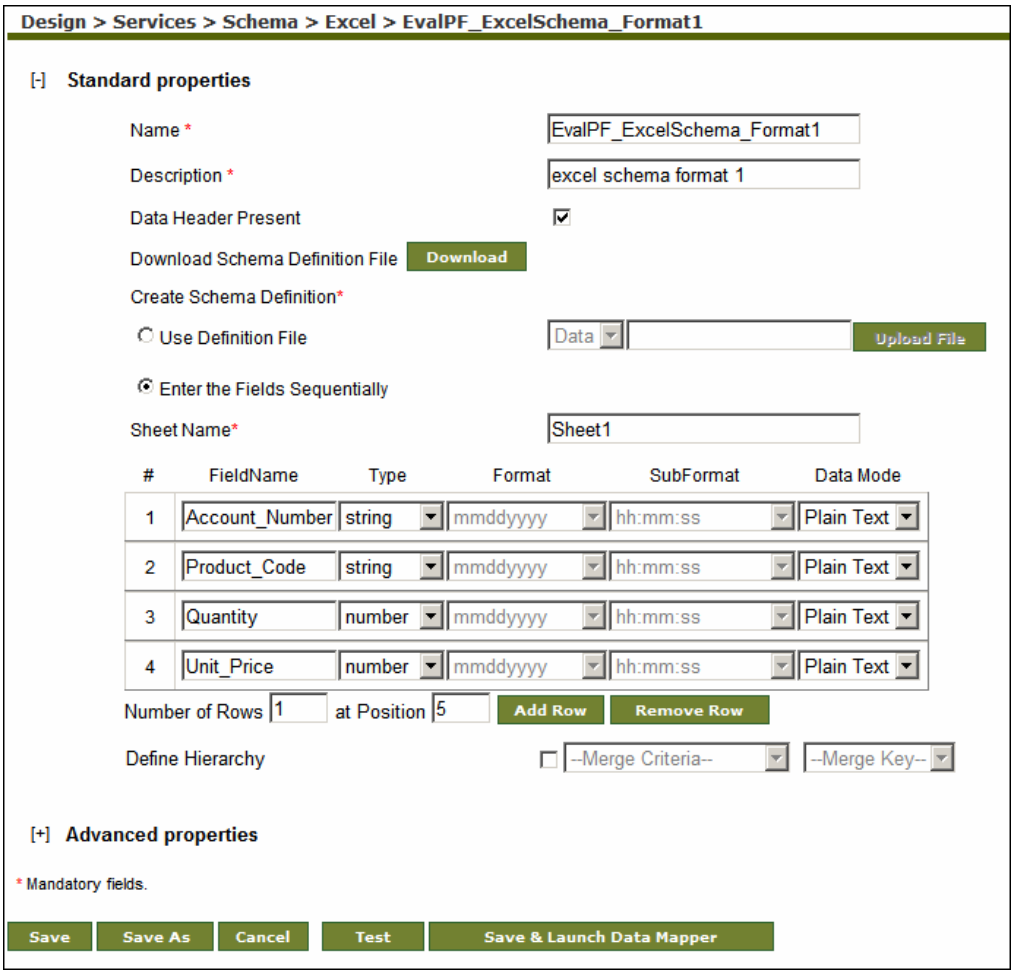



Figure 6.8: Edit *EvalPF_ExcelSchema_Format1* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.6: Details of Fields on Edit Excel Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Excel Schema |
| Description | Description of the Excel Schema |
| Data Header Present | Data Header usually contains the titles of the fields in a file. If you enable this option, Field Names of Excel Schema are written as column's name in target excel file |
| Sheet Name | Name specified here becomes Book Name of the excel sheet. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File<br>▪ Enter the Field Sequentially<br><br>Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |
| Field Name | Name of the Fields |
| Data Type | There are three data types: |

| | | |
|---|---|---|
| | String | String can be used for any type of data. |
| | Number | Contains numbers |
| | Date | Contains Date and Time |
| | Currency | Contains Currency Value |
| Format | If data type is *Date*, select the data format from *Format* drop-down list. | |
| SubFormat | If data type is *Date*, select time format from *SubFormat* drop-down list. | |
| Data Mode | If data is in Plain Text, select *Plain Text* option. If data is encrypted, then select *Encrypted* option. | |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Excel Schema activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the excel schema (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the excel schema has been updated successfully.

Repeat the same steps to edit *EvalPF_ExcelSchema_Format2*.

## Testing Excel Schema (EvalPF_ExcelSchema)

You can verify the excel schema activity at design time.

***Steps to verify schema activity***

1. Click **Test** button on the Edit Excel Schema screen. The Test Schema screen is displayed (refer to Figure 7.14).
2. Select the type of schema to test, from the *Type* drop-down list. By default, *Source* is selected.
3. Enter the full path (with file name and extension) of the source excel file in the *Source File Name* field.
4. Enter the full path of the XML target file, where it will be generated, in the *Target File Name* field.

5. Enter the full path of the XML file where errors will be stored, in the *Error File Name* field.
6. Click **Submit** button. This tests the validity of the excel schema.

## Editing Mapping Activity

Mapping is used to map data fields of source schema with data fields of target schema. In this Process Flow multiple schemas are used at target end.

***Steps to edit the Mapping activity (EvalPF_MappingTransformation_Format1):***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed with the list of existing mapping activities (refer to Figure 7.15).
4. Select the radio button adjacent to *EvalPF_MappingTransformation_Format1* activity and then click **Edit** link. This displays the Edit *EvalPF_MappingTransformation_Format1* activity screen with the name and description of the activity displayed in their respective fields (see Figure 6.9).

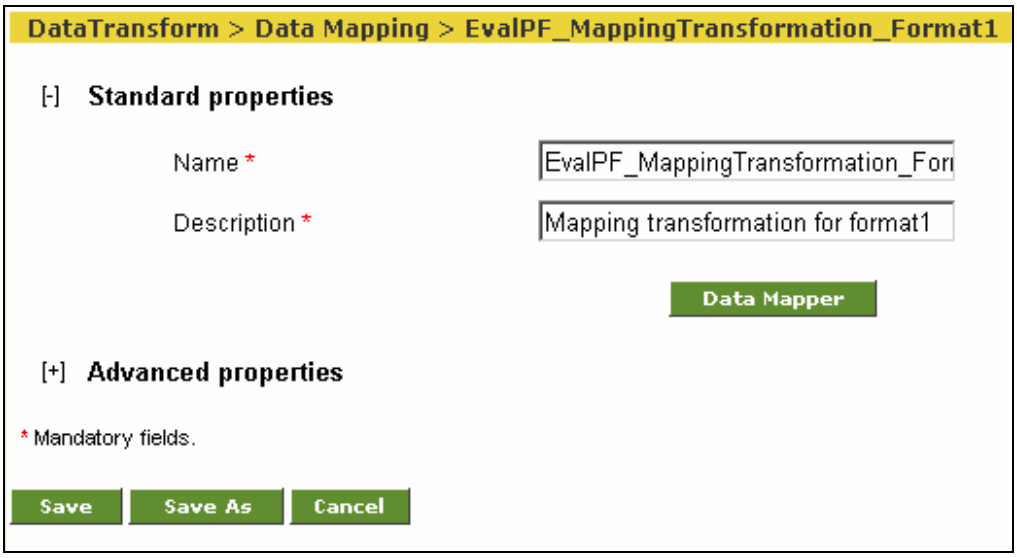



Figure 6.9: Edit *EvalPF_MappingTransformation_Format1* Activity

5. Click the **Data Mapper** button**.** The Data Mapper applet is displayed showing mapping between data fields of source and target schema (see Figure 6.10).

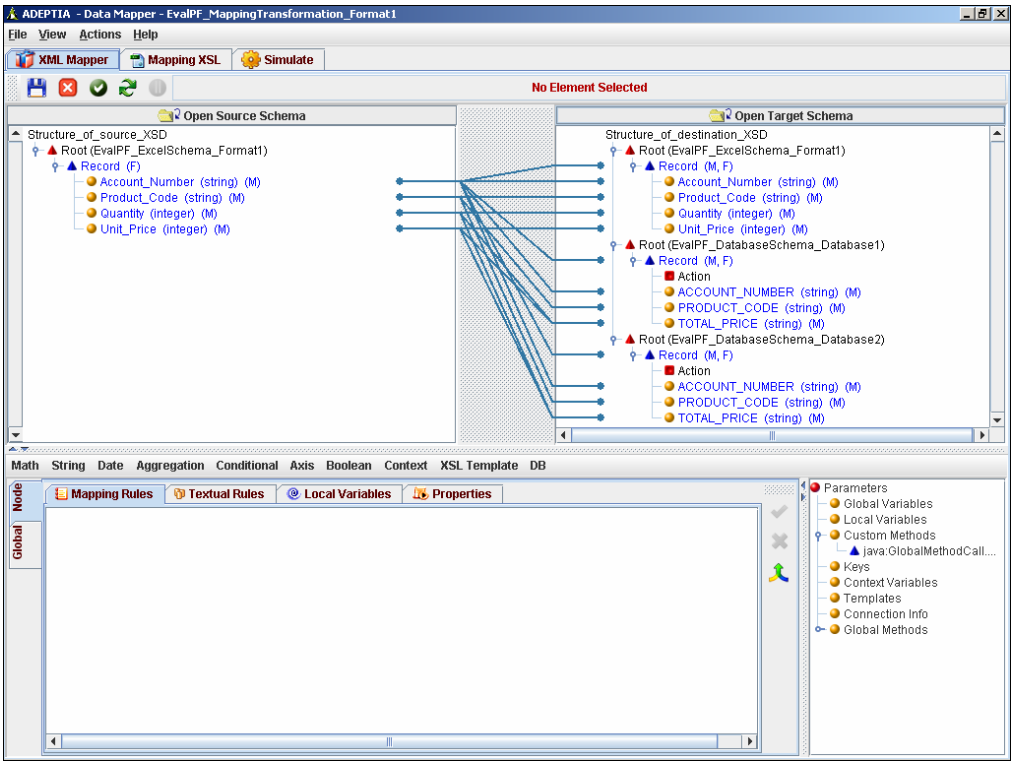



Figure 6.10: Edit *EvalPF_MappingTransformation_Format1* Activity in Data Mapper Applet

6. To view how mapping function is used, click any of the target elements (*Record*) in Target Panel. The mapping function used for the selected element is displayed in the Mapping Graph Area (see Figure 6.11).

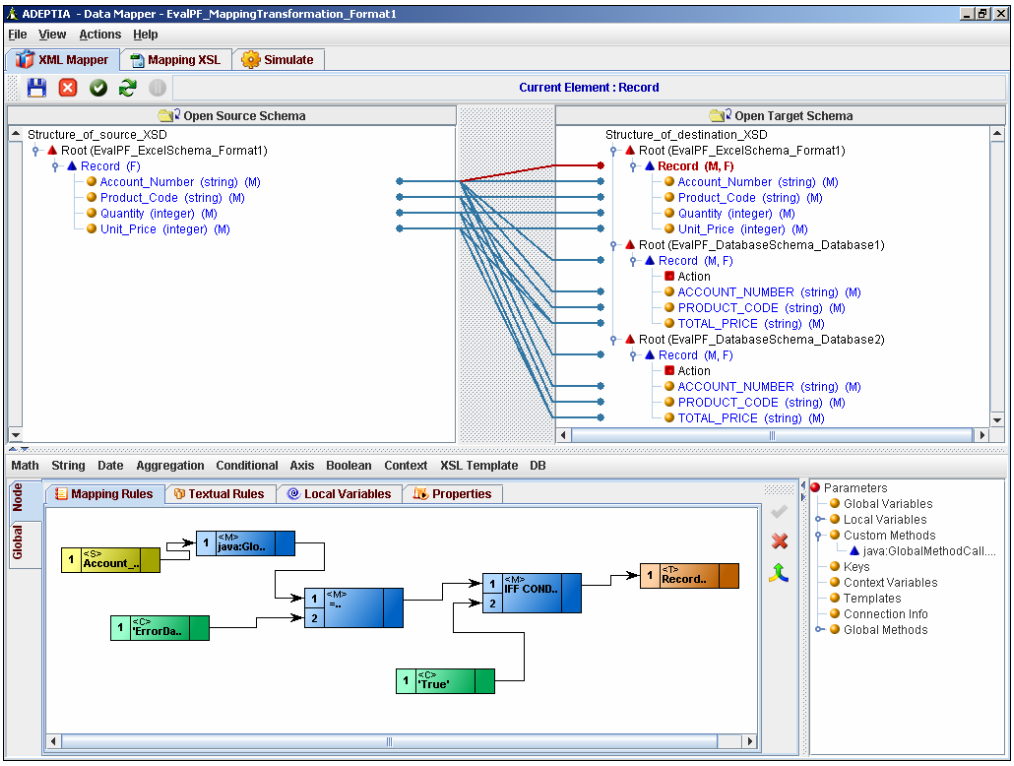



Figure 6.11: View Mapping Functions Used in *EvalPF_MappingTransformation_Format1* Activity

7. To analyze the impact of mapping function on output records, right-click any of function node (**IFF Condition**) and select **Information** (see Figure 6.12).

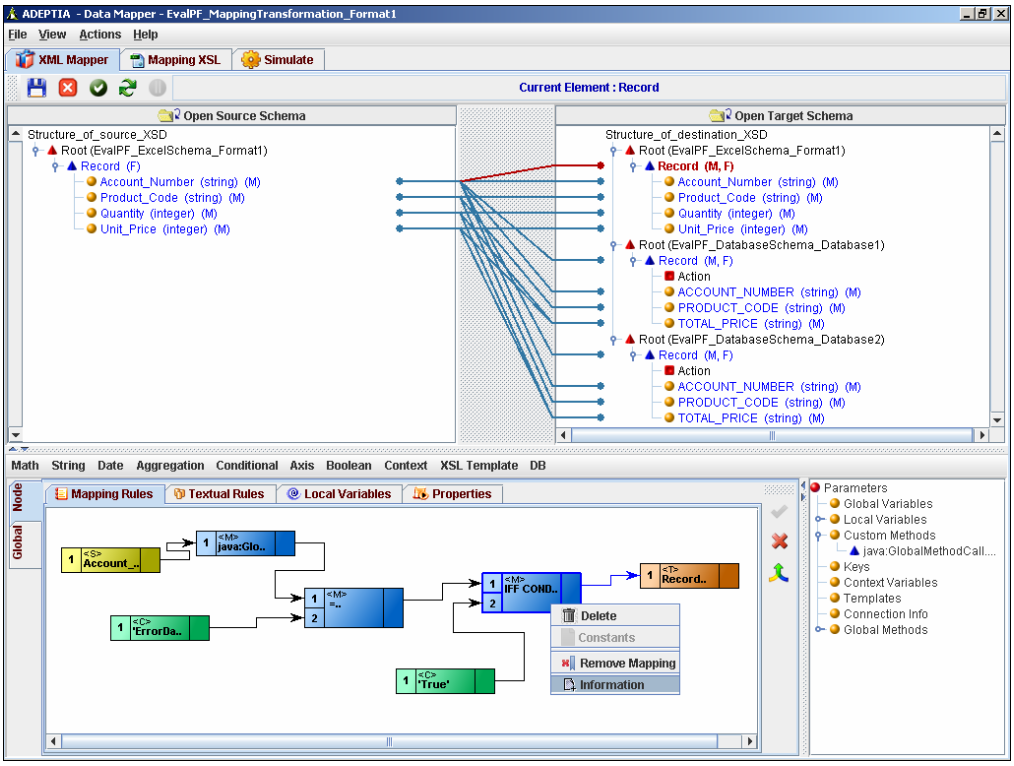



Figure 6.12: Select Mapping Function Information

8. A dialog box is displayed that shows information about that mapping function (see Figure 6.13).

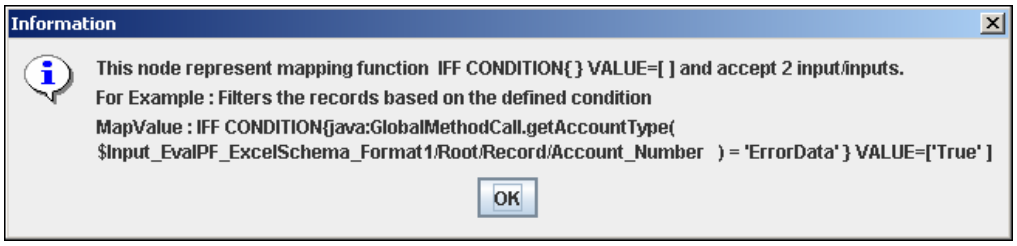



Figure 6.13: View Mapping Function Information

9. Click the **OK** button to close the dialog box.
10. Make the necessary changes to the mapping between the source and target data fields.
11. Once you have made the required changes, save the mapping by clicking the **File** menu and selecting **Save**. A dialog box is displayed confirming that the Mapping activity has been saved successfully.
12. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping (refer to Figure 7.18).
13. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

14. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.

| | |
|---|---|
| | Alternately, you can save the mapping by clicking **Save** (  ) button on the Tool Bar. |

15. Exit the Data Mapper applet by clicking the **File** menu and selecting **Exit**.

| | |
|---|---|
| | Repeat the same steps, to edit *EvalPF_MappingTarnsformation_Format2* activity. |

.

| | |
|---|---|
| | To know, how this mapping activity has been created, refer to **Creating Mapping Activity** section. |

## Editing Database Driver

## (EvalPF_DatabseDriver_SQLServer)

Database Driver is used to specify the type of database and to upload driver jar files required to connect to that database.

***Steps to edit Database Driver:***

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **Database Driver**. The Manage Database Driver screen is displayed with the list of existing Database Drivers (refer to Figure 7.19).
3. Select the radio button adjacent to *EvalPF_DatabseDriver_SQLServer* activity and then click **Edit** link. This displays the Edit *EvalPF_DatabseDriver_SQLServer* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.14).

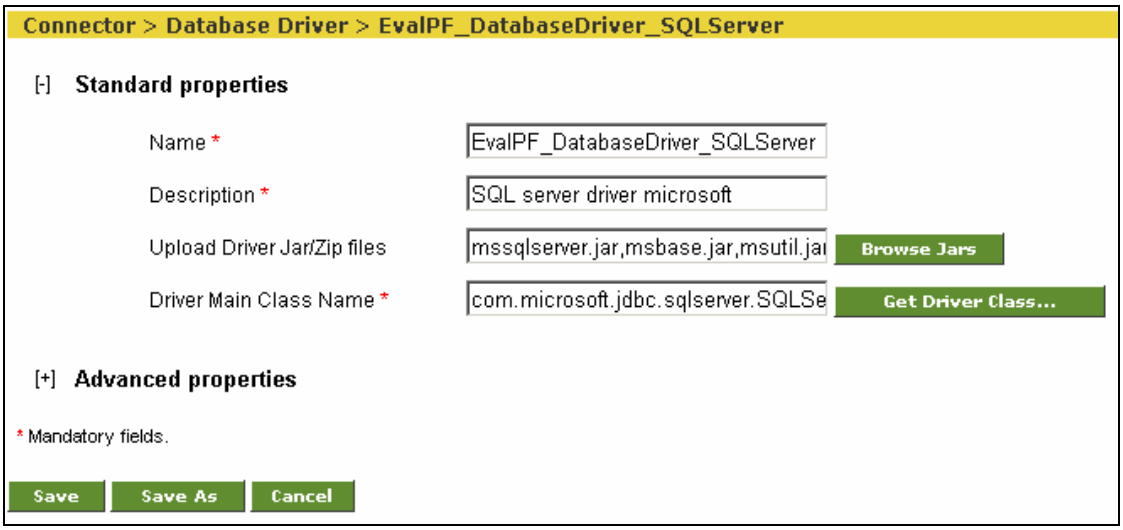



Figure 6.14: Edit *EvalPF_DatabaseDriver_SQLServer*

A detailed description of fields on this screen is explicated in the table below.

Table 6.7: Details of Fields on Edit Database Driver Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Driver |
| Description | Description of the Database Driver |
| Upload Driver Jar Files | JDBC Driver files, which are used to connect Database Server. Click the **Browse Jars** button to select Jar files. Following is the list of databases and the required Jar files:<br>Oracle — Classes12.jar<br>IBM DB2 (Ver 7.1) — db2java.zip (7.1 version)<br>IBM DB2 (Ver 8.1) — db2jcc.jar<br>MS SQL — msbase.jar, mssqlserver.jar and msutil.jar<br>JTDS- SQL Server — Jtds.jar<br>HSQL DB — hsqldb-1.7.2.jar |
| Driver Main Class Name | Driver Main Class Name is a fully qualified java class name for the main database driver class. The driver class name typically starts with a com., net. or org. followed by the company domain, for example the JDBC driver class from mysql.com is called com.mysql.jdbc.Driver. Click the **Help** button to select Driver Main Class Name from the drop-down list. Following is the list of Driver Main Class Name of different databases: |

| | | |
|---|---|---|
| | Oracle | oracle.jdbc.driver.OracleDriver |
| | SQLSERVER | com.microsoft.jdbc.sqlserver.SQLServerDriver |
| | IBM-DB2SERVER-V7 | COM.ibm.db2.jdbc.net.DB2Driver |
| | IBM-DB2-SERVER-V8 | com.ibm.db2.jcc.DB2Driver |
| | HSQL | org.hsqldb.jdbcDriver |
| | JTDS-SQLSERVER | net.sourceforge.jtds.jdbc.Driver |
| | AS400 | sun.jdbc.odbc.JdbcOdbcDriver |
| | MYSQL | com.ibm.as400.access.AS400JDBCDriver |
| | | com.mysql.jdbc.Driver |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Driver has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Database Driver (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the database driver has been updated successfully.

In this sample Process Flow *SQL Server* is used as target database. If you want to use another database, upload the appropriate Jar files and select Driver Main Class Name for that database.

## Editing Database Info

### (EvalPF_DatabseInfo_SQLServer)

Database Info activity is used to specify JDBC URL and Username and Password to access the database. Database Info uses Database Driver to connect to specified Database Server.

***Steps to edit the Database Info:***

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **Database Info**. The Manage Database Info screen is displayed with a list of existing **Database Info** (refer to Figure 7.21).
3. Select the radio button adjacent to *EvalPF_DataBaseinfo_SQLServer* activity and then click **Edit** link. This displays the *EvalPF_DataBaseinfo_SQLServer*

activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.15).

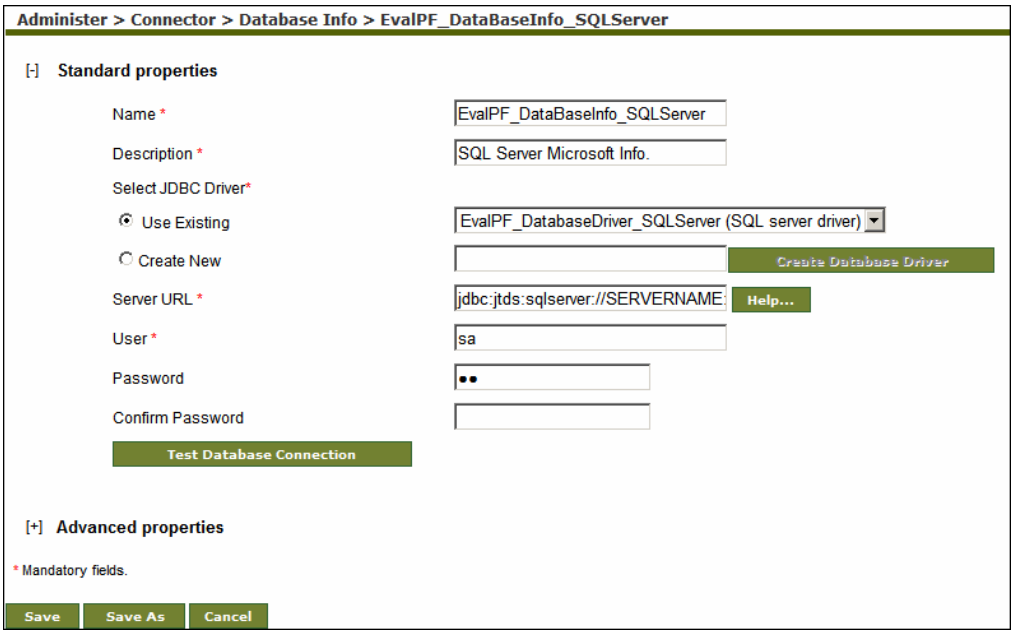



Figure 6.15: Edit *EvalPF_Databaseinfo_SQLServer*

A detailed description of fields on this screen is explicated in the table below.

Table 6.8: Details of Fields on Edit Database Info Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Info |
| Description | Description of the Database Info |
| JDBC Driver | Database Driver is created to connect to the database Server. For more details refer to section **Editing Database Driver**. |
| Server URL | Server URL or JDBC URL points to a specific database on a specified database server. There is no standard for JDBC URL. Every JDBC driver uses a slightly different syntax. For Example a JDBC URL for a MySQL database using the `com.mysql.jdbc.Driver` direct from MySQL might look like this:<br>`jdbc:mysql://localhost/databaseName`. To specify Server URL, Click the **Help** button and enter the following information:<br>Database Type — Type of the database<br>Host Name — Name of the server on which database server is running<br>Port — Port at which database server is running<br>Database Name — Name of the database |
| | Following is the list of Server URL's of different databases: |

| | | |
|---|---|---|
| | Oracle | jdbc:oracle:thin:@databaseserver:1521:orcl |
| | IBM DB2 (Ver 7.1) | jdbc:db2://databaseserver:6789/TOOLSDB |
| | IBM DB2 (Ver 8.1) | jdbc:db2://databaseserver:50000/TOOLSDB |
| | MS SQL | jdbc:microsoft:sqlserver://databaseserver:1433;DatabaseName=master |
| | SQL JTDS | jdbc:jtds:sqlserver://databaseserver:1433/master |
| | MS Access | jdbc:odbc:Driver={MicroSoft Access Driver (*.mdb)}; DBQ=c:/test/db1.mdb |
| | MS Excel | Jdbc:odbc:*ExcelJDBCTest*<br>where *ExcelJDBCTest* is the ODBC object that you need to create using DSN. |
| | HSQL DB | jdbc:hsqldb:hsql://databaseserver:2476 |
| | Here **database server** is the name of the server where database is running. | |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Info has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Database Info (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the database Info has been updated successfully.

In this sample Process Flow *SQL Server* is used as target database. If you want to use another database, select appropriate Server URL for that database.

## Editing Database Schema

Database Schema defines the structure of database table. Database Schema is used to define how records can be read from a database table or can be written into a database to table. In this sample process Database Schema is being used at target end. At the target end it converts data from intermediate XML format into database specific format. Database Schema uses Database Info activity to connect to the database Server. This should be noted that Database Schema does not

directly take part in creation of Process Flow. It is used by Database Target activity and the Database Target activity is used in Process Flow.

***Steps to edit the Database Schema (EvalPF_DatabaseSchema_Database1) activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Database**. The Manage Adv. Database Schema screen is displayed with a list of existing Database Schemas (refer to Figure 7.23).
4. Select the radio button adjacent to *EvalPF_DatabaseSchema_Database1* activity and then click **Edit** link. This displays the Edit *EvalPF_DatabaseSchema_Database1* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.16).

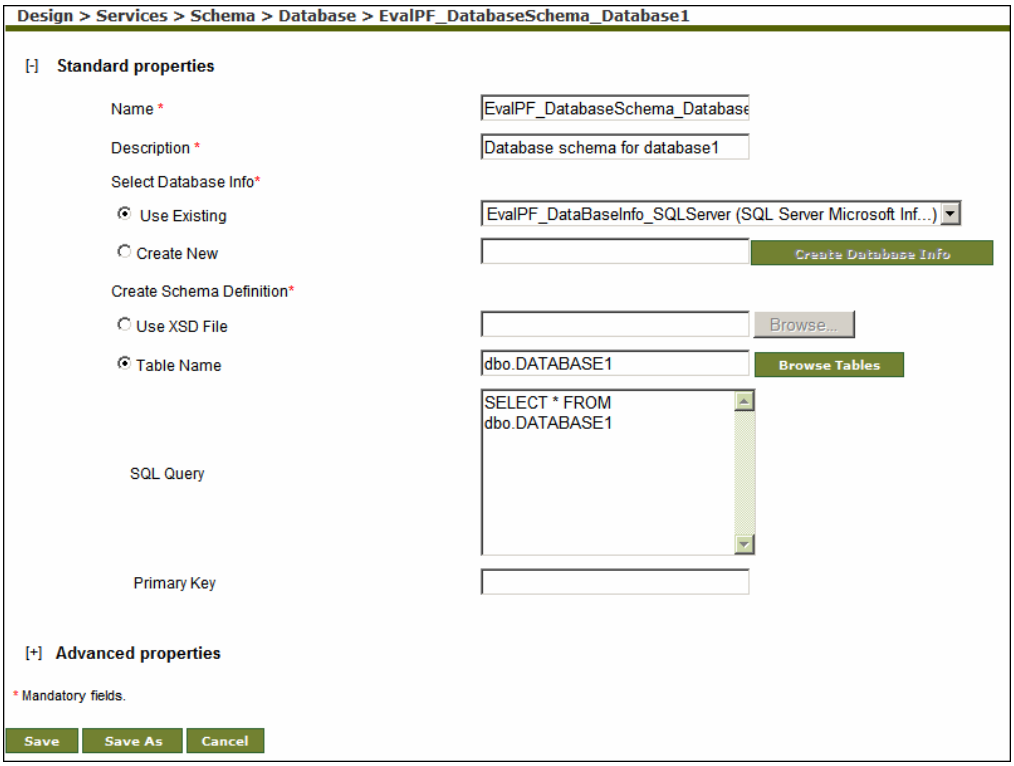



Figure 6.16: Edit *EvalPF_DatabaseSchema_Database1* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.9: Details of Fields on Edit Database Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Schema |
| Description | Description of the Database Schema |
| Database Info | Database Info created to connect to the specified Database Server. For more details refer to **Editing Database Info** |

| | |
|---|---|
| | section. |
| Create Schema Definition | Schema Definition can be created using one of the following options:<br>▪ Use XSD File<br>▪ Table Name<br>Database Schema used in this Process Flow is created using second option i.e. Table Name. To select database tables, select **Table Name** radio button and then Click the **Browse Tables**. *Select Table* screen is displayed with the list of database Table. Select the required table and click **Get Columns** button. Click **Close** button to close the *Select Table* screen and return to Database Schema screen.<br>SQL Query box automatically gets populated after selecting database tables. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Source has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database source activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the database source activity has been updated successfully.

Repeat the same steps to edit *EvalPF_DatabaseSchema_Database2* activity.

## Editing Database Target

Database Target is used to insert data into a database server. Database Target uses Database Info for Server URL and login information and Database Schema to get information for database tables and data type etc.

***Steps to edit the Database Target (EvalPF_DatabaseTarget_Database1):***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **Database**. The Manage Database Target screen is displayed with the list of existing Database Target activities (refer to Figure 7.25).

4. Select the radio button adjacent to *EvalPF_DatabaseTarget_Database1* activity and then click **Edit** link. This displays the Edit *EvalPF_DatabaseTarget_Database1* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.17).

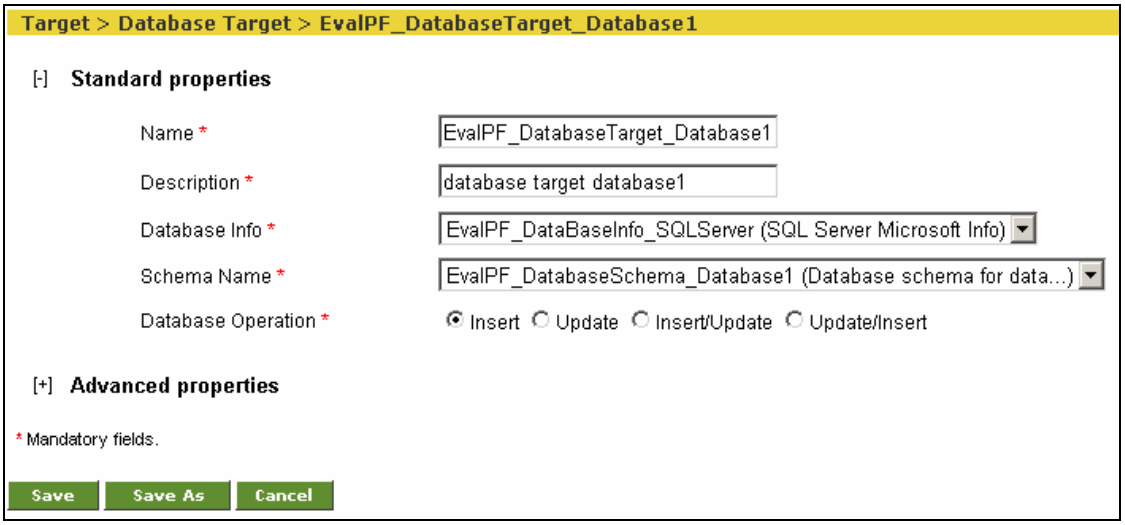



Figure 6.17: Edit *EvalPF_DatabaseTarget_Database1* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.10: Details of Fields on Edit Database Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Target |
| Description | Description of the Database Target |
| Database Info | Database Info created to connect to the specified Database Server. For more details refer to Editing Database Info section. |
| Schema Name | Database Schema, which describes the structure of database table. For more details refer to Editing Database Schema section. |

| | |
|---|---|
| Database Operation | Database operation specifies how data records are inserted into database tables. Select one of the following database operations:<br>▪ Insert<br>▪ Update<br>▪ Insert/Update<br>▪ Update/Insert<br><br>When **Insert** option is selected, records are inserted into the database tables. If records already exist in the database table, new records are added in the database table along with existing records. When a column e.g. *Account Number* of database table is marked as Primary Key, more than one record cannot exist in the database table for the same Account Number. In this case if data of an Account Number already exist, insert operation fails and data is not inserted into the database.<br><br>**Update** option is selected when you want to update the existing record. To use updated option a column of database must be marked as Primary Key. When Update option is selected, database target first checks which column of the database table is marked as Primary Key. Suppose Account Number column is marked as Primary Key. Now database target check whether data of a particular Account Number exist or not. If data for that account number already exists, database target updates the existing records. If none of the column is marked as Primary key, Update operation fails.<br><br>When **Insert/Update** option is selected, database target first tries to insert the data into database table. If insert operation fails, database target tries to update the data.<br><br>When **Update/Insert** option is selected, database target first tries to update the database table. If update operation fails, database target tries to insert the data.<br><br>*Note:* To know, how to mark a column of a database table as Primary Key, refer to the documentation of Database Server you are using.<br><br>In this sample Process Flow **Update/Insert** option is used. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Target activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database target activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the database target activity has been updated successfully.

| | |
|---|---|
| | Repeat the same steps to edit *EvalPF_DatabaseTarget_Database2* activity. |

## Editing File Target Activity

In this Process Flow file target activities are used to store error records. *EvalPF_FileTarget_ErrorRecords* activity is used to store those records, which do not match either of the Account Number format (i.e. 99-999999 or AA-99999). Records, which cannot be inserted into the Database1, are stored in *EvalPF_FileTarget_InsertError.* Records, which cannot be updated into the Database2, are stored in *EvalPF_FileTarget_UpdateError.*

***Steps to edit the File Target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **File**. The Manage File Target screen is displayed with the list of existing File Target activities (refer to Figure 3.17).
4. Select the radio button adjacent to *EvalPF_FileTarget_ErrorRecords* activity and then click **Edit** link. This displays the Edit *EvalPF_FileTarget_ErrorRecords* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.18).

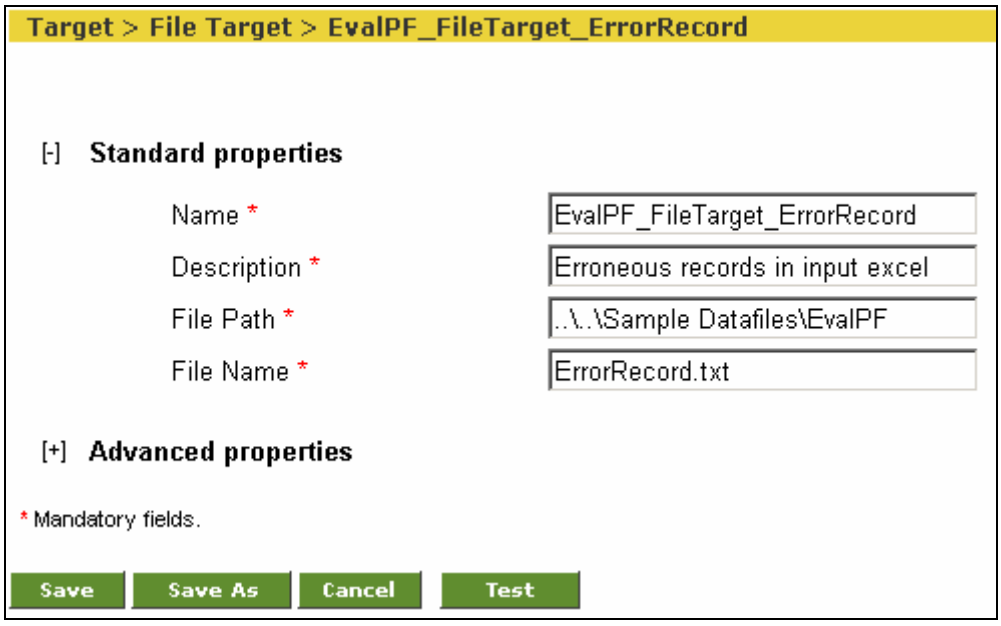


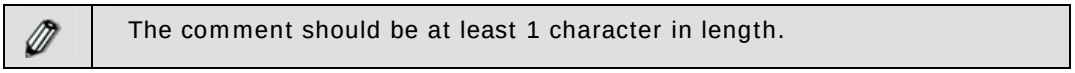

Figure 6.18: Edit *EvalPF_FileTarget_ErrorRecord* Activity

A detailed description of fields on this screen is explicated in the table below.

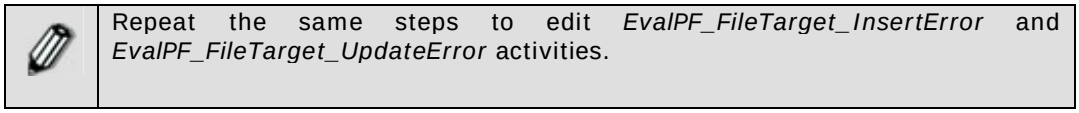
Table 6.11: Details of Fields on Edit File Target Screen

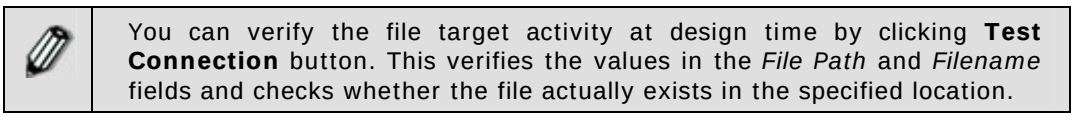
| Field Name | Field Description |
|---|---|
| Name | Name of the File Target |
| Description | Description of the File Target |
| File Path | Path of the target file.<br>For example:<br>**../../Solutions/Demo/EvalPF/**<br>In the activities *EvalPF_FileTarget_InsertError* and *EvalPF_FileTarget_UpdateError*, the path will be<br>../../Solutions/Demo/EvalPF/<br>and<br>../../Solutions/Demo/EvalPF/ respectively. |
| File Name | Name of the target file. |

5. Make the necessary changes.
6. Once you have the made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file target activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.




8. Click **OK** to save the comments. This displays a screen confirming that the file target activity has been updated successfully.




## CREATING MAPPING ACTIVITY

Mapping is used to map data fields of source schema and target schema. In this Process Flow, two mapping activities are used. When the subject of the incoming mail is Format1, the mapping activity EvalPF_MappingTransformation_Format1 is used. When the subject of the incoming mail is Format2, the mapping activity EvalPF_MappingTransformation_Format2 is used.

In both the mapping activities, Excel Schema is used at the source end. At the target end three schemas (one excel schema that is also being used at source end and two database schemas) are used. Excel Schema is used to define the structure of target excel file, which is created in case format of the Account_Number field is not correct. Database Schemas are used to define the structure of two different database tables.

In both the mapping activity a Global Method is used, which calls a Java Method. This java method returns different values based on the format of Account_Number. If the format of account number is *99-999999,* the Java method returns the value *NumericNumeric*. If the format of account number is *AA-999999*, the Java method returns the value *AlphabetNumeric*. Here AA denotes any alphabet and 999999 denotes any numeric value. If the format of Account_Number is different than either of these two formats, the Java Method returns the value *ErrorData*. The records are sent to one of the three target activities depending upon the value returned by the Java Method as shown in the table below.

Table 6.12: Target Activities and Java Method Values

| Format of Account_Number | Values returned by Java Method | Target Activities |
|---|---|---|
| 99-999999 | NumericNumeric | EvalPF_DatabaseTarget_Database1 |
| AA-999999 | AlphabetNumeric | EvalPF_DatabaseTarget_Database2 |
| Other Format | ErrorData | EvalPF_FileTarget_ErrorRecord |

***Steps to create the Mapping Activity:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed with the list of existing mapping activities (refer to Figure 7.15).
4. Click **New Mapping Transformation** button. The Create Mapping Activity screen is displayed (refer to Figure 3.21).
5. Enter the name and description of the new mapping activity in the *Name* and *Description* fields respectively.
6. Click **Data Mapper** button. This displays the Data Mapper applet (refer to Figure 3.22).
7. Click **Open Source Schema** button to load the source schema. This displays the Select Schema screen (refer to Figure 3.23).
8. Select the Source Schema (*EvalPF_ExcelSchema_Format1*) checkbox from the *Source Schema* list and click **Load** button. This loads the selected schema in the Source Panel of the Data Mapper applet.
9. Similarly, load the Target Schemas (*EvalPF_ExcelSchema_Format1, EvalPF_DatabaseSchema_Database1, EvalPF_DatabaseSchema_Database2*) from the *Target Schema* list.
10. Click (  ) to expand the tree structure for Source Schema and the Target Schema (see Figure 6.19).

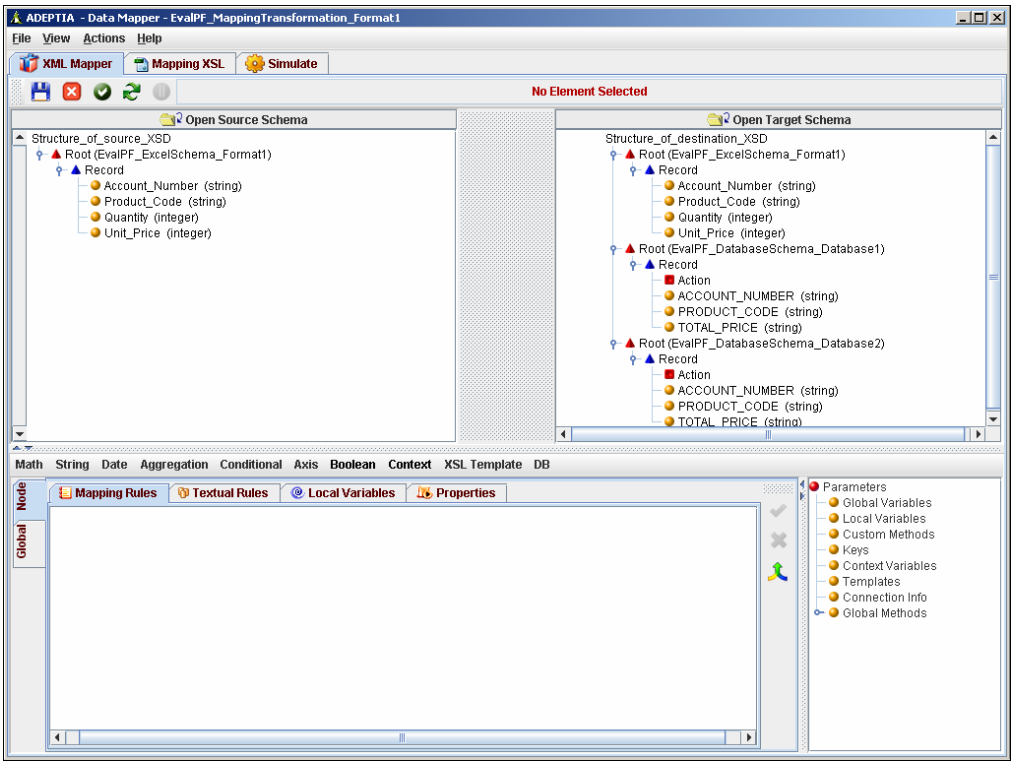



Figure 6.19: Expand Source and Target Schemas

11. Select the **Record** element of the first target schema and then click **Properties** tab in the Mapping Graph Area.

12. Click the **For Each** field and then double-click the **Record** element of the source schema. Click **Save Properties** to save the *For Each* property (see Figure 6.20).

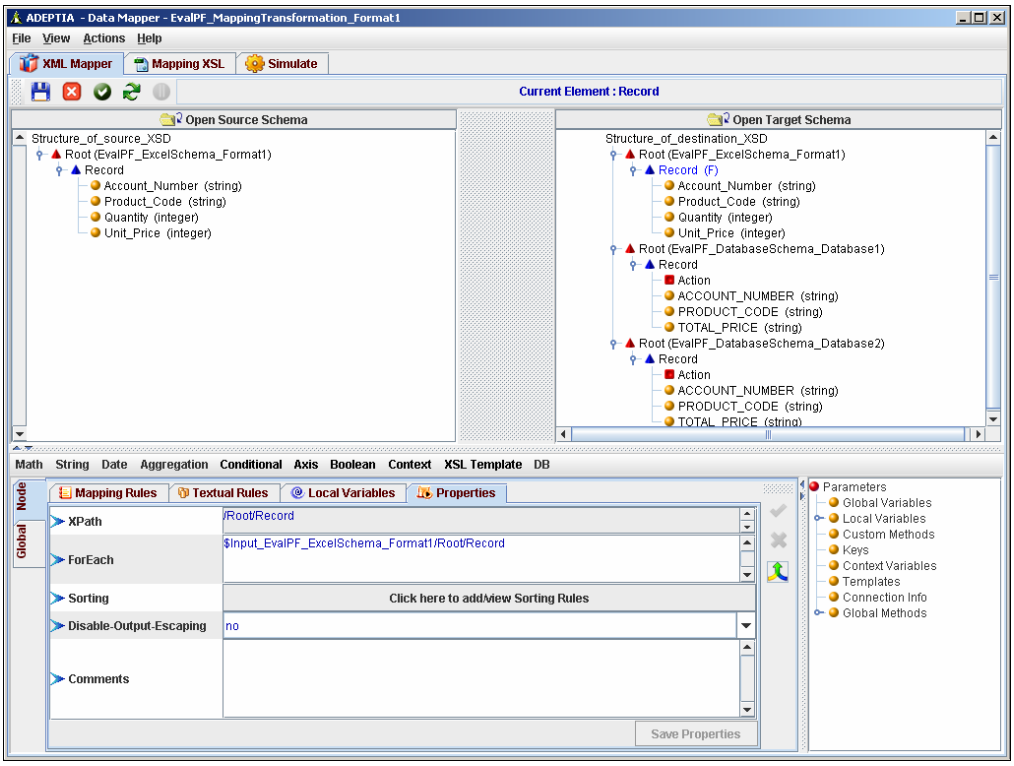



Figure 6.20: Apply For Each Property

13. Similarly, apply the *For Each* property on the **Record** element of the second and third target schemas.

14. Click the **Account_Number** node in the Source Panel, and drag the mouse pointer to the **Account_Number** node of the first schema in the Target Panel (see Figure 6.21).

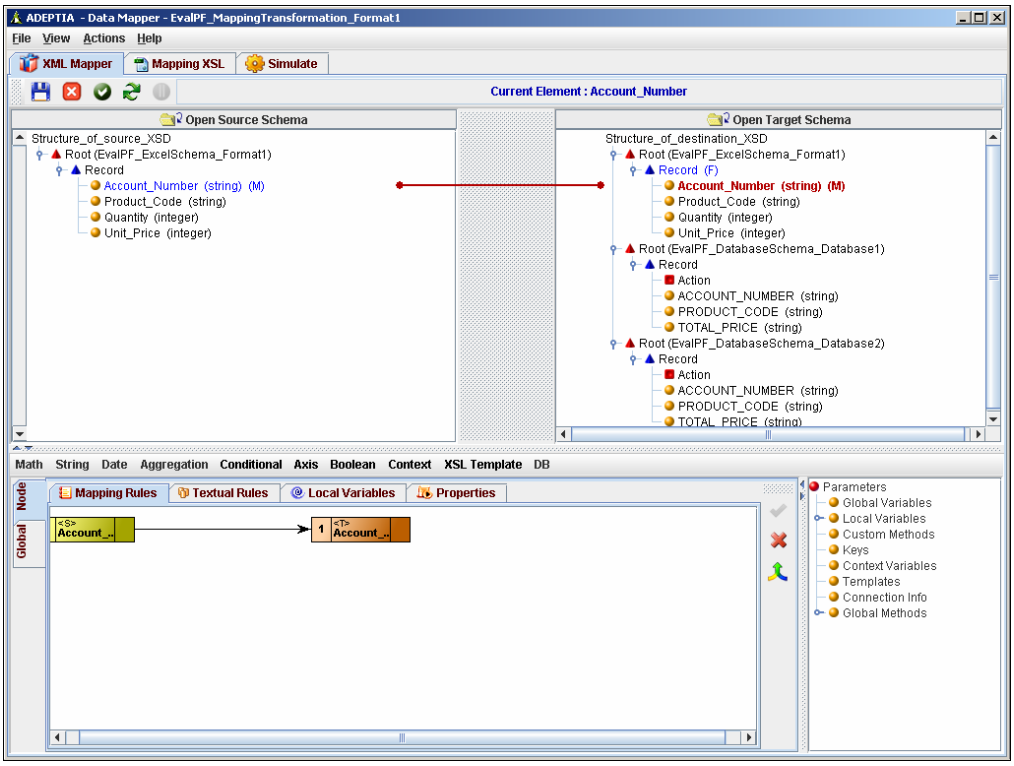



Figure 6.21: Map Account Numbers

15. Similarly, click the **Product_Code** node in the Source Panel and drag the mouse pointer to the **Product_Code** node of the first schema in the Target Panel.

16. Click the **Quantity** node in the Source Panel and drag the mouse pointer to the **Quantity** node of the first schema in the Target Panel.

17. Now click the **Unit_Price** node in the Source Panel and drag the mouse pointer to the **Unit_Price** node of the first schema in the Target Panel (see Figure 6.22).

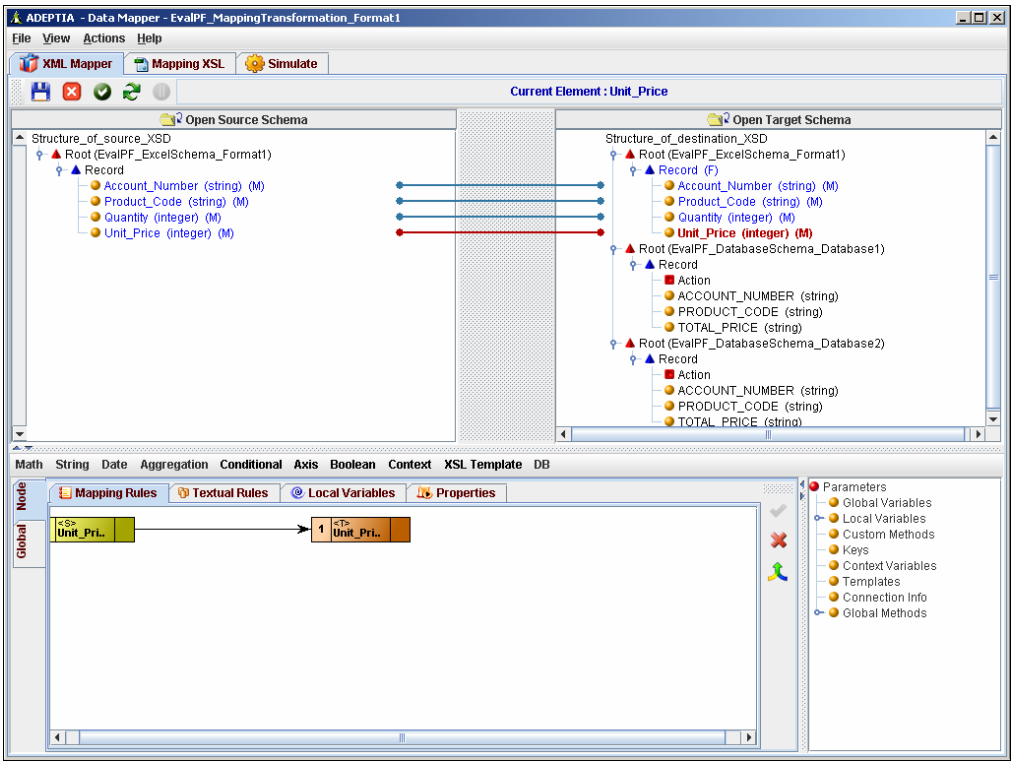



Figure 6.22: Map Source and Target Elements of First Target Schema

18. Similarly, map the fields of the schema in the Source Panel to the fields of the second and third schemas in the Target Panel (see Figure 6.23).

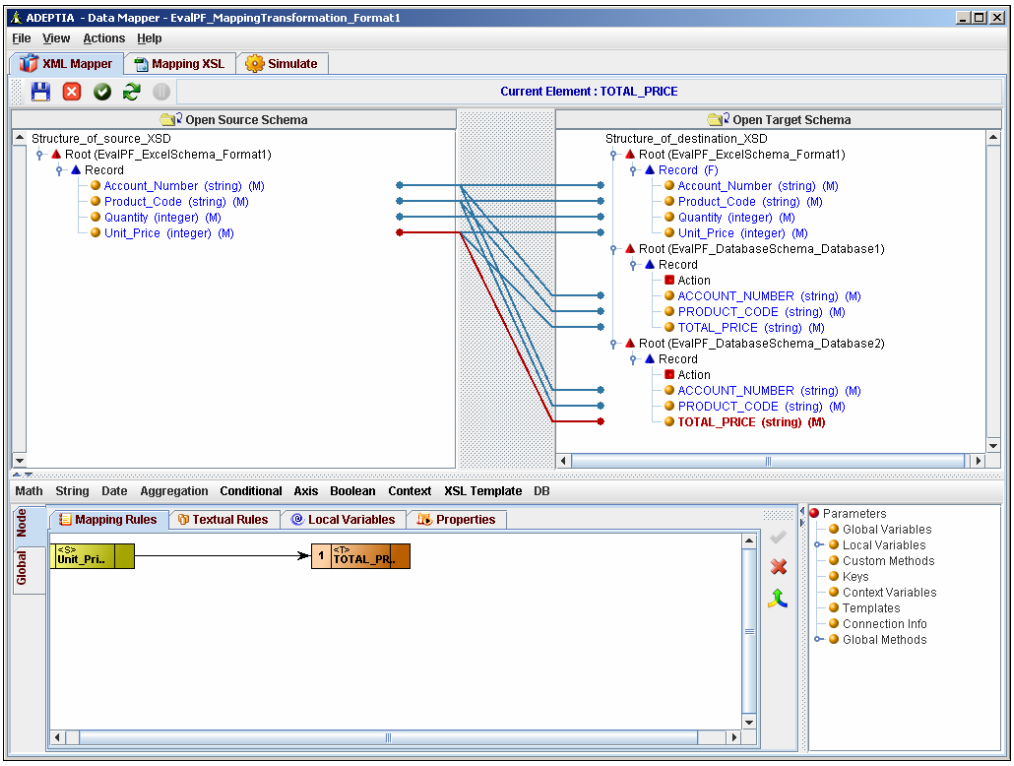



Figure 6.23: Mapping Source and Target Elements of All Target Schemas

19. To filter the records based on *Account_Number* field of the source schema, Custom Method is used.
20. To define a Custom Method, click the **Global** tab in the Mapping Graph Area.
21. Click the **Custom Methods** tab. The Custom Methods Panel is displayed (see Figure 6.24).

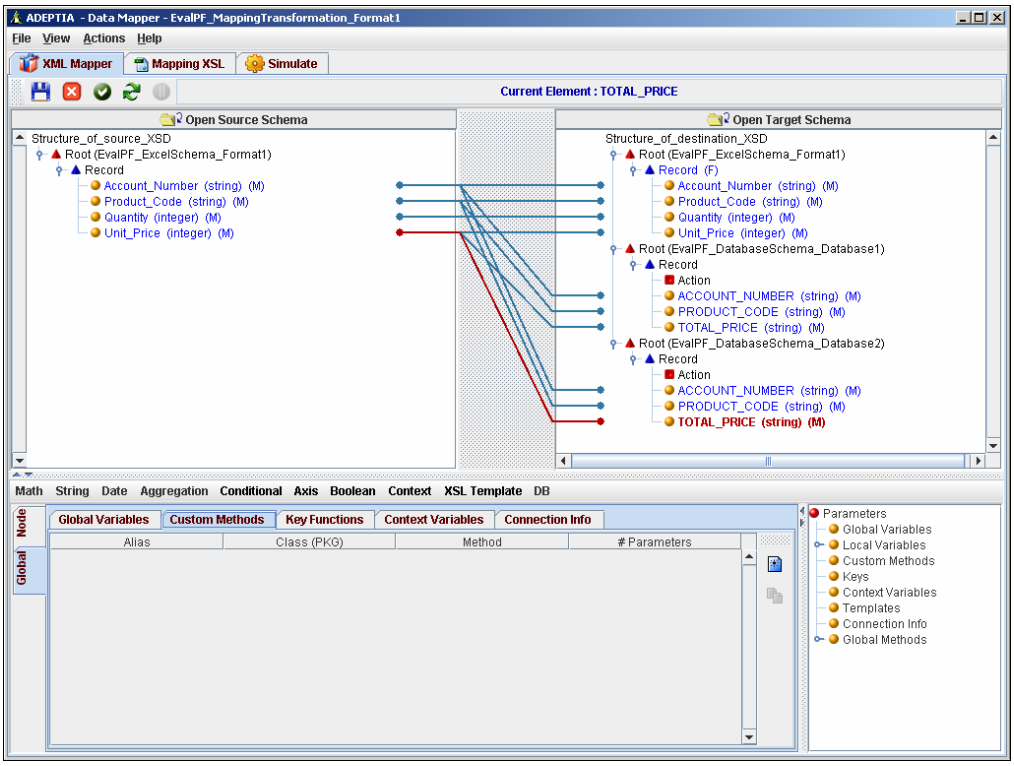



Figure 6.24: Custom Methods Pane

22. Click the **Add Method** ( ) button to add a Custom Method. A row is inserted with following columns (see Figure 6.25):

    - Alias
    - Class (PKG)
    - Method
    - # Parameters

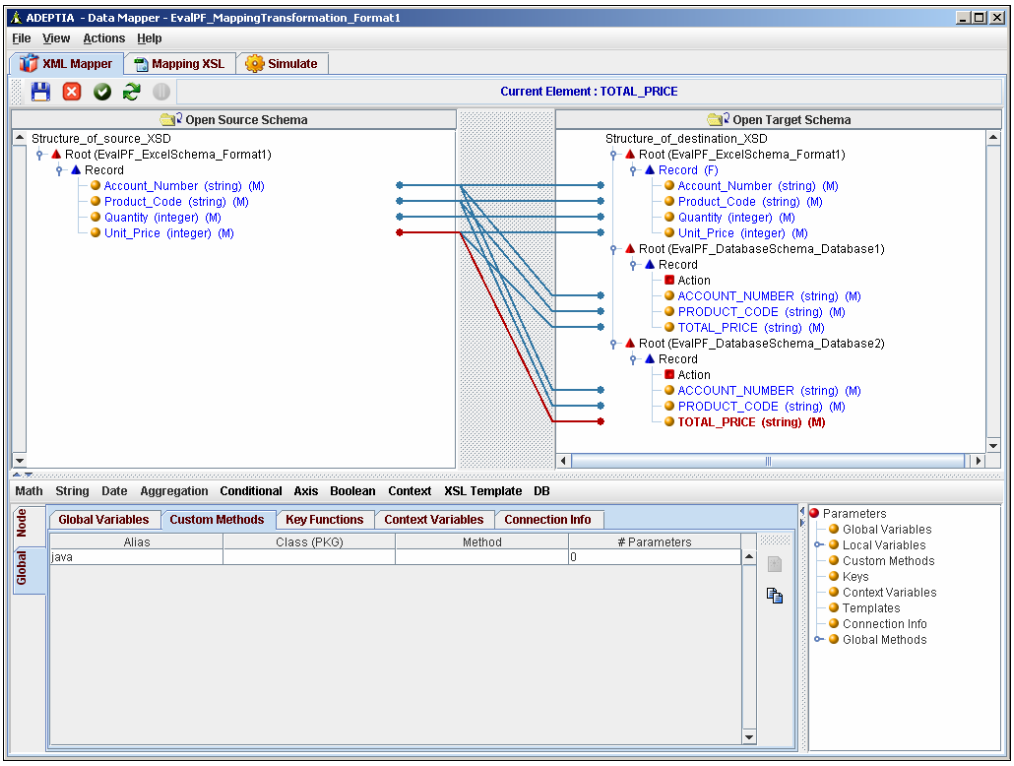



Figure 6.25: Create Custom Method

23. *Alias* column automatically gets populated with value 'java'.
24. Click the *Alias* field, press [**Tab**] or [**Enter**] key to go to the *Class (PKG)* field.
25. Enter the fully qualified name of the Java class (*CustomMethodCall*) in the *Class (PKG)* field.
26. Press the [**Tab**] or [**Enter**] key to go to the *Method* field and enter the name of the method (*getAccountType*) in the *Method* column.

| | |
|---|---|
|  | The Java Method specified here is stored in **../serverkernel/CustomClasess** folder. |

27. Press the [**Tab**] or [**Enter**] key to go to the # *Parameter* field and enter the number of arguments (1) taken by Method in the # *Parameter* column.
28. Press the [**Tab**] or [**Enter**] key to return to the *Alias* field. This will save the added Custom Method (see Figure 6.26).

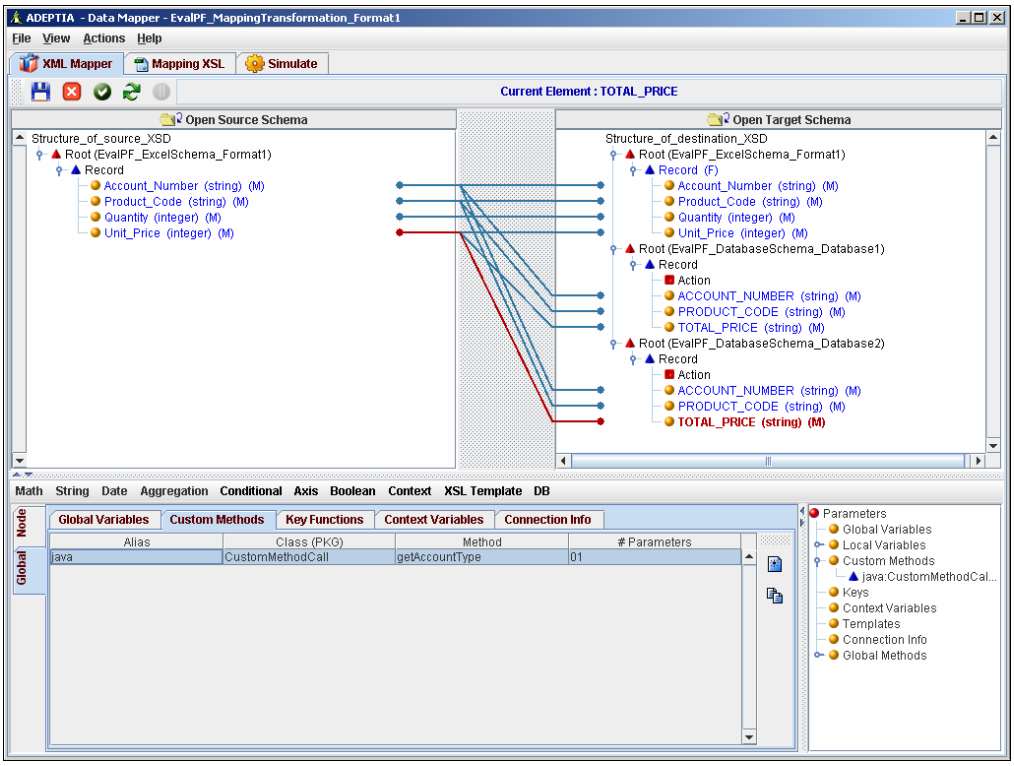



Figure 6.26: Save Custom Method

| | |
|---|---|
|  | The saved custom method is displayed under *Custom Methods* in the Parameters Panel. |

29. Click the **Node** tab to map above declared Custom Method to the *Account_Number* node. The Mapping Graph Area is displayed.
30. Select the node **Record** element of first target schema. The *Record* node is shown in the Mapping Graph Area.
31. Double-click the **Account_Number** node in the Source Panel. The *Account_Number* node is shown in the Mapping Graph Area.
32. Expand the *Custom Methods* tree in the Parameters Panel. Double click the defined Custom Method. The selected Custom Method Node is displayed in the Mapping Graph Area (see Figure 6.27).

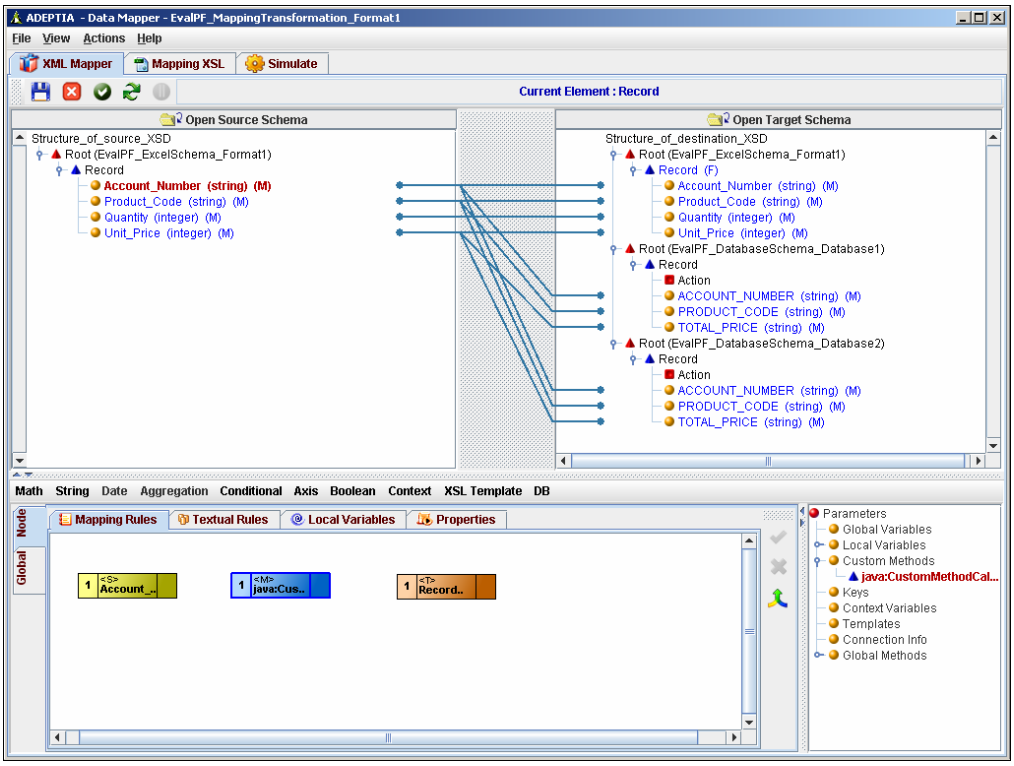



Figure 6.27: Select Custom Method

33. To add the constant value (**'***Errordata***'**), right-click the blank space in the Mapping Graph area and select the **Constant** option. A *Constant* node is displayed in the Mapping Graph area.

34. Double-click the **Constant** node. The Input dialog box is displayed (refer to Figure 3.41).

35. Enter the required constant value *Errordata* in the *Enter the Value* field.

36. Check the *Add Quotes* checkbox and click the **OK** button. The entered value is shown in the *Constant* node (see Figure 6.28).

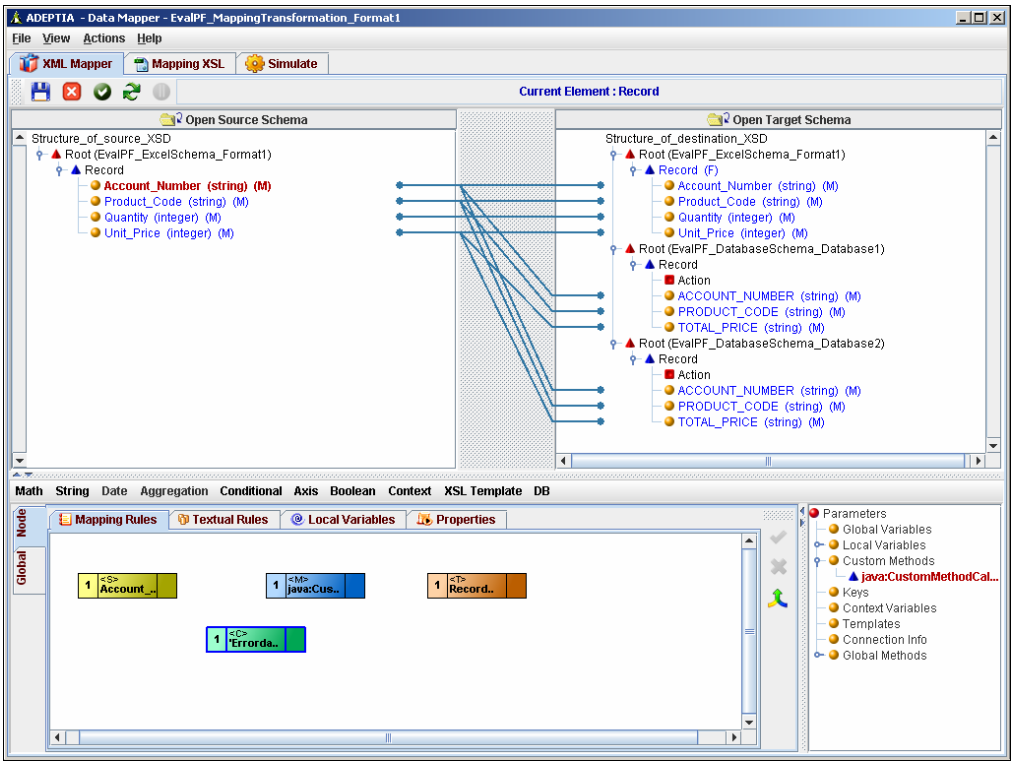



Figure 6.28: Constant Value Entered

37. Click the **Boolean** mapping function and select **Equal =** sub-function. The *Equal* function node is displayed in the Mapping Graph Area.

38. Create a link from the output of the *Account_Number* node to the input of the *Custom Method* node.

39. Create a link from the *Custom Method* node to the first input of the *Equal* function node.

40. Create a link from the output of the *Constant* function node to the second input of the *Equal* function node (see Figure 6.29).

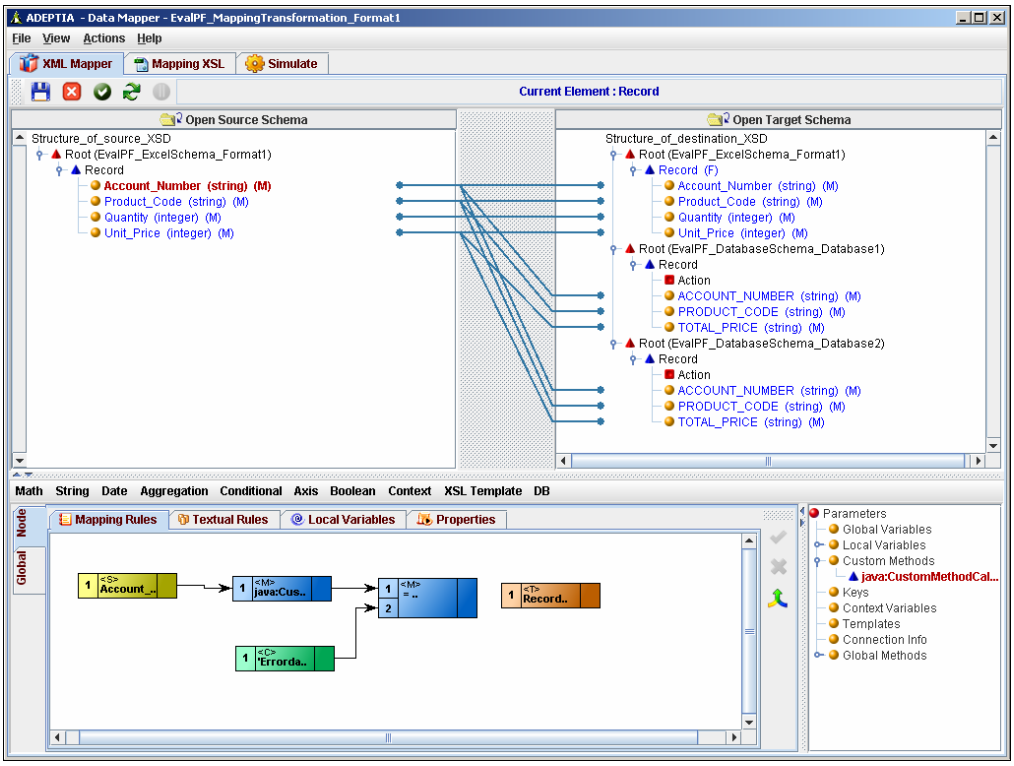



Figure 6.29: Create Links

41. Create another constant node with the value ('*True*').

| | |
|---|---|
|  | You need to add quotes while adding the value **True**. |

42. Click the **Conditional** mapping function and select **IF CONDITION > For Filtering Records** sub-function. The *IFF Condition* node is displayed in the Mapping Graph Area (see Figure 6.30).

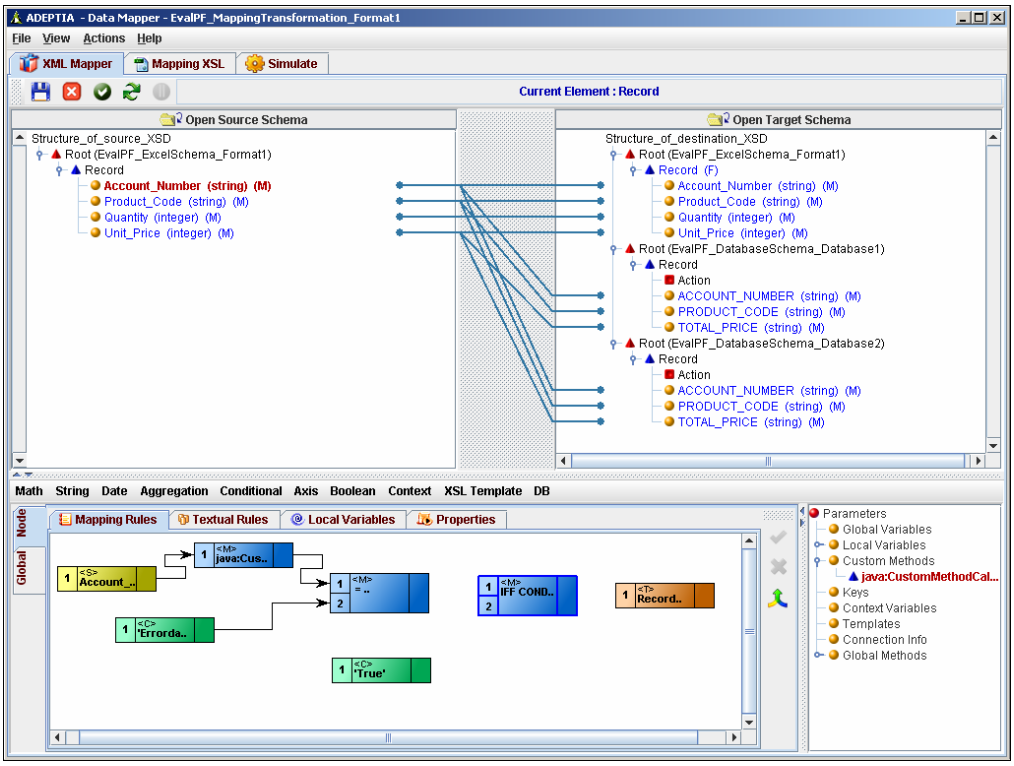



Figure 6.30: Select IF Condition

43. Create a link from the output of the *Equal* function node to the first input of the *IFF Condition* node.

44. Create a link from the output of the *Constant* node (‘*True*’) to the second input of the *IFF Condition* node.

45. Create a link from the output of the *IFF Condition* node to the input of the *Record* node.

46. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 6.31).

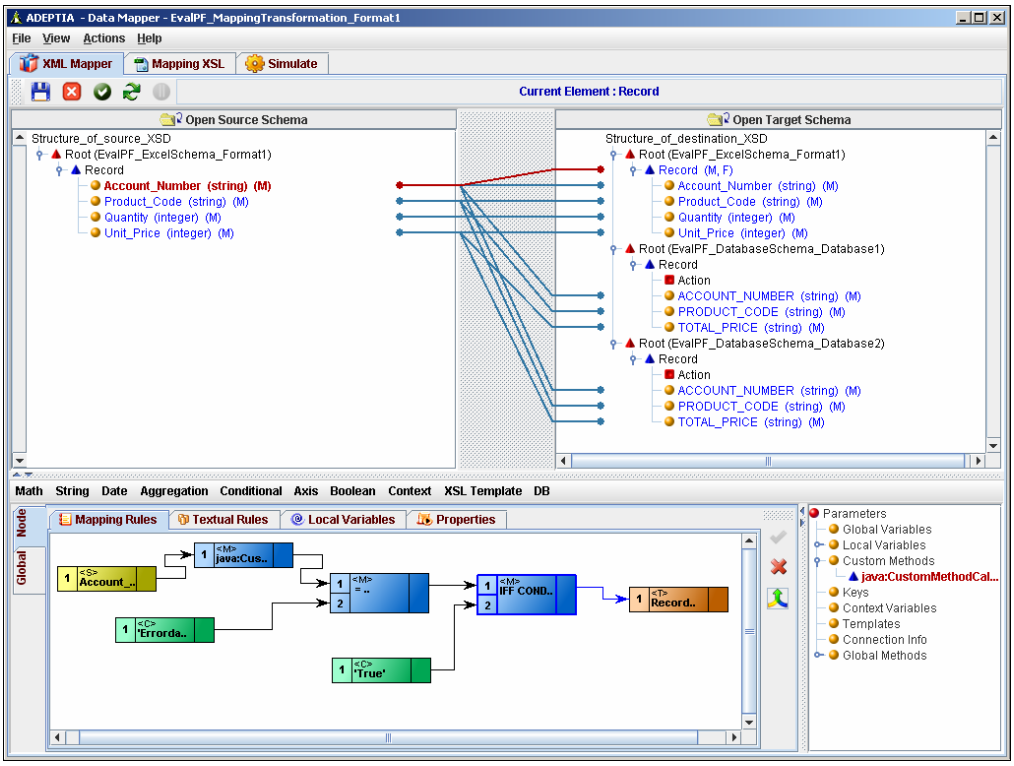



Figure 6.31: Apply Mapping

47. Similarly, to apply Custom Method on second schema (*EvalPF_DatabaseSchema_Database1*), select the **Record** node of the second schema and follow the steps 29 to 45.

| | |
|---|---|
| | The only difference is that you have to enter *NumericNumeric* as the constant value instead of *ErrorData.* |

48. To apply Custom Method on third schema (*EvalPF_DatabaseSchema_Database2*) select the **Record** node of the second schema and follow the steps 29 to 45.

| | |
|---|---|
| | The only difference is that you have to enter *AlphabetNumeric* as the constant value instead of *ErrorData.* |

49. Right-click the **Root** of the second target schema and select **Assign Stream** option. The **Assign Streams** dialog box is displayed (see Figure 6.32).

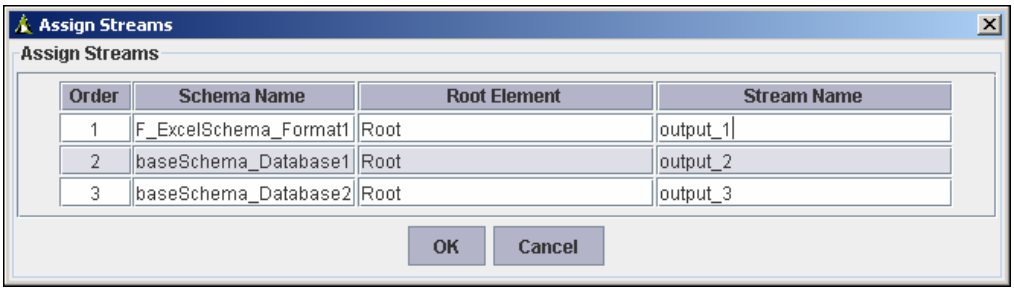



Figure 6.32: Assign Streams

50. Enter any stream name (*default*) for the Root having Occurrence Order *1*.
51. Enter any stream name (*database1*) for the Root having Occurrence Order *2*.
52. Enter any stream name (*database2*) for the Root having Occurrence Order *3* (see Figure 6.33).

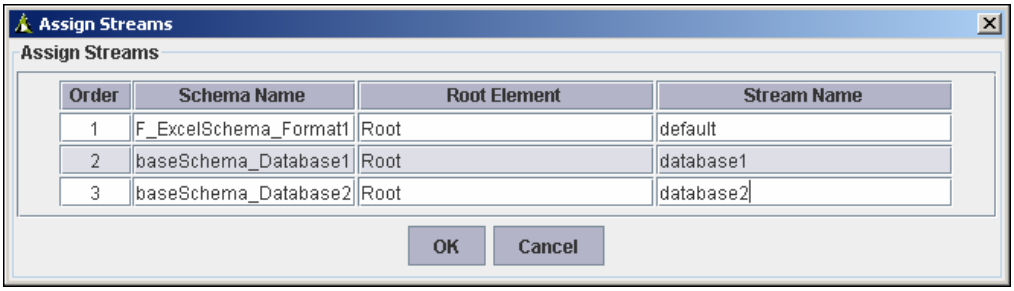



Figure 6.33: Enter Stream Name

| | |
|---|---|
|  | The stream for Root having Occurrence Order 1 is not activated in the Assign Stream dialog box because it has default stream assigned to it. |

53. Click on **OK** to close the Assign Stream dialog box.
54. Save the mapping by clicking the **File** menu and selecting **Save**. A dialog box is displayed confirming that the mapping activity has been saved successfully.
55. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping (refer to Figure 7.18).
56. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

57. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.
58. Exit the Data Mapper applet by clicking the **File** menu and selecting **Exit**.

# CREATING PROCESS FLOW

## (EvalPF_ProcessFlow_SalesReport)

Process Flow is the set of activities arranged in a sequence to perform a specific task(s). Process Flow is created by combining various activities i.e. Source, Target, Schema or Transformer etc. Process Designer is used to create Process Flow. Process Designer has list of activities created. You only need to arrange them in a logical sequence and connect them with BPMN Flows.

***Steps to create this sample Process Flow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (Refer to Figure 7.27).
3. Click the **New Process Flow** button. The Create Process Flow screen is displayed (Refer to Figure 7.28).
4. Enter the name and the description of the new Process Flow in the *Name* and *Description* fields respectively.
5. Select the logging level from the *Logging Level* drop-down list. There are four levels of logging. These are described in Table 7.9.
6. Select repository file retention from the *Repository File Retention* option. During execution Process Flow creates a temporary repository file to store the intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand sometime these repository files can be helpful in case of failure of Process Flow execution. . For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files and target formatted data. By default repository files are being stored in the repository folder of the Adeptia Suite. You can also choose an option to delete them or to archive them in a different location. There are four options for the Repository File Retention. These are described in Table 7.10.
7. Click the **Process Designer** button to open Process Designer. The Process Designer Screen is displayed (Refer to Figure 7.29).
8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

| | |
|---|---|
|  | Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar. |

9. Click **[+]Activities** in Repository View, to expand the list of services and then click **[+] Source**. All the items in the **Source** category are displayed.

10. Click **[+] Mail Source** under **Source** category. A list of existing Mail Source activities is displayed.

11. Select **EvalPF_MailSource** and drag it to the Graph Canvas Area (see Figure 6.34).

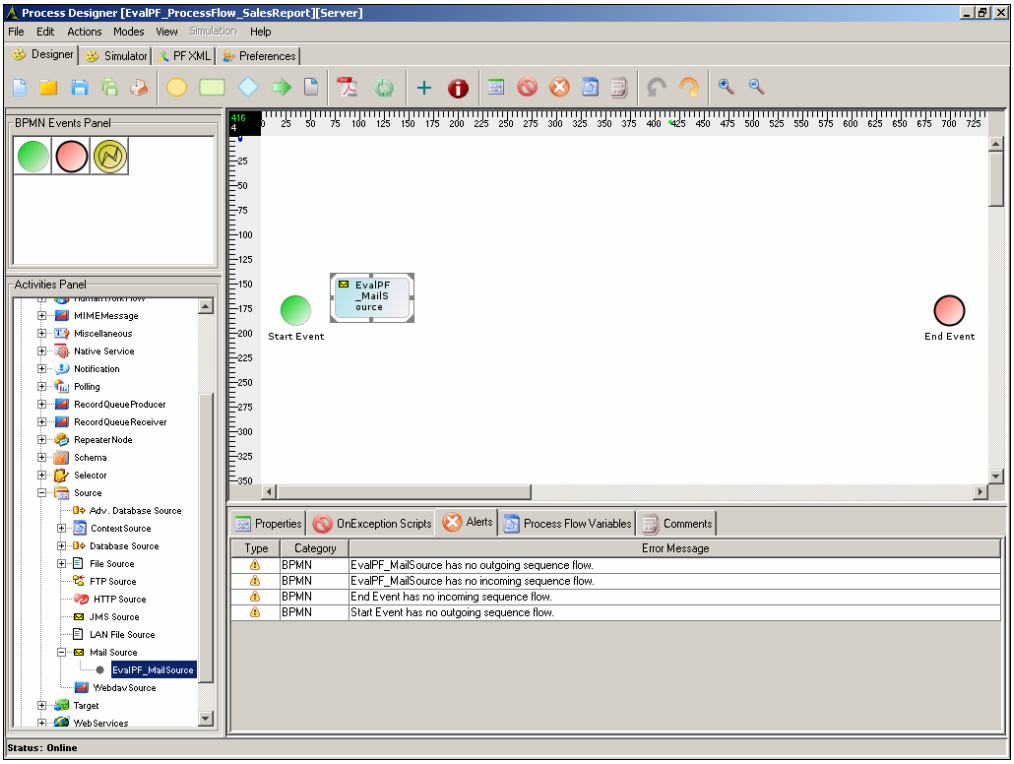



Figure 6.34: Drag File Source Activity to Graph Canvas Area

12. Right click on the *EvalPF_MailSource* activity and select *View Properties.* Properties if the *EvalPF_MailSource* activity are shown in the *Properties* panel of of the Bottom Pane.

13. Change the value of *eventContextEnabled* property from *false* to *true*.

14. To select a BPMN Gateway, click the **Gateway** (  ) icon in the Palette and drag it to the Graph Canvas Area (see Figure 6.35). In this Process Flow, **Gateway** is used to check the subject of the incoming mail and to decide which schema is to be used.

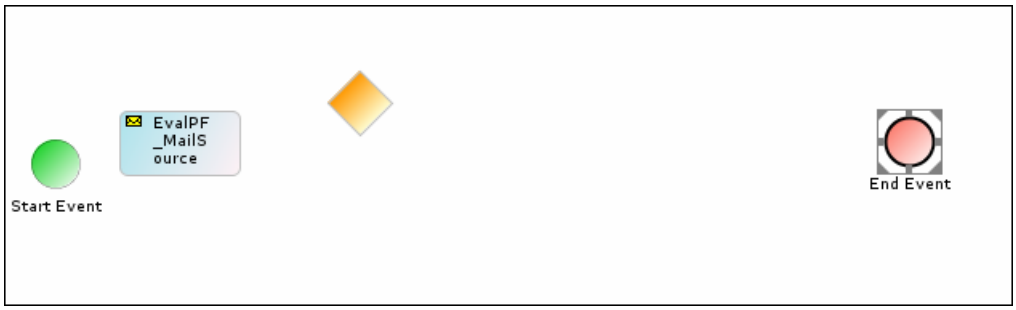



Figure 6.35: Drag BPMN Gateway Element to Graph Canvas Area

15. Click **[+] Schema** in the Activities Panel, and then click **[+] Excel Schema**. Select **EvalPF_ExcelSchema_Format1** activity and drag it to the Graph Canvas Area.
16. Similarly, drag **EvalPF_ExcelSchema_Format2** activity to the Graph Canvas Area.
17. Similarly, click **[+] DataTransform** and then **[+] Data Mapping**. Select **EvalPF_MappingTransformation_Format1** activity and drag it to the Graph Canvas Area.
18. Select **EvalPF_MappingTransformation_Format2** activity and drag it to the Graph Canvas Area.
19. After these activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
20. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
21. To connect *Start Event* with *EvalPF_MailSource*, drag mouse pointer from *Start Event* to *EvalPF_MailSource* (see Figure 6.36).

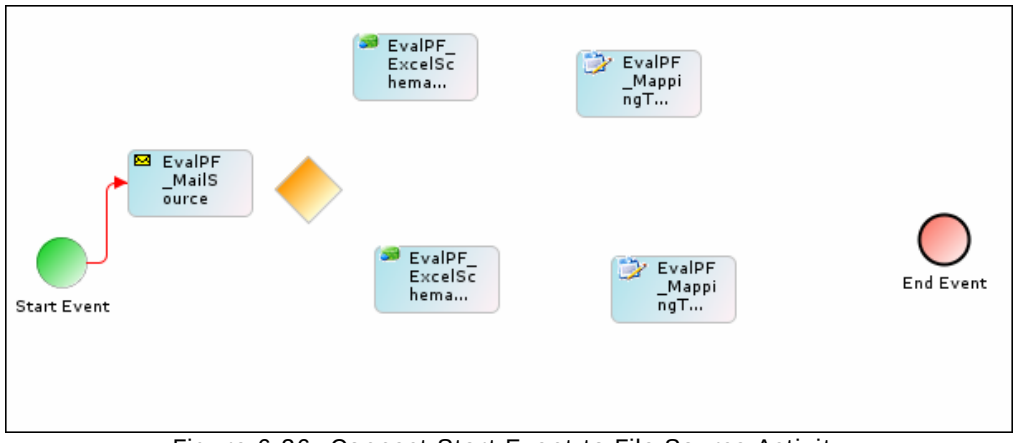



Figure 6.36: Connect Start Event to File Source Activity

22. Similarly, connect all other activities (see Figure 6.37).

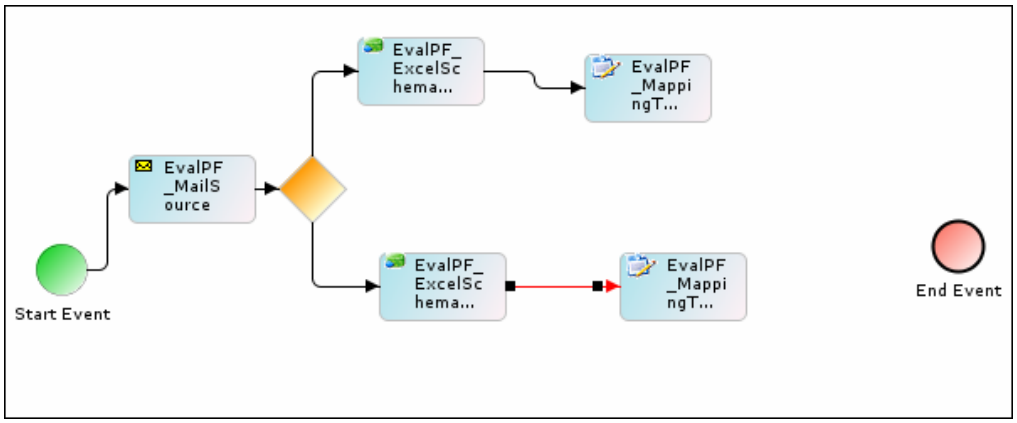



Figure 6.37: Connect all Activities

23. After connecting these activities, you need to define decision criteria on the BPMN Gateway. To define the decision criteria, right-click the BPMN Flow which is connecting *Gateway* and *EvalPF_ExcelSchema_Format1* activity and select **View Properties** (see Figure 6.38).

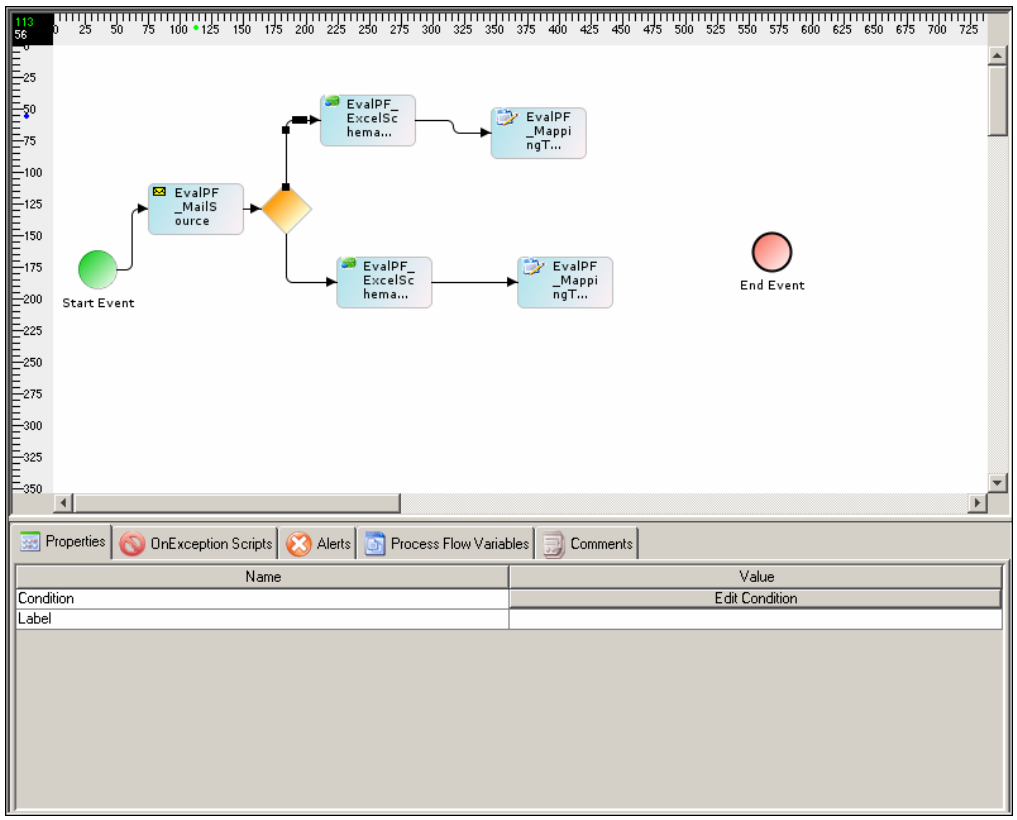



Figure 6.38: Define Decision Criteria

24. Click **Edit Condition** in the Properties Panel. The Condition Wizard is displayed (Refer to Figure 8.30).

25. Select **Process Flow Variable Condition** and click the **Next** button. The Process Flow Variable Condition type screen is displayed (see Figure 6.39).

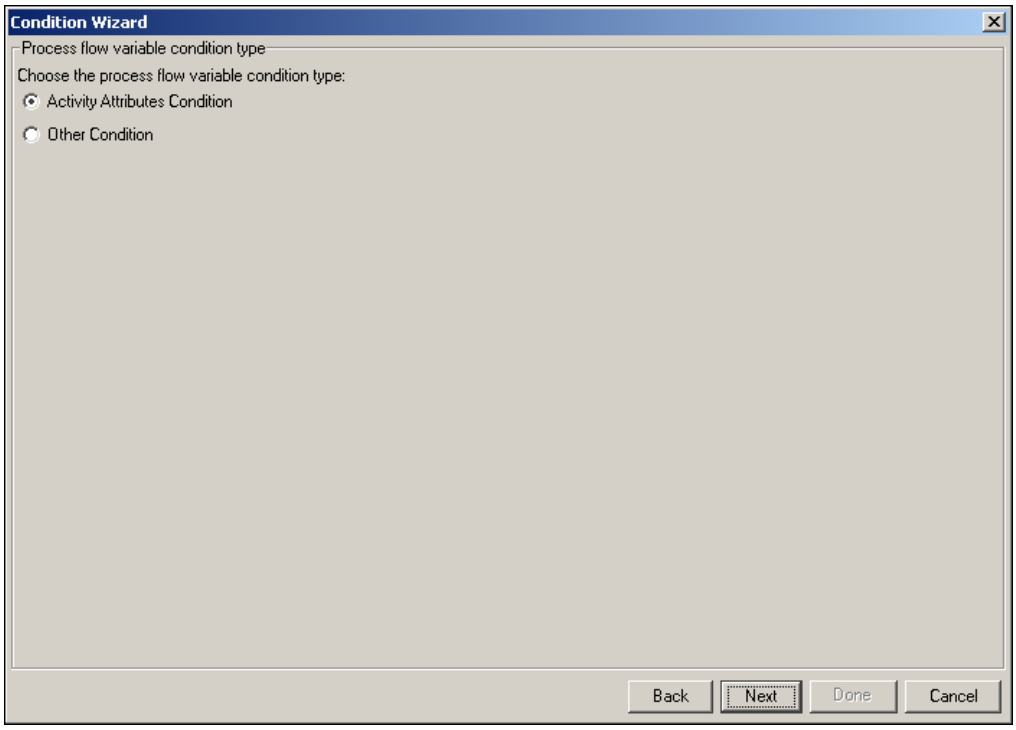



Figure 6.39: Process Flow Variable Condition Type

26. Select **Activity Attributes Condition** and click the **Next** button. The Activity Attribute Condition screen is displayed (see Figure 6.40).

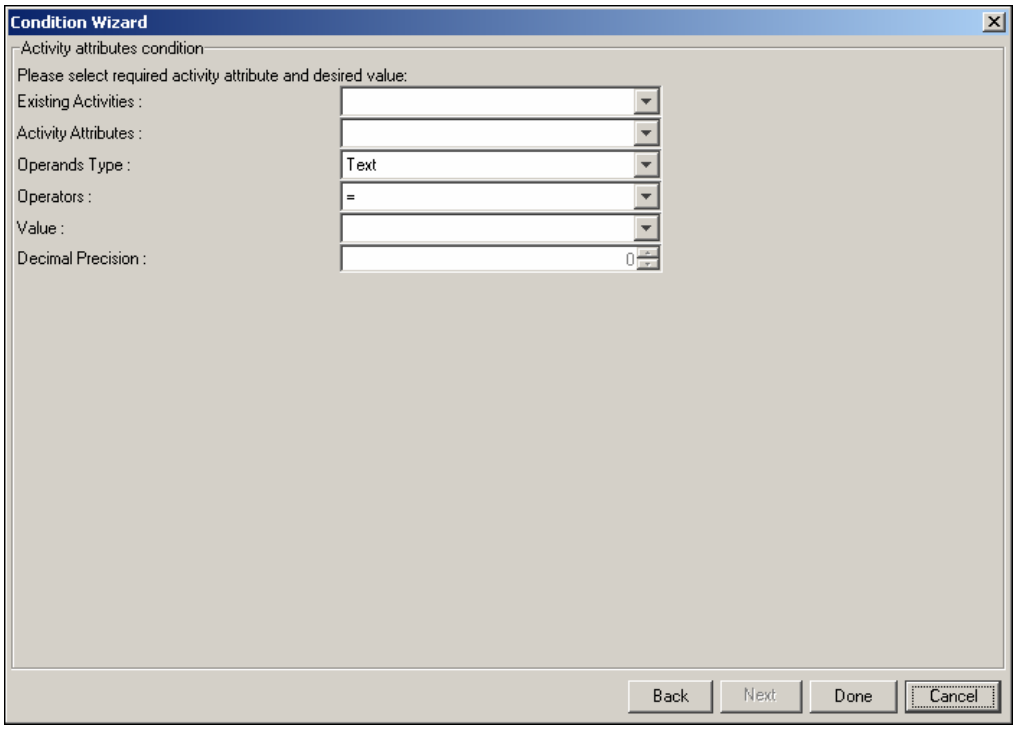



Figure 6.40: Select Activity Attributes

27. Select **EvalPF_MailSource** from the *Existing Activities* drop-down list, and select **mailSubject** from *Activity Attributes* drop-down list.
28. Select **Text** from the *Operand Type* drop-down list, and select = from the *Operators* drop-down list.
29. Enter *Format1* in the *Value* field (see Figure 6.41).

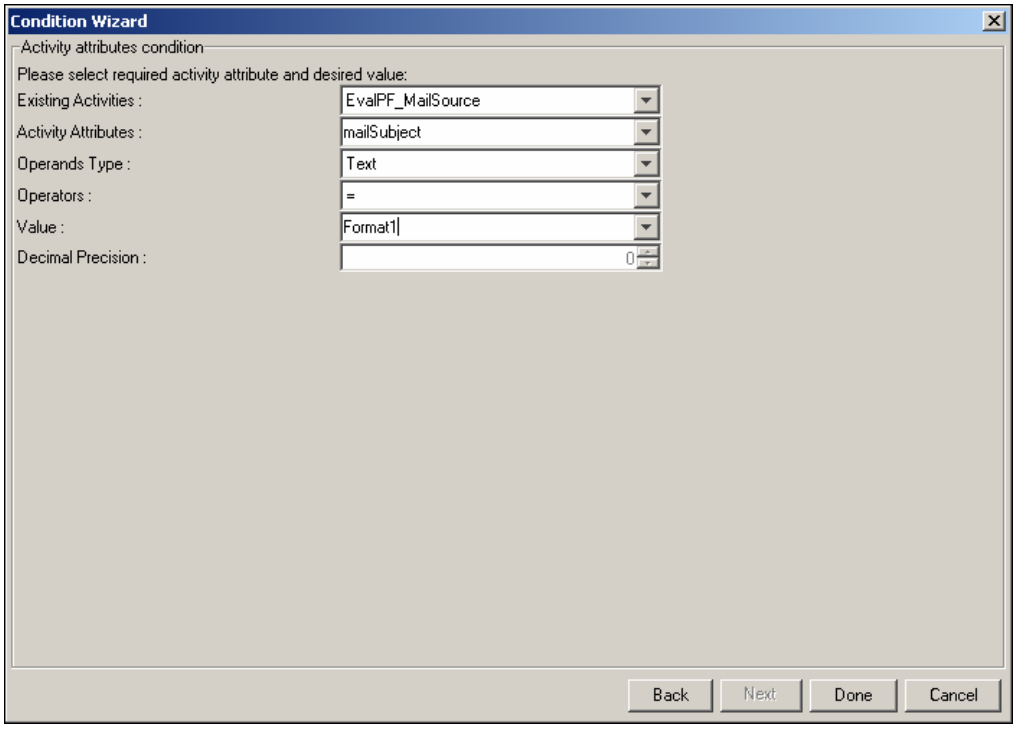



Figure 6.41: Select Attribute Value for *Format1*

30. Click the **Done** button to close the **Condition Wizard**.
31. Similarly, right-click the **BPMN Flow**, which is connecting *Gateway* and *EvalPF_ExcelSchema_Format2* activity and select **View Properties.**
32. Repeat the steps 26 to 30 and enter *Format2* in *Value* field (see Figure 6.42).

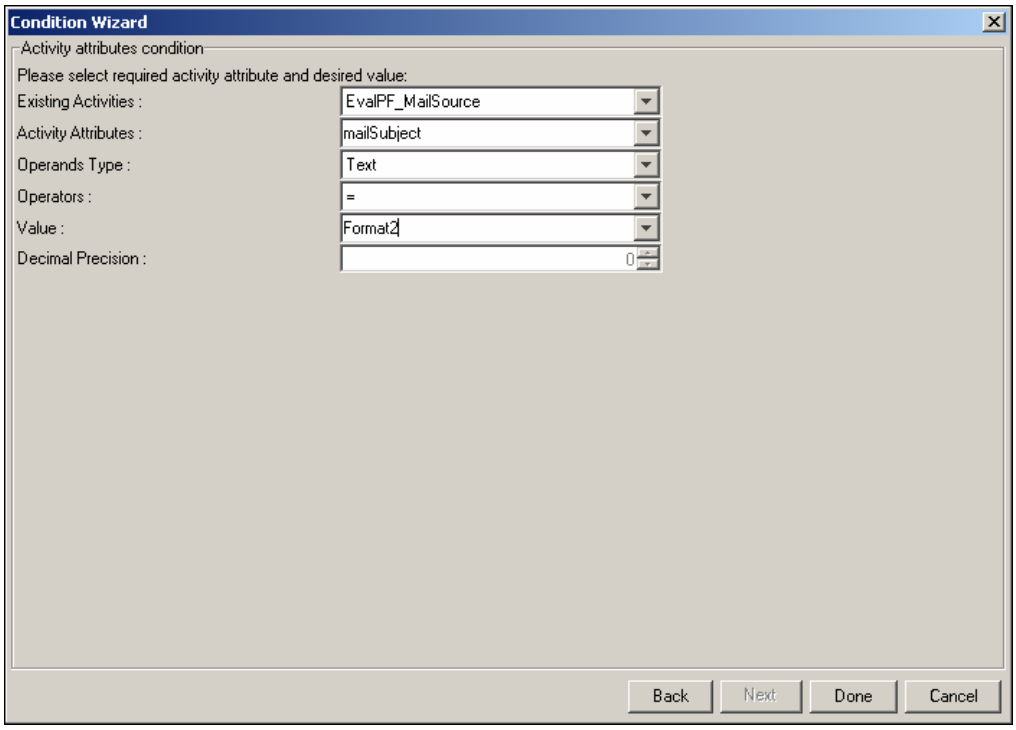



Figure 6.42: Select Activity Attributes for Format2

33. Click the **Done** button to close the Condition Wizard.

34. Click **[+] Target** and then **[+] File Target**. Select **EvalPF_FileTarget_ErrorRecord** activity and drag it to the Graph Canvas Area.

35. Click **[+] Action** in the Repository View to expand the list of Actions.

| | |
|---|---|
|  | An Action name is always unique. |

36. Select **Put-Context-Var** and drag it to the Graph Canvas Area and change its *label* to **Seq1_File_Path** (see Figure 6.43).

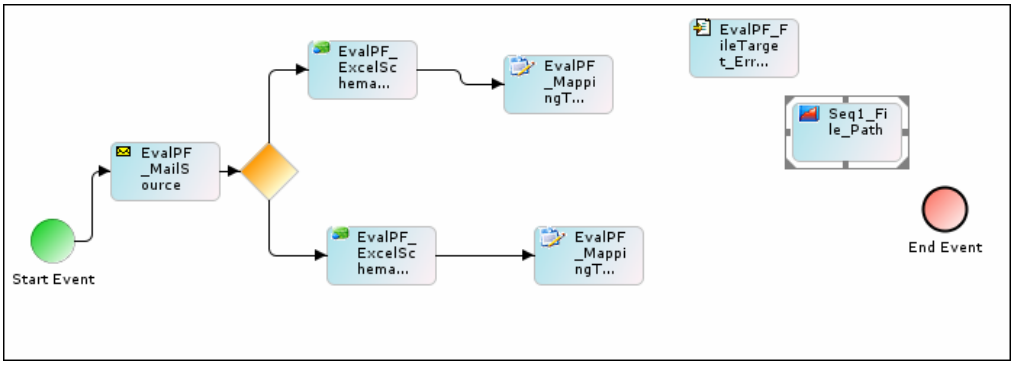



Figure 6.43: Drag *Put-Context-Var* to Graph Canvas Area

37. Connect **EvalPF_MappingTarnsformation_Format1** with **Seq1_File_Path** and then **Seq1_File_Path** with **EvalPF_FileTargetErrorRecord** (see Figure 6.44).

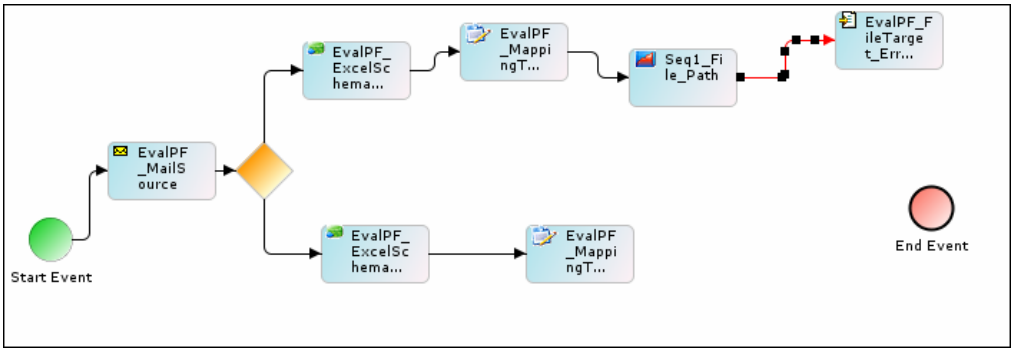



Figure 6.44: Connect Activities

38. In Graph Canvas Area, right-click **Seq1_File_Path** and select **View Properties**. Properties of *Seq1_File_Path* are shown in the Properties Panel (see Figure 6.45).

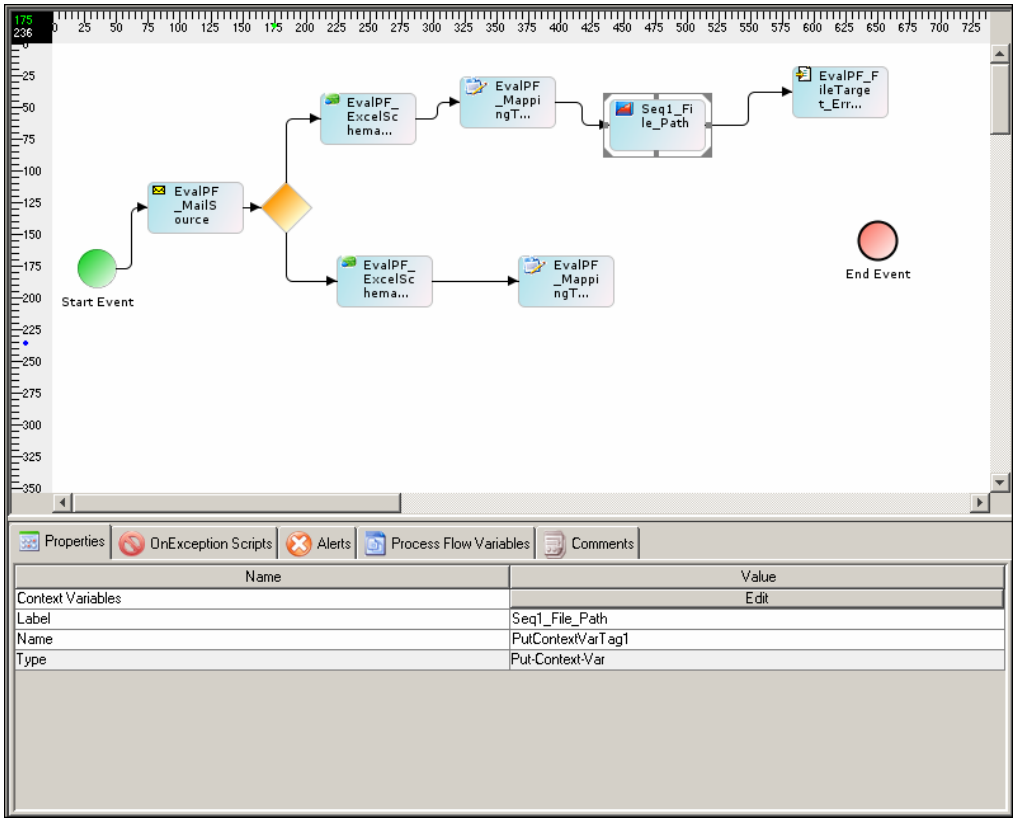



Figure 6.45: View Properties of *Seq_File_Path*

39. Click *Edit* from the *value* column of the *Context Variable* property. The *Edit Context Variables* screnn is displayed (see Figure 6.46 ).

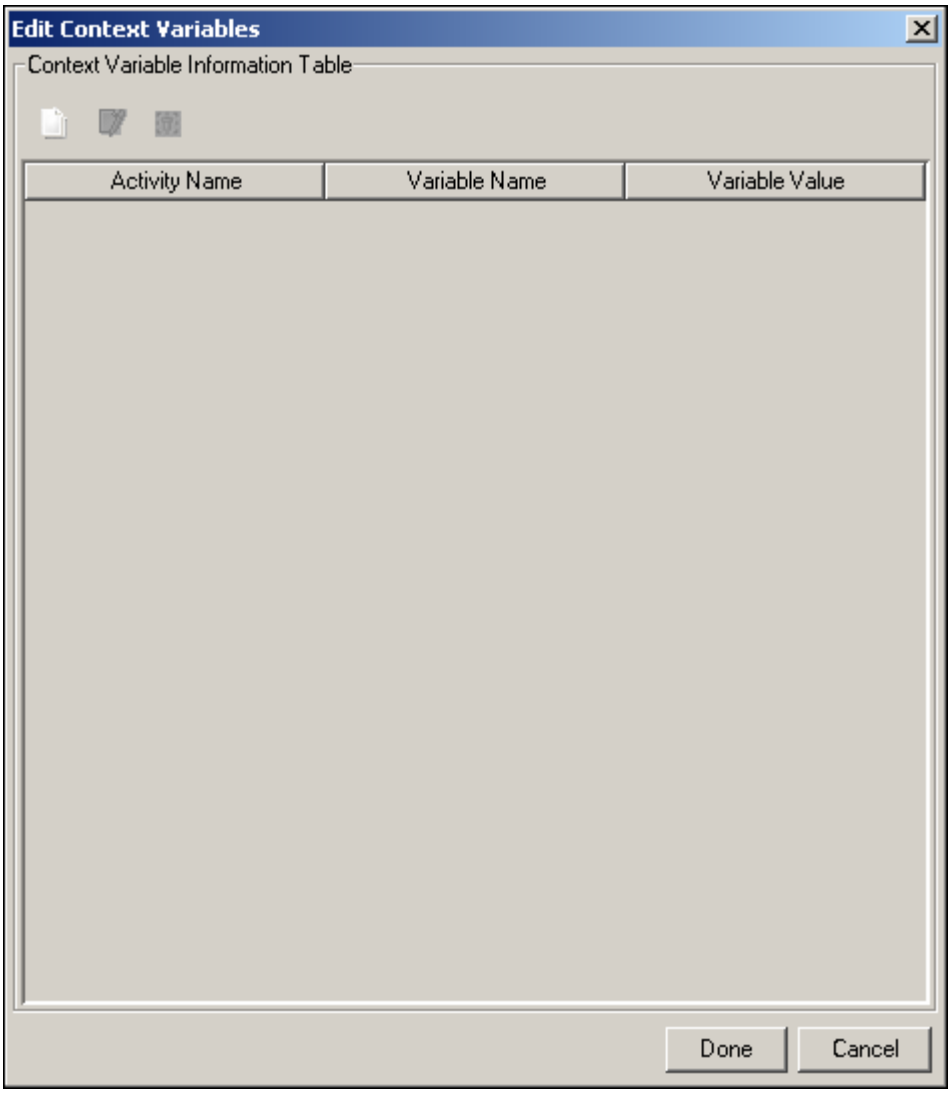



Figure 6.46: Edit Context Variable

40. Click *New Context Variable* ( ) button. The *Context Variable Information* dialog box is displayed (see Figure 6.47 ).

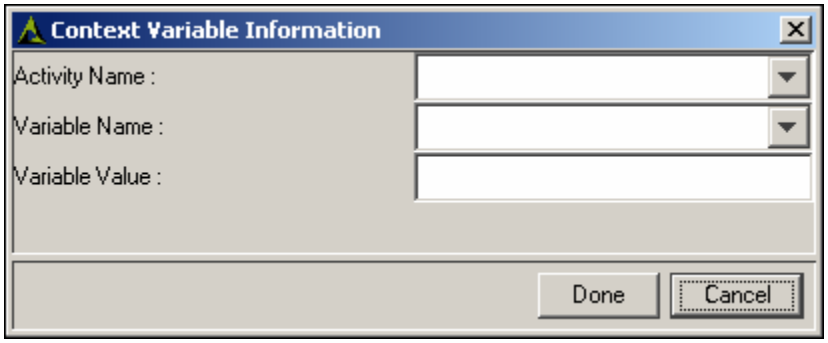



Figure 6.47: Add Context Variable

41. Select *EvalPF_FileTarget_ErrorRecord* from the *Activity Name* drop-down list.
42. Select *filepath* from the *Variable Name* drop-down list.
43. Enter the path of the target file with date and time format in the *Variable Value* field.

    For example *../../Solutions/Demo/EvalPF/ErrorRecord -%%yyyy-mm-dd%% %%hh-mm-ss%%.txt*

44. If the target file is saved on 2005-02-05 at 06-30-35, name of the file will be *ErrorRecords-2005-02-05-06-30-35.txt*
45. Click **[+] Target** and then **[+] Database Target**. Select **EvalPF_DatabaseTarget_Database1** activity and drag it to the Graph Canvas Area.
46. Similarly, drag **EvalPF_DatabaseTarget_Database2** activity to the Graph Canvas Area.
47. Connect these activities as shown in Figure 6.48.

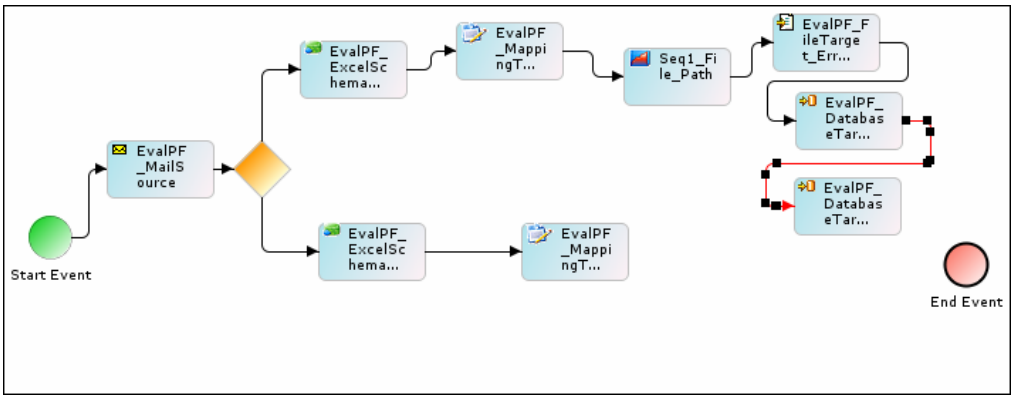



Figure 6.48: Connect Activities

48. Click **[+] Target** and then **[+] File Target**. Select **EvalPF_FileTarget_InsertError** activity and drag it to the Graph Canvas Area.
49. Similarly, drag **EvalPF_FileTarget_UpdateError** activity to the Graph Canvas Area.
50. Click **Action** and drag the **Put-Context-Var** action in the Graph Canvas Area (see Figure 6.49).
51. Right-click it and change the properties as shown in the table below.

Table 6.13: Changed Properties

| Properties | Value |
|---|---|
| Label | Seq2_File_Path |
| Activity | EvalPF_FileTarget_InsertError |
| Variable Name | filePath |
| Variable Value | ../../Solutions/Demo/EvalPF/InsertError -%%yyyy-mm-dd%% %%hh-mm-ss%%.txt |

52. From the Activities Panel, again drag **the Put-Context-Var** action in the Graph Canvas Area. Right-click it and change the properties as shown in the table below.

Table 6.14: Changed Properties

| Properties | Value |
|---|---|
| Label | Seq3_File_Path |
| Activity | EvalPF_FileTarget_UpdateError |
| Variable Name | filePath |
| Variable Value | `../../Solutions/Demo/EvalPF/UpdateError - %%yyyy-mm-dd%% %%hh-mm-ss%%.txt` |

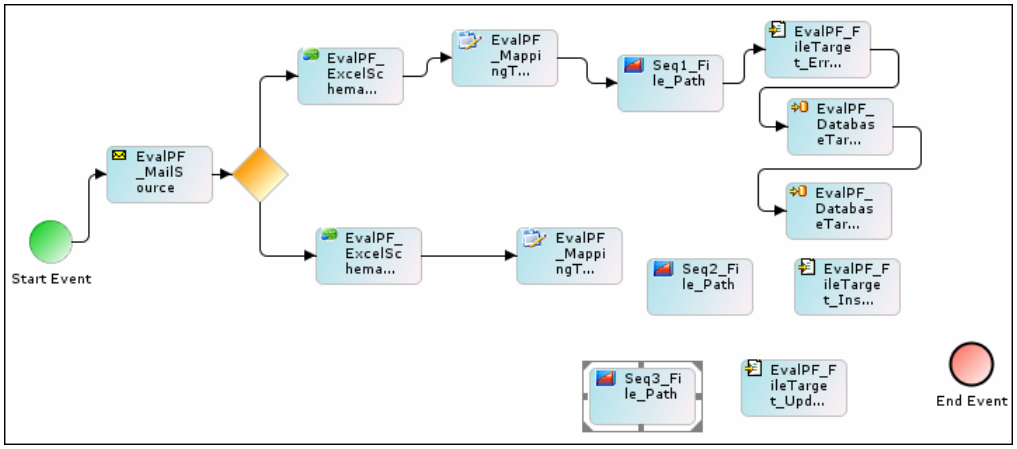



Figure 6.49: Drag *Put-Context-Var* to Graph Canvas Area

53. Connect these activities as shown in the Figure 6.50.

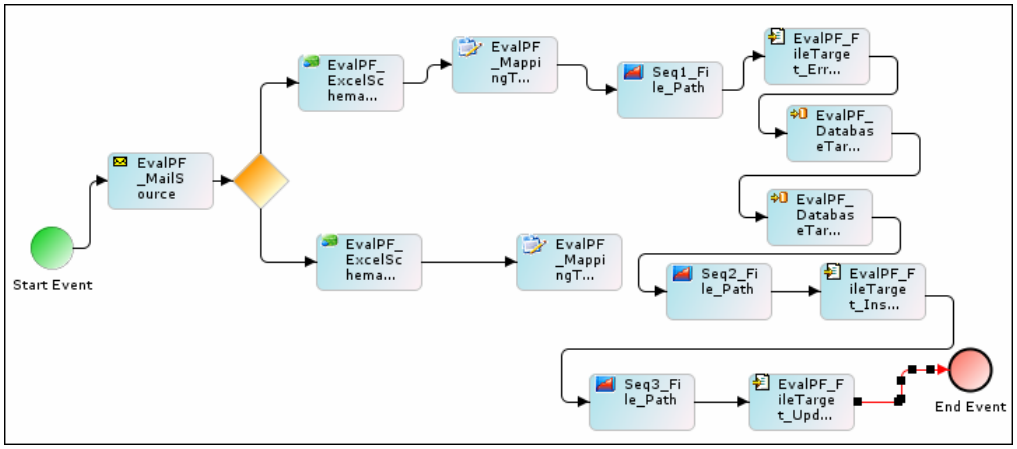



Figure 6.50: Connect Activities

54. Repeat the steps 36 to 55 to make another sequence of activities from **EvalPF_MappingTarnsformation_Format2** (see Figure 6.51).

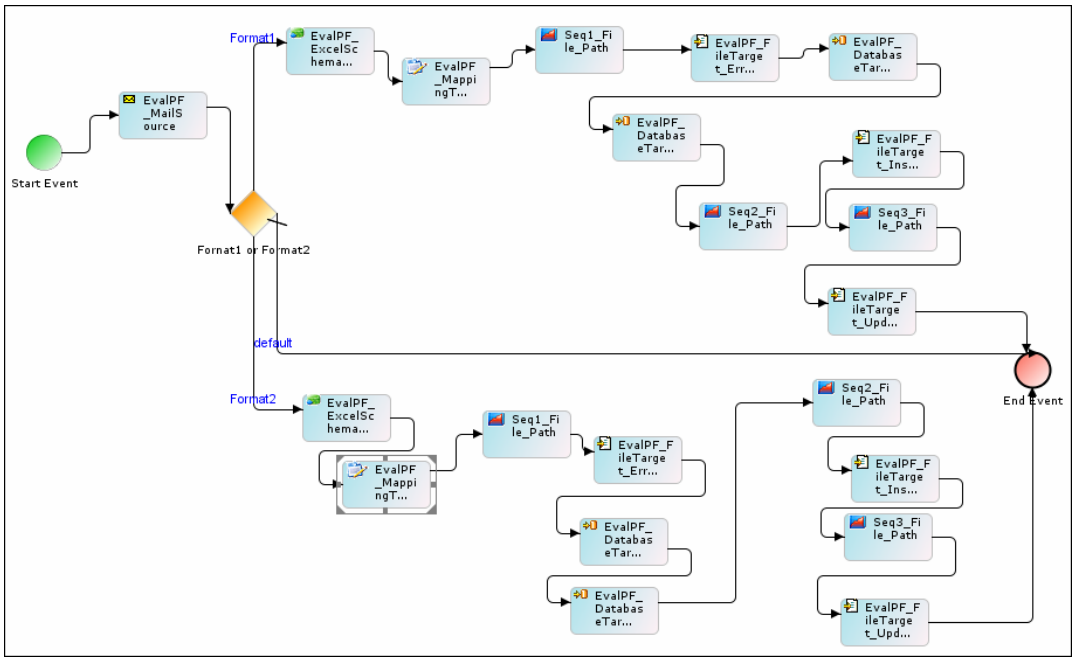



Figure 6.51: Connect Activities

- Since Process Designer does not allow more that one activity with the same name in one Process Flow, Error message is shown in the Error Panel of the bottom. To avoid this situation you can Save As those activities (being used more than once) with some different name. To save as these activities with different name refer to the Editing Activities section.

- A default outgoing sequence flow is added from the Gateway to End Event. While executing a process flow, if none of the specified conditions are met, then the End Event is executed. If there is no default outgoing flow specified, and none of the specified conditions are met, then an error occurs at the Process Flow execution.

- Figure 6.51 show only the control flow of the Process Flow. Control Flows only shows the order in which activities of a Process Flow is executed. It does not show the flow of data. For example as you can see in the figure 9.53, control flow from EvalPF_MailSource goes to Decision Node and the EvalPF_ExcelSchema_Format1 or EvalPF_ExcelSchema_Format2. But data is not passed from EvalPF_MailSource to Decision Node. Data is directly passed from EvalPF_MailSource to either of EvalPF_ExcelSchema_Format1 or EvalPF_ExcelSchema_Format2 based on the decision taken by Decision Node. To create the data flow, you need to create multiple streams.

55. To create multiple streams, right-click the **EvalPF_MailSource** activity and select **Multiple Stream**. The Multiple Stream dialog box is displayed (Refer to Figure 4.22).

56. Enter any name for the Data Stream (default) in the *Steam Name* field and click **Add Stream**.
57. Ensure that **EvalPF_ExcelSchema_Format1** is selected in the *Activities* drop-down list.
58. Click the **Map** button. A stream between **EvalPF_MailSource** and **EvalPF_ExcelSchema_Format1** is created.
59. Make sure the *Explicit Stream* checkbox is checked. A stream between **EvalPF-_MailSource** and **EvalPF_ExcelSchema_Format1** is created.
60. To create another stream, click the *Activities* drop-down list and select **EvalPF_ExcelSchema_Format2** and then click the **Map** button. A dialog box is displayed (Refer to Figure 5.23).
61. Click the **Yes** button. Another stream between **EvalPF_MailSource** and **EvalPF_ExcelSchema_Format2** is created.
62. Click the **Done** button to close the Multiple Stream Dialog box. Streams created are shown in the Graph Canvas area (see Figure 6.52).

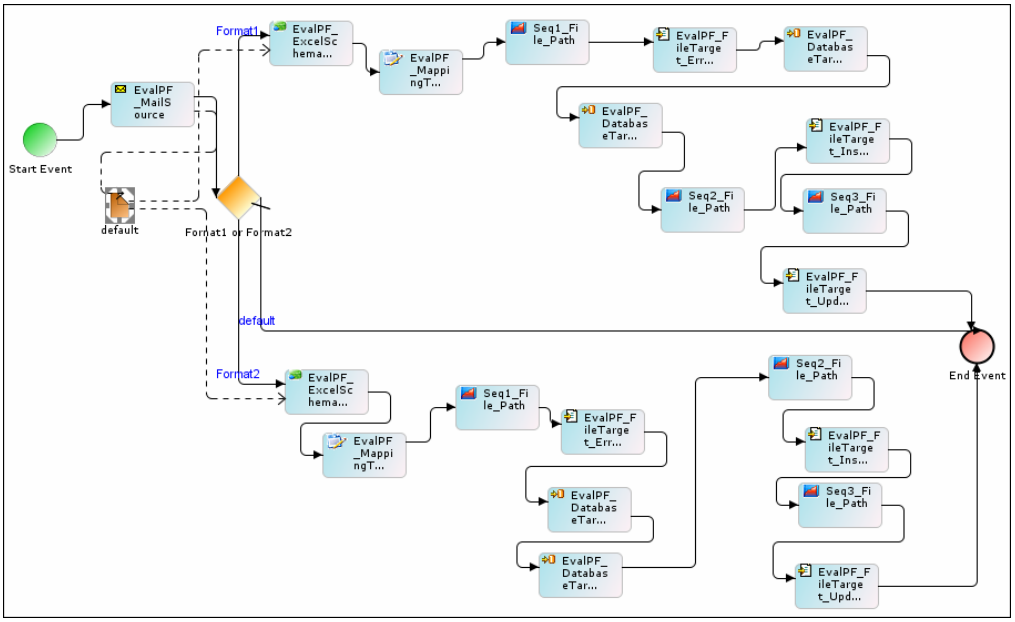



Figure 6.52: Multiple Streams Created

63. Repeat steps 57 to 64 to create another streams between activities listed below (see Figure 6.53):

    - From EvalPF_Mapping_Format1 to EvalPF_FileTarget_ErrorRecord, EvalPF_DatabaseTarget_Database1 and EvalPF_DatabaseTarget_Database2
    - From EvalPF_Mapping_Format2 to EvalPF_FileTarget_ErrorRecord, EvalPF_DatabaseTarget_Database1 and EvalPF_DatabaseTarget_Database2
    - From EvalPF_DatabaseTarget_Database1 to EvalPF_FileTarget_InsertError

- From EvalPF_DatabaseTarget_Database2 to EvalPF_FileTarget_UpdateError

Streams between following activities must be selected as *Error Stream*.

- EvalPF_DatabaseTarget_Database1 and EvalPF_FileTarget_InsertError
- EvalPF_DatabaseTarget_Database2 and EvalPF_FileTarget_UpdateError

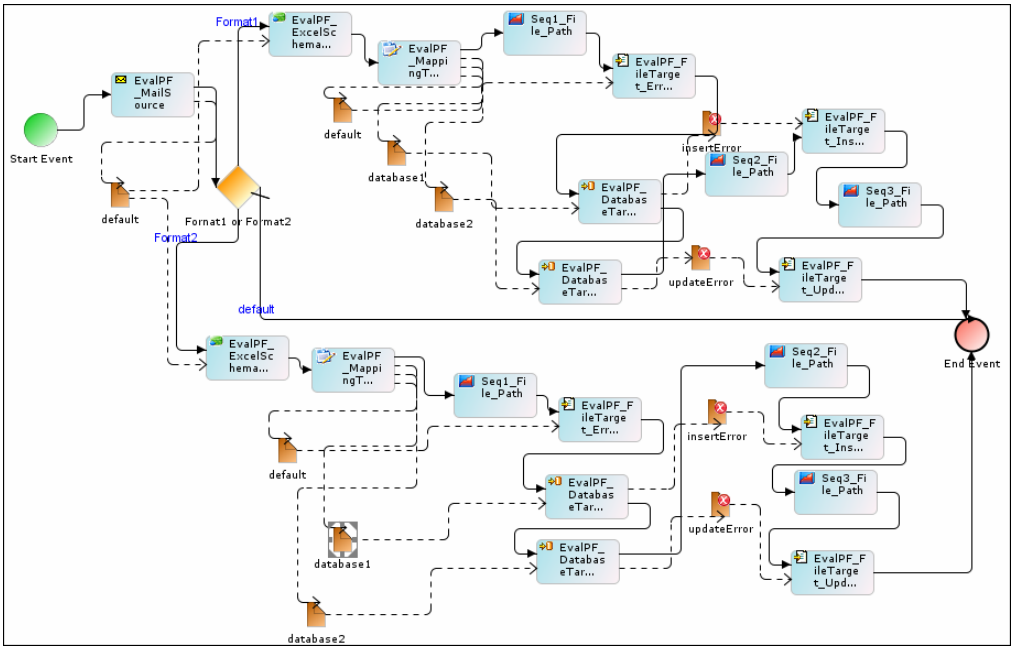



Figure 6.53: Multiple Streams Created for Multiple Activities

64. Save the Process Flow by clicking the **File** menu and selecting **Save Process Flow to Server**. A dialog box is displayed confirming that the *EvalPF_ProcessFlow_SalesReport* has been saved successfully.
65. Click the **Done** button to close the dialog box.
66. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

# REGISTERING PROCESS FLOW WITH MAIL EVENTS

After the Process Flow is created it must be registered with both of the Mail Events. Mails Event triggers the Process Flow when a mail with specified subject arrives on mail server. To register the Process Flow with the Mail Event, Event Registry activity is created. Event Registry *(EvalPF_EventRegistry_Format1* and *EvalPF_EventRegistry_Format2)* used to trigger *EvalJMSE_ProcessFlow_SalesReport* are already created. This section describes how to edit Event Registry.

***Steps to edit the Event Registry:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Event Registry**. The Manage Event Registry screen is displayed with the list of existing Event Registries (Refer to Figure 7.38).
3. Click **Edit** in the **Action** column of the *EvalPF_EventRegistry_Format1* activity. This displays the Edit *EvalPF_EventRegistry_Format1* activity screen, with the properties of the activity displayed in their respective fields (see Figure 6.54).

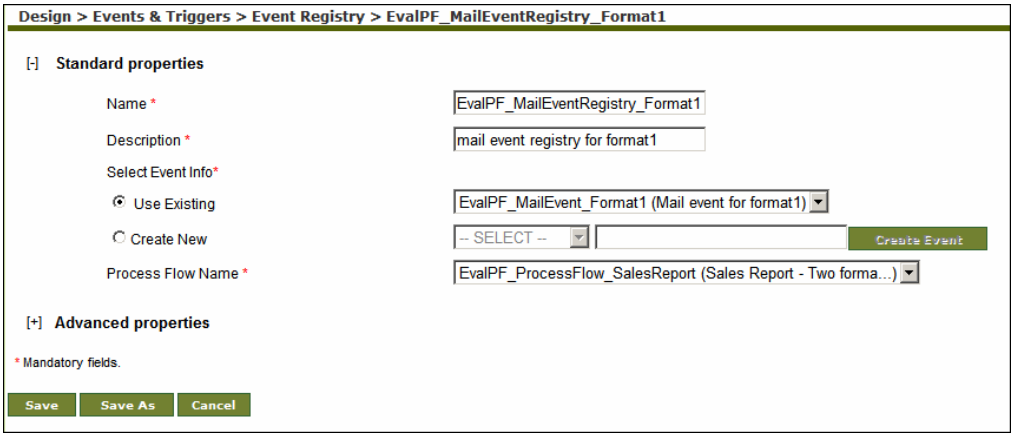



Figure 6.54: Edit *EvalPF_MailEventRegistry_Format1* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 6.15: Details of Fields on Edit Event Registry Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Event Registry |
| Description | Description of the Event Registry |
| Event Name | Name of the JMS Event, which triggers the Process Flow |
| Process Flow Name | Name of the Process Flow, which is triggered by JMS Event |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Event Registry has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the event registry (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

7. Click **OK** to save the comments. This displays a screen confirming that the event registry has been updated successfully.

Repeat the same steps to register the Process Flow with *EvalPF_EventRegistry_Format2*.

# 7 JMS EVENT DRIVEN PROCESS FLOW

This section describes the JMS Event Driven Process Flow.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This sample Process Flow demonstrates the use of a JMS Event for triggering a Process Flow. The JMS Event is configured to listen for the data (containing real-time stock quotes) from a JMS server. The JMS Event is registered with the Process Flow. JMS Event on receiving the data, gets fired, and in turn triggers the Process Flow. The Process Flow receives data from the JMS Event, converts it into database specific format and finally inserts the data into a database server.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- JMS Event to trigger a Process Flow.
- Context Source to receive data from the JMS Event.
- Text Schema to parse the data (coming from JMS Server) and to convert it into an intermediate XML format.
- Mapping to map the data fields of the text schema and the database schema.
- Database Schema to convert the data from intermediate XML format into the database specific format.
- Database Driver and Database Info.
- Database Target

## DESCRIPTION

This sample Process Flow consists of three components (see Figure 7.1):

- JMS Event to trigger the Process Flow
- Event Registry to register the Process Flow with the JMS Event

- Process Flow

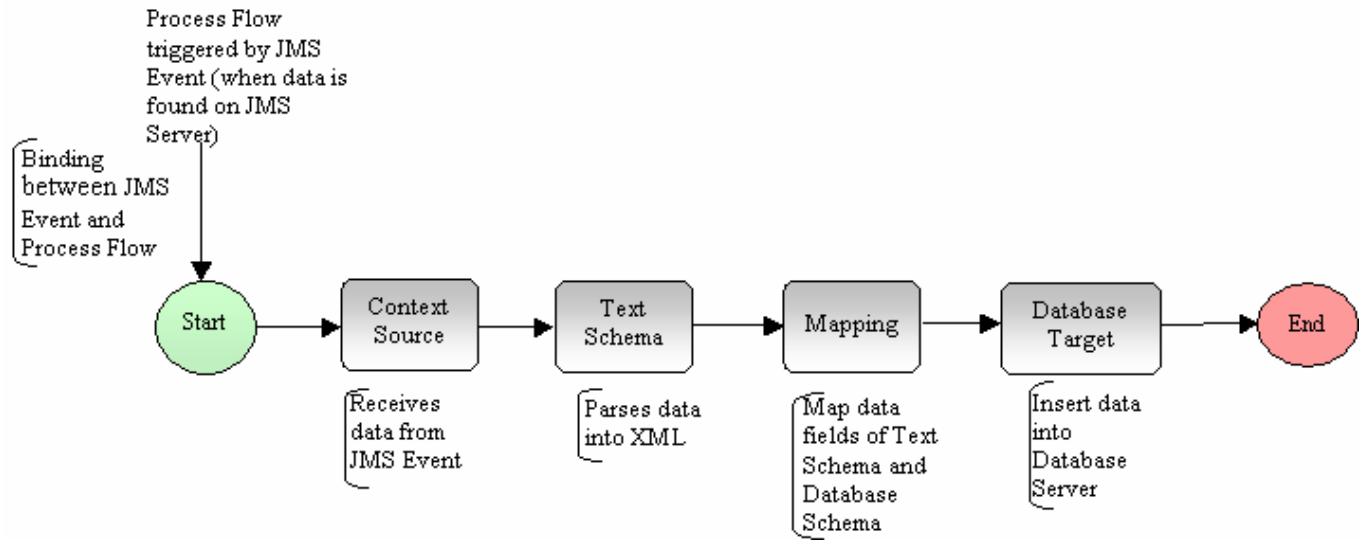



Figure 7.1: Flow Chart showing JMS Driven Process Flow

## JMS Event (EvalJMSE_JMSEvent)

**JMS event** is used to listen for the data on a JMS Queue. The moment JMS Event receives the data; it triggers the Process Flow and passes the data (received from the JMS server containing stock quotes) to context source. JMS Event uses JMS Provider (EvalJMSE_JMSProvider) to connect to JMS Server.

## Event Registry (EvalJMSE_EventRegistry)

**Event Registry** is used to register the JMS Event with the Process Flow. In other words, Event Registry is a link between the JMS Event and the Process Flow.

## Process Flow (EvalJMSE_ProcessFlow)

**Process Flow** receives data from the JMS Event and converts it into database specific format, which is stored in a database server. Process Flow does this conversion using following activities:

- **Context Source** (EvalJMSE_ContextSource)

JMS Event cannot pass the data to the Process Flow and, hence, cannot be used as a source of the Process Flow. The JMS Event sets the data to a Process Flow Variable. To convert the value of the Process Variable into data stream, which is to be consumed by Text Schema, context source is used.

- **Text Schema** (EvalJMSE_TextSchema)

Text Schema is used to parse the data received from the JMS Server and to convert it into an intermediate XML format.

- **Mapping** (EvalJMSE_Mapping)

Mapping is used to map the data fields of Text Schema to the data fields of the Database Schema. In this Process Flow simple one to one mapping is used.

- **Database Driver** (*SQLServerJTDSDriver*)

Database Driver is used to specify the type of database and to upload driver jar files required to connect to that database.

- **Database Info** (EvalJMSE_DBInfo)

Database Info activity is used to specify the JDBC URL or Server URL and Username and Password to access the database. Database Info uses Database driver to connect to the specified Database Server.

- **Database Schema** (EvalJMSE_DBSchema)

At the target end, database Schema is used to convert data from intermediate XML format to database specific format. Database Schema uses Database Driver (*SQLServerJTDSDriver*) and Database Info (EvalJMSE_DBInfo) to connect to Database Server.

- **Database Target** (EvalJMSE_DBTarget)

Database target is used to specify the database server and name of the database, where the target data is to be stored. Database Target uses Database Schema (EvalJMSE_DbSchema).

## USAGE SCENARIO

This sample Process Flow can be used whenever you wish to automate the execution of a Process Flow through JMS Event.

## DATA DESCRIPTION

The Stock Quote Data contains real-time stock updates for several companies.

Data contains:

- One record per stock
- Records are separated by new line
- Fields under a record are separated by Comma (**,**)

The structure of the Stock Quote Data is displayed in the table below.

Table 7.1: Structure of Stock Quote Data

| Field Name | Description | Data Type |
|---|---|---|

| | | |
|---|---|---|
| Symbol | A unique symbol is assigned to a security | String |
| CompanyName | Name of the company | String |
| LastTradePrice | Last trade price | Number |
| TradeTime | The date and time on which a security trade occurs | Date |
| PreviousClosePrice | The final price at which a security is traded on a given trading time | Number |
| OpenPrice | Start of trading on the securities exchange | Number |
| OneYearTargetEst | The projected price level as stated by an investment analyst or advisor. | Number |
| AvgVolume | The number of shares traded per day, averaged over some time period, usually one year. | Number |
| MarketCap | Market value of a company | Number |

Database Tables also have same fields as shown in the Table 7.1.

**NOTE**

By default JMS Event is in the deactivated state. It must be activated before executing the Process Flow. When in activated state, JMS event continuously listens for data from JMS Server. JMS Event on receiving the data, is fired, and in turn triggers the Process Flow. After execution of the Process Flow, JMS Event again starts listening for new data.

## PREREQUISITES

- OpenJMS must be installed and running.

To know how to install and configure OpenJMS, refer to **Appendix A: Setting up OpenJMS.**

- Ensure that Queue is available in OpenJMS. *Queue1* is the default queue for OpenJMS.
- Table must be present in database server used as target.

To know, how to create table into your database, refer to **Creating Table into Database** section.

- Before executing this process flow, you must edit the following activities to point to the database which is used as source:
  - EvalXform_DBDriver
  - EvalXform_DBInfo

To know, how to edit these activities refer to **Editing Activities** section.

## CREATING TABLE INTO DATABASE

In this process flow, a database table is used as target. A SQL script is provided with Adeptia Suite to create a table in your database. This SQL script is located in ***../AdeptiaServer-5.0/Serverkernel/Solutions/Demo/EvalJMSE*** folder. To create table where data is to be populated into your target database, you need to execute the respective SQL Script, using the database client application.

## USING ANOTHER JMS SERVER

The sample Process Flow is configured with OpenJMS server. If another JMS Server is to be used, some activities must be changed. These activities are outlined as:

- EvalJMSE_JMSProvider
- EvalJMSE_JMSEvent

To know, how to edit these activities refer to the **Editing Activities** section.

## EXECUTING AND MONITORING

This section explains the execution and monitoring of sample Process Flow. Steps involved in execution of this sample Process Flow can be broadly divided as:

1. Activating JMS Event
2. Sending data to JMS Server
3. Monitoring Process Flow execution

### Activating JMS Event

By default, JMS Event is in *deactivated* state. It must be activated before executing the sample Process Flow.

***Steps to activate the JMS Event***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.

2. Click **[+] Events & Triggers** to expand the tree and then click **JMS**. The Manage JMS Event screen is displayed with the list of existing JMS Events (see Figure 7.2).

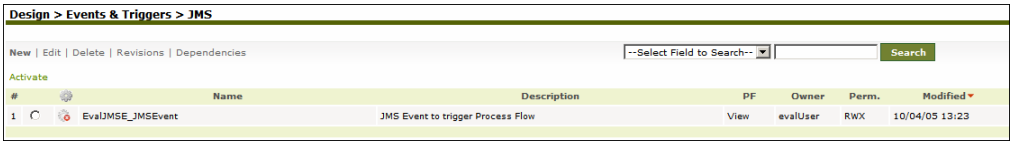



Figure 7.2: Manage JMS Event

3. To activate the JMS Event **(EvalJMSE_JMSEvent)**, select the radio button adjacent to *EvalJMSE_JMSEvent* and then click **Activate** link. A screen is displayed confirming that the JMS Event activity has been activated successfully.

## Sending Data to JMS Server

Once the JMS Event is activated, it starts listening for the data from JMS Server. To execute the sample Process Flow, you only need to send the data to the JMS Server. After the data is sent to the JMS Server, the JMS Event triggers the Process Flow.

The sample data file, ***EvalJMSE_StockQuotes.txt,*** is located in **<drive>/<base directory>/Solutions/Demo/EvalJMSE** directory. The batch file, ***EvalJMSE.bat***, is also saved in the same directory. This batch file is used to send the sample data file, ***EvalJMSE_StockQuotes.txt,*** to the JMS Server.

***Steps to send data to the JMS Server***

1. Enter the command, '*Cd ../../Solutions/Demo/EvalJMSE*' at the command prompt.
2. Execute the ***EvalJMSE_JMS.bat*** batch file. The JMS dialog box is displayed (see Figure 7.3).

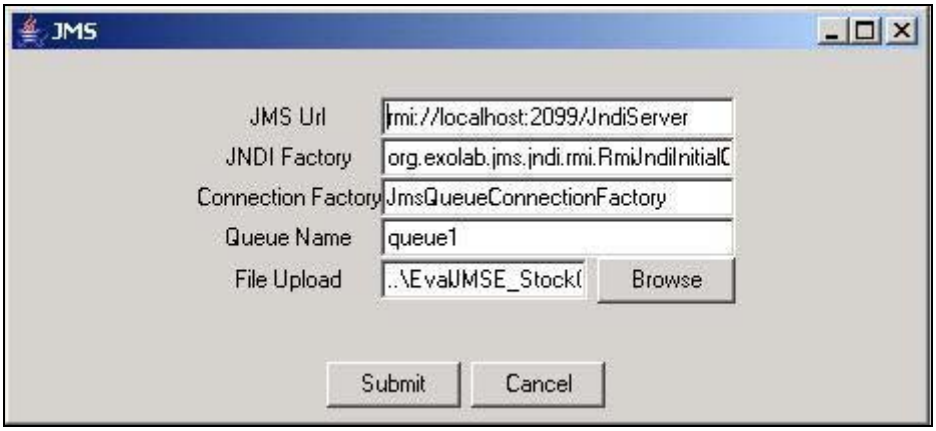



Figure 7.3: Send File to OpenJMS Server

3. Click **Submit** to send the data to the OpenJMS Server.

If you are using another JMS Server, change the required parameters and then click the **Submit** button.

Once the data is sent to the JMS Server, JMS Event triggers the Process Flow.

## Monitoring Process Flow Execution

During execution of the Process Flow you can view status of the Process Flow as well as status of each activity of the Process Flow.

***Steps to monitor Process Flow execution***

1. In the Adeptia Suite homepage menu, click **[+] History** to expand the tree. All the items in the **History** category are displayed.
2. Click **Process Flow Log**. The Process Flow Log screen is displayed (see Figure 7.4).

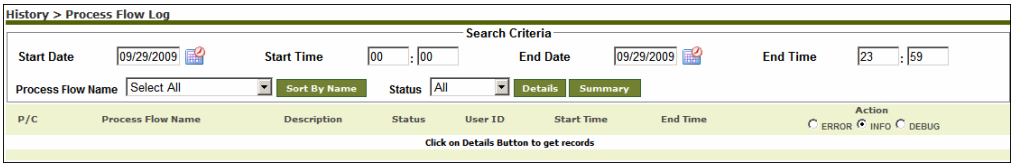



Figure 7.4: Process Flow Log

3. Select the Process Flow (*EvalJMSE_ProcessFlow*) from the *Select Process Flow Name* drop-down list and select *Executed* from the *Status* drop-down list.
4. Click the **Details** button. This displays the list of activities of the selected process flow conforming to the entered criteria (see Figure 7.5).

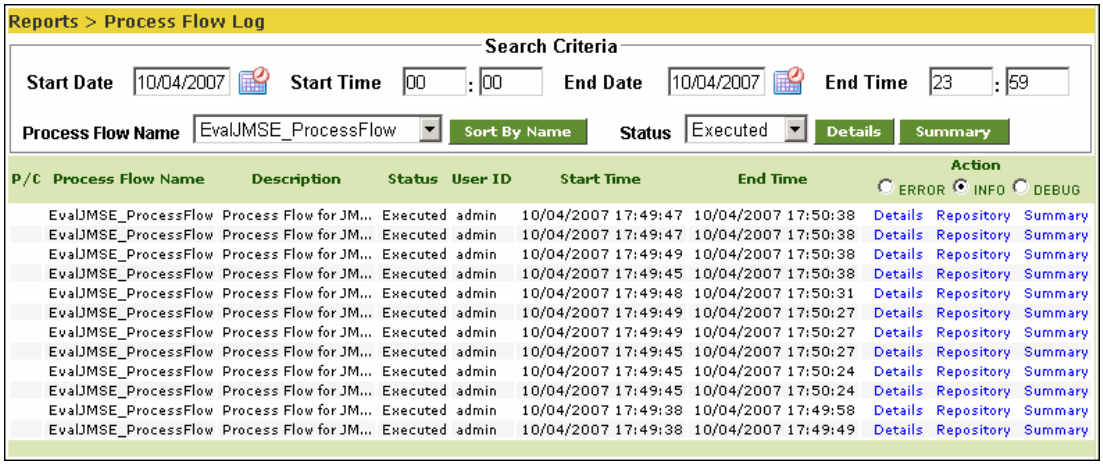



Figure 7.5: Searched Process Flows

5. Click **Details** in the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 7.6).

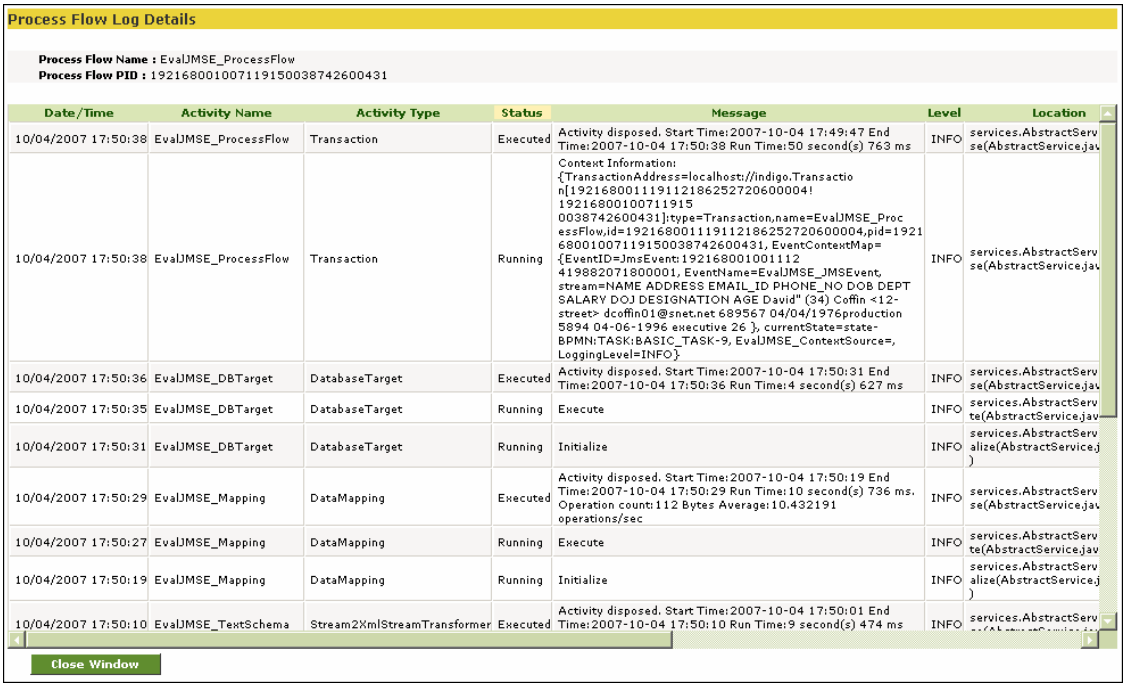



Figure 7.6: Process Flow Log Details

6. To view the summary of all instances of the process flow execution, click the **Summary** button. This displays the Process Flows Summary screen (see Figure 7.7).

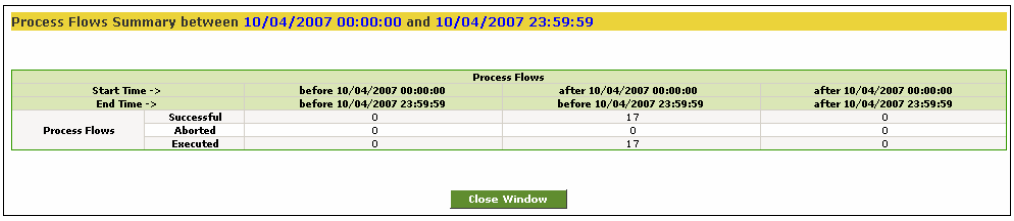



Figure 7.7: Process Flows Summary

# EDITING ACTIVITIES

Activities used in this sample Process Flow are pre-created. This section describes the process of editing these pre-created activities.

## Editing JMS Provider

## (EvalJMSE_JMSProvider)

JMS Provider is used to connect to JMS Server. While creating JMS Provider, you need to specify the Provider Jar files, which are used to connect to JMS server. JMS Event further uses JMS Provider. This should be noted that JMS Provider does not directly take part in the creation of Process Flow. It just helps the JMS Event to connect to the JMS Server.

***Steps to edit JMS Provider:***

1. Click [+] **Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **JMS Provider**. The Manage JMS Provider screen is displayed with the list of existing JMS Providers (see Figure 7.8).

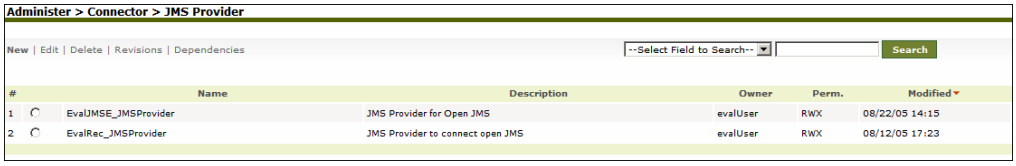



Figure 7.8: Manage JMS Provider

3. Select the radio button adjacent to *EvalJMSE_JMSProvider* activity and then click **Edit** link. This displays the Edit *EvalJMSE_JMSProvider* activity screen, with properties of the activity displayed in their respective fields (see Figure 7.9).

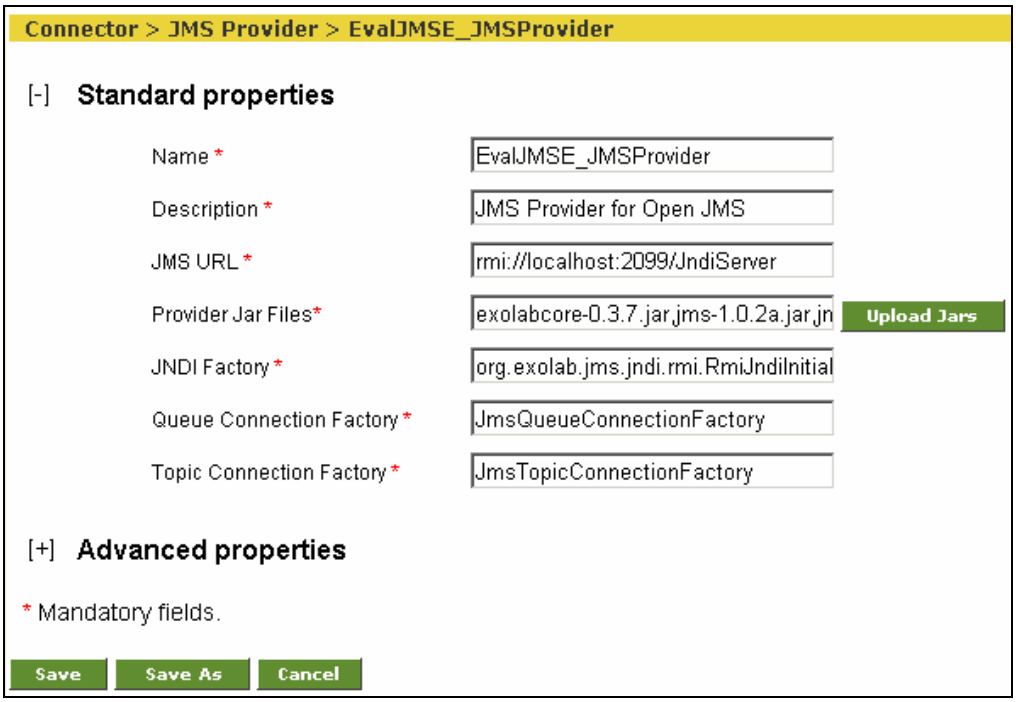



Figure 7.9: Edit JMS Provider Activity

A detailed description of fields on this screen is explicated in the table below.

Table 7.2: Details of Fields on Edit JMS Provider Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the JMS Provider |
| Description | Description of the JMS Provider |
| JMS URL | URL of the JMS Server |
| Provider Jar Files | Jar files, which are used to connect to JMS servers. Click |

| | |
|---|---|
| | **Upload Jars** button to browse and upload Jar files. |
| JNDI Factory | The factory name used to access the external JMS JNDI name service. |
| Queue Connection Factory | Queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. |
| Topic Connection Factory | JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the JMS Provider has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the JMS Provider (see Figure 7.10).

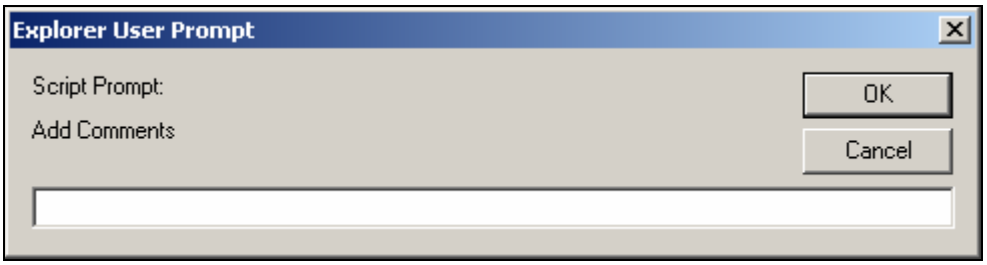



Figure 7.10: Enter Comments

6. Enter the comments in the *Add Comments* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

7. Click **OK** to save the comments. This displays a screen confirming that the JMS Provider has been updated successfully.

| | |
|---|---|
| | JMS provider, pre-created with the Adeptia Suite, is configured for *OpenJMS* server. If you want to use another JMS server, load appropriate Provider Jar files and change other parameters. |

## Editing JMS Event

## (EvalJMSE_JMSEvent)

JMS Event activity is used to trigger the Process Flows, when a message is found on a JMS Server. In JMS Event activity, you can specify connection type and message selector criteria of the JMS Server.

***Steps to edit the JMS Event activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.

2. Click **[+] Events & Triggers** to expand the tree and then click **JMS**. The Manage JMS Event screen is displayed with the list of existing JMS Event activities (Refer to Figure 7.2).

3. Select the radio button adjacent to *EvalJMSE_JMSEvent* and then click **Edit** link. This displays the Edit *EvalJMSE_JMSEvent* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.11).

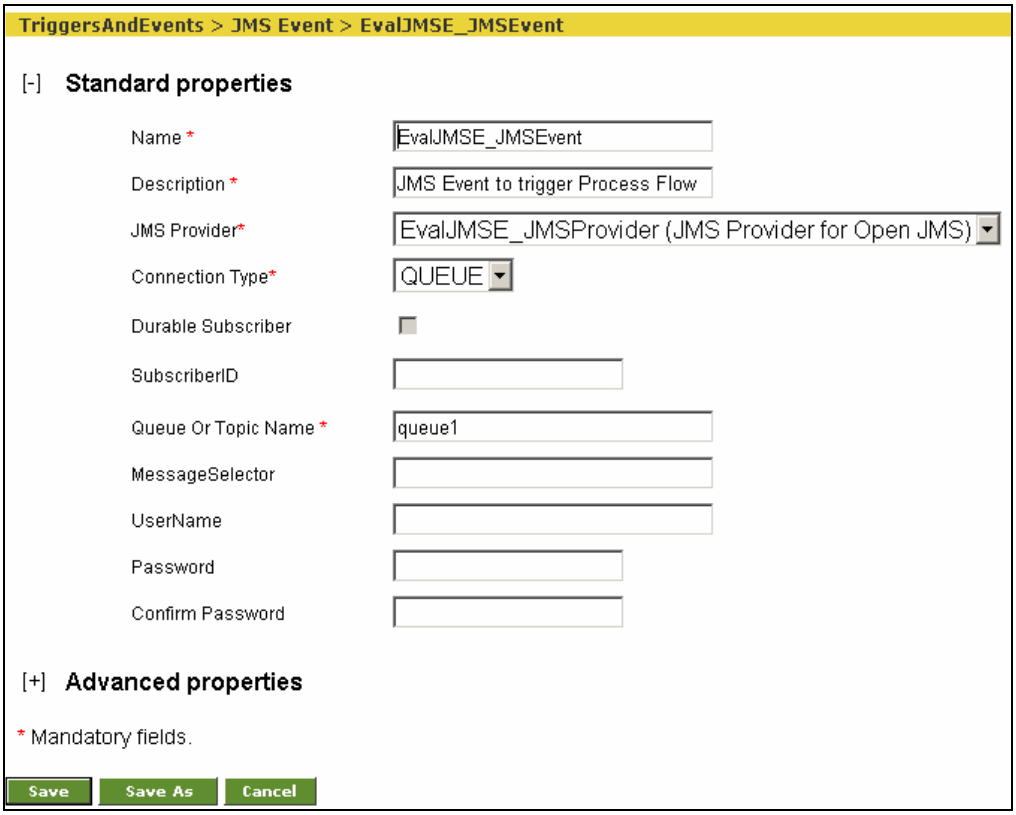



Figure 7.11: Edit JMS Event Activity

A detailed description of fields on this screen is explicated in the table below.

Table 7.3: Details of Fields on Edit JMS Events Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the JMS Event activity |
| Description | Description of the JMS Event activity |
| JMS Provider | JMS Provider used to connect to JMS Server. For more details, refer to the section **Editing JMS Provider**. |
| Connection Type | JMS Connection type, either TOPIC or QUEUE<br>TOPIC Used for one to many messaging. It supports publish subscribe model of messaging.<br>QUEUE Used for one-to-one messaging. It supports Point-to-Point Messaging. |
| Durable | If a client needs to receive all the messages published on a |

| | |
|---|---|
| Subscriber | topic, including the ones published while the subscriber is inactive, it uses a Durable Subscriber. This is applicable only when the connection type is **Topic**. |
| Subscriber ID | Uniquely identify the subscriber of the connection |
| Queue or Topic Name | Name of the Queue or Topic from which JMS Event receives data |
| Create Dynamically | Creates Queue or Topic specified above if it does not already exists in the specified JMS Server |
| Message Selector | Used to filter the messages received from JMS Server. |
| Username | Username required to connect to JMS Server |
| Password | Password required to connect to JMS Server |
| Confirm Password | Re-enter the Password |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the JMS Event activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the JMS Event (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the JMS Event has been updated successfully.

- Pre-created JMS Event activity is configured to listen on **queue1**, which is default queue of the JMS Server.
- JMS Event activity must be deactivated before editing.

## Editing Text Schema

## (EvalJMSE_TextSchema)

Text Schema describes the structure of a text file. Text Schema activity is used to define, how a text file is read or written in a predefined format. To create a Text Schema activity, you need to specify the record separator, field separator and the field names of the text file.

***Steps to edit the Text Schema***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed with the list of existing Text Schema activities (see Figure 7.12).

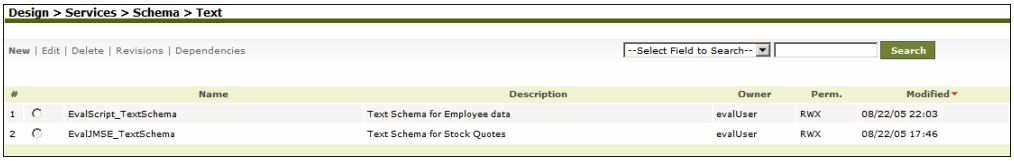



Figure 7.12: Manage Text Schema

4. Select the radio button adjacent to *EvalJMSE_TextSchema* activity and then click **Edit** link. This displays the Edit *EvalJMSE_TextSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.13).

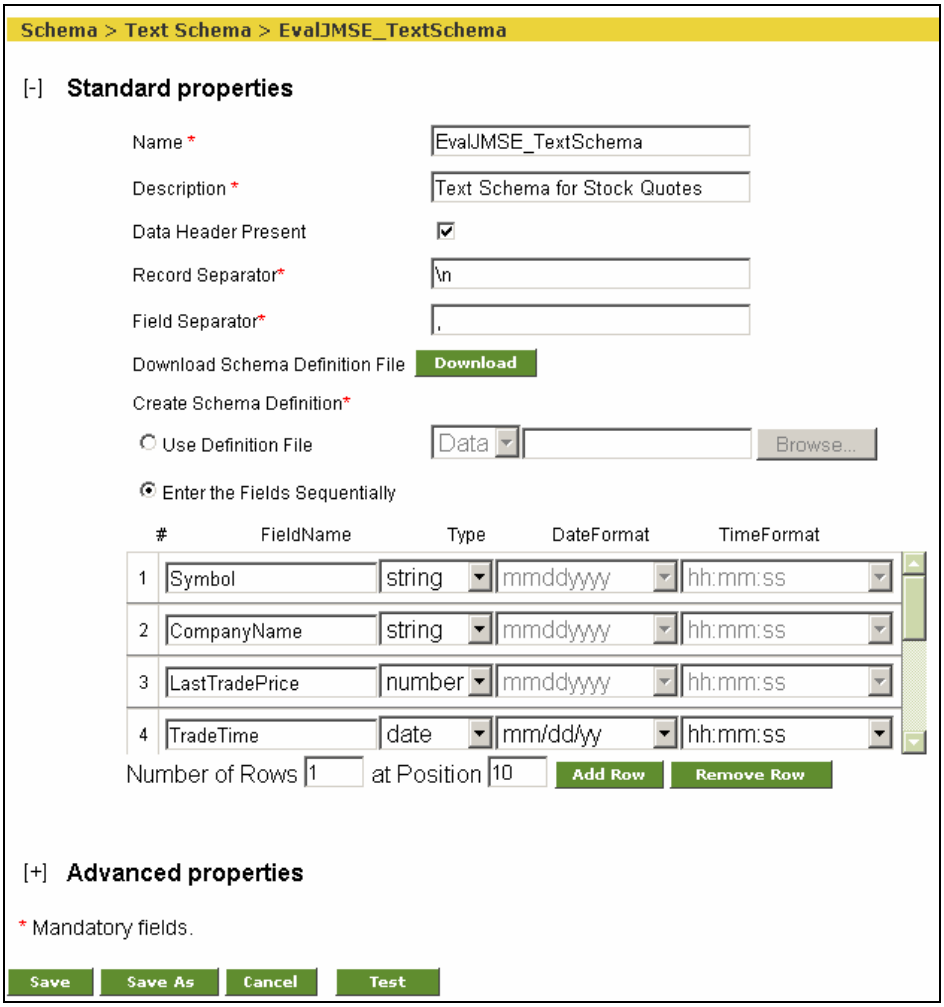



Figure 7.13: Edit Text Schema Activity

A detailed description of fields on this screen is explicated in the table below.

Table 7.4: Details of Fields on Edit Text Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Text Schema activity |
| Description | Description of the Text Schema activity |

| | |
|---|---|
| Data Header Present | Data Header usually contains the titles of the fields in a text file. If data header is present in the text file, check the **Data Header Present** checkbox |
| Record Separator | Character or set of characters that are used to mark the end of a record. For Example **\n** for **New Line**. |
| Field Separator | Character or set of characters that are used to separate fields. For example comma (**,**) |
| Download Schema Definition File | Click **Download** to download the schema definition file. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File<br>▪ Enter the Field Sequentially<br><br>Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |
| Field Name | Name of the Fields |
| Data Type | There are three data types:<br>String — String can be used for any type of data.<br>Number — Contains numbers<br>Date — Contains Date and Time |
| Quotes Handling On | Suppose a character (say $) is specified as Field Separator in a record. Now any $ character in data field of that record (Chocolate$20$perpack) is considered as Field Separator, even though it is part of the data field. In the above example the $ after 20 is also considered as Field Separator, whereas it is data. To avoid this situation put those fields within the double quotes i.e. (Chocolate$"20$perpack") and check **Quotes Handling On** checkbox. Now characters within double quotes are considered as one Field even though there is a $ sign.<br><br>This option is available in *Advanced Properties* of Text Schema. |

5. Make the necessary changes.
6. Once you have changed the required values, click the **Save** button to save the changes. A screen is displayed confirming that the Text Schema activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the text schema (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the text schema has been updated successfully.

## Testing Text Schema
## (EvalJMSE_TextSchema)

You can verify the text schema activity at design time.

***Steps to verify schema activity***

1. Click **Test** button on the Edit Text Schema screen. The Test Schema screen is displayed (see Figure 7.14).

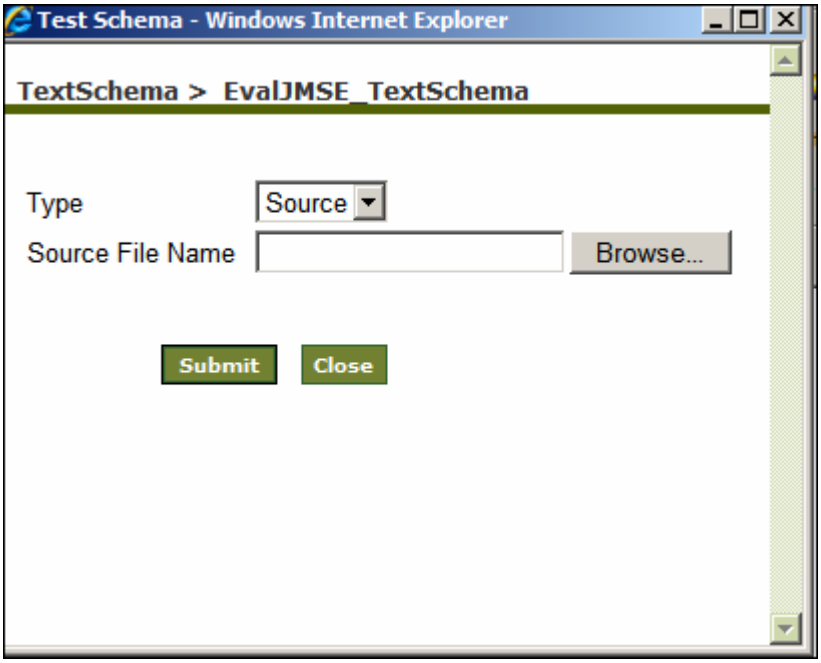



Figure 7.14: Test Schema

2. Select the type of schema to test, from the *Type* drop-down list. By default, *Source* is selected.
3. Enter the full path (with file name and extension) of the source file in the *Source File Name* field.
4. Enter the full path of the XML target file, where it will be generated, in the *Target File Name* field.
5. Enter the full path of the XML file where error records will be stored, in the *Error File Name* field.
6. Click **Submit** button. This tests the validity of the text schema.

## Editing Mapping
## (EvalJMSE_Mapping)

Mapping is used to map data fields of Source Schema with the data fields of Target Schema.

***Steps to edit the Mapping activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed with the list of existing mapping activities (see Figure 7.15).

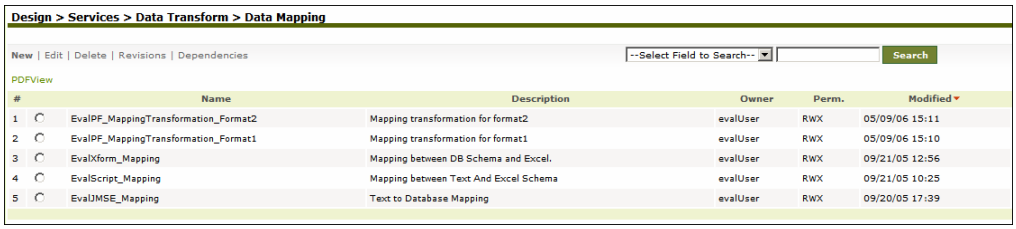



Figure 7.15: Manage Data Mapping

4. Select the radio button adjacent to *EvalJMSE_Mapping* activity and then click **Edit** link. This displays the Edit *EvalJMSE_Mapping* activity screen, with the name and description of the activity displayed in their respective fields (see Figure 7.16).

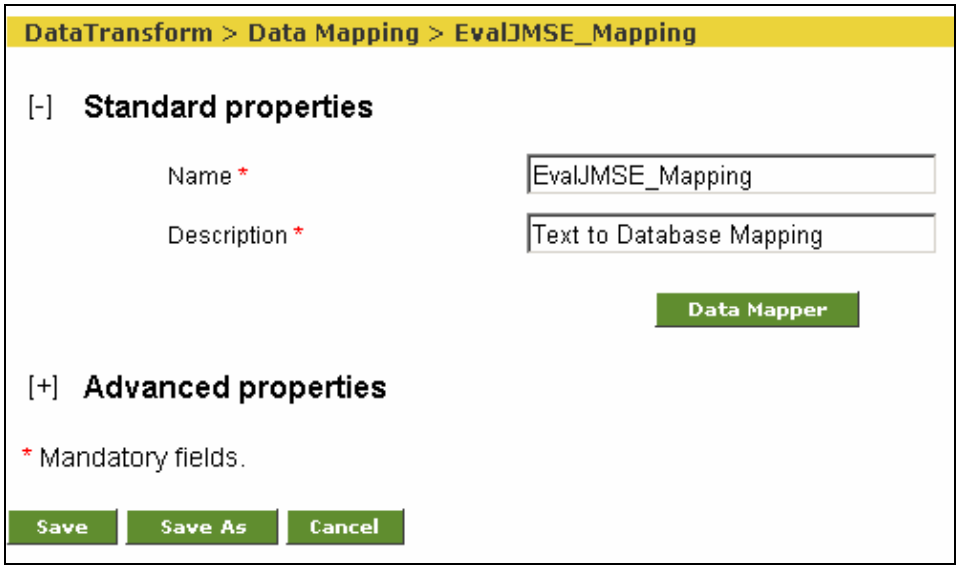



Figure 7.16: Edit *EvalJMSE_Mapping* Activity

5. Click **Data Mapper** button. The Data Mapper applet is displayed showing the mapping between the data fields of the source and target schema (see Figure 7.17).

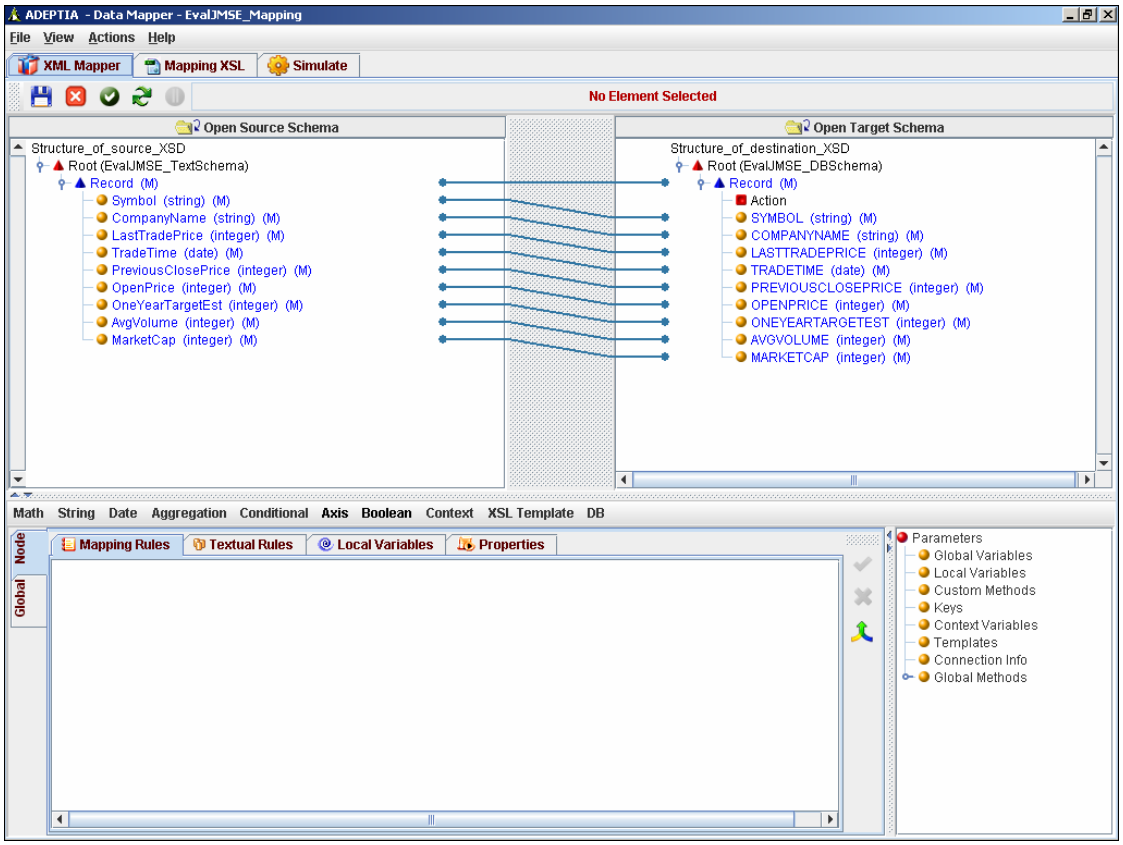



Figure 7.17: Mapping between Source and Target Schemas

6. Make the necessary changes to the mapping between the source and target schemas.

7. Once you have made the required changes, save the mapping by clicking **File** menu and selecting **Save**. A dialog box is displayed confirming that the mapping activity has been saved successfully.

| | |
|---|---|
|  | Alternately, you can save the mapping by clicking **Save** (  ) button on the Tool Bar. |

8. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping. (see Figure 7.18).

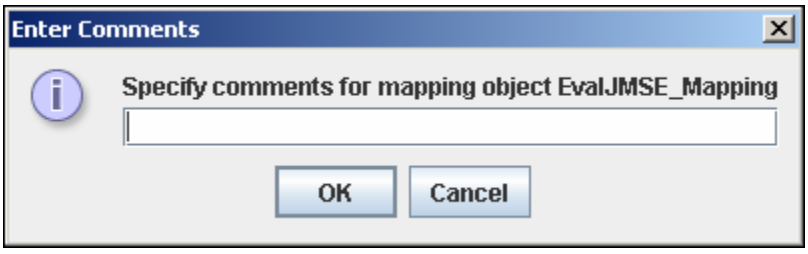



Figure 7.18: Enter Comments (Mapping)

9. Enter the comments in the *Specify comments for mapping object (object name)* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

10. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.
11. Exit the Data Mapper applet by clicking **File** menu and selecting **Exit**.

## Editing Database Driver (EvalJMSE_DBDriver)

Database Driver is used to specify the type of database and to upload driver jar files required to connect to that database.

### *Steps to edit Database Driver*

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **Database Driver**. The Manage Database Driver screen is displayed with the list of existing Database Drivers (see Figure 7.19)

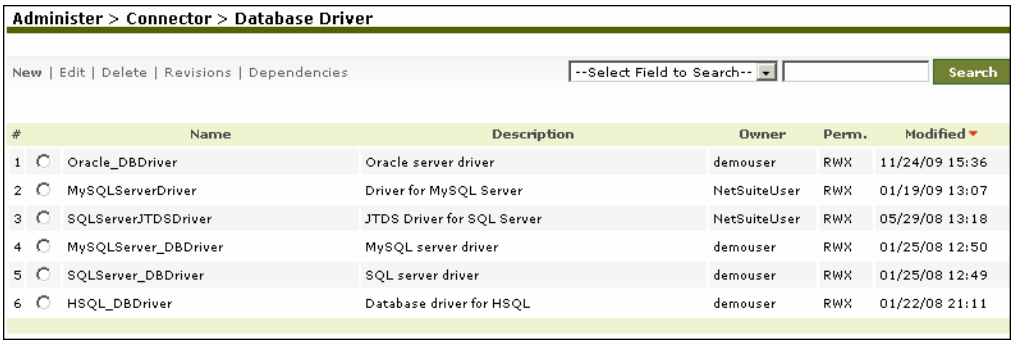



Figure 7.19: Manage Database Driver

3. Select the radio button adjacent to the Database Driver activity that is being used in *EvalJMSE_DbInfo* activity and then click **Edit link.** This displays the Edit *DBDriver* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.20).

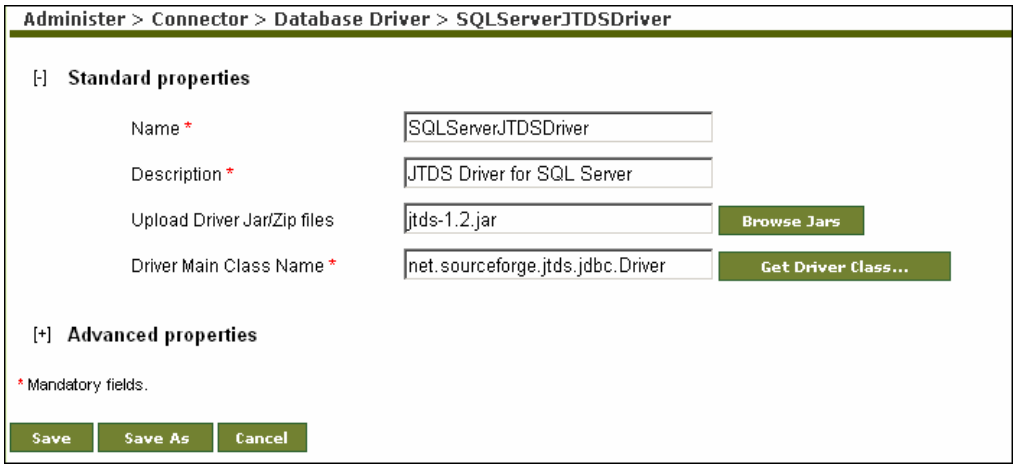



Figure 7.20: Edit *SQLServerJTDSDriver*

A detailed description of fields on this screen is explicated in the table below.

Table 7.5: Details of Fields on Edit Database Driver Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Driver |
| Description | Description of the Database Driver |
| Upload Driver Jar Files | JDBC Driver files, which are used to connect Database Server. Click the **Browse Jars** button to select Jar files. Following is the list of databases and the required Jar files:<br>Oracle — Classes12.jar<br>IBM DB2 (Ver 7.1) — db2java.zip (7.1 version)<br>IBM DB2 (Ver 8.1) — db2jcc.jar<br>MS SQL — msbase.jar, mssqlserver.jar and msutil.jar<br>JTDS- SQL Server — Jtds.jar<br>HSQL DB — hsqldb-1.7.2.jar |
| Driver Main Class Name | Driver Main Class Name is a fully qualified java class name for the main database driver class. The driver class name typically starts with a com., net. or org. followed by the company domain, for example the JDBC driver class from mysql.com is called com.mysql.jdbc.Driver. Click the **Help** button to select Driver Main Class Name from the drop-down list. Following is the list of Driver Main Class Name of different databases: |

| | | |
|---|---|---|
| | Oracle | oracle.jdbc.driver.OracleDriver |
| | IBM DB2 (Ver 7.1) | COM.ibm.db2.jdbc.net.DB2Driver |
| | IBM DB2 (Ver 8.1) | com.ibm.db2.jcc.DB2Driver |
| | MS SQL | com.microsoft.jdbc.sqlserver.SQLServerDriver |
| | JTDS-SQL Server | net.sourceforge.jtds.jdbc.Driver |
| | HSQLDB | org.hsqldb.jdbcDriver |
| | MS Access | sun.jdbc.odbc.JdbcOdbcDriver |
| | MS Excel | sun.jdbc.odbc.JdbcOdbcDriver |

4. Make the necessary changes.
5. Once you have changed the required values, click the **Save** button to save the changes. A screen is displayed confirming that the Database Driver has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Database Driver (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the Database Driver has been updated successfully.

## Editing Database Info

## (EvalJMSE_DBInfo)

Database Info activity is used to specify the JDBC URL or Server URL and Username and Password to access the database. Database Info uses Database driver to connect to specified Database Server.

***Steps to edit the Database Info***

1. Click **[+] Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **Database Info**. The Manage Database Info screen is displayed with a list of existing **Database Info** (see Figure 7.21).

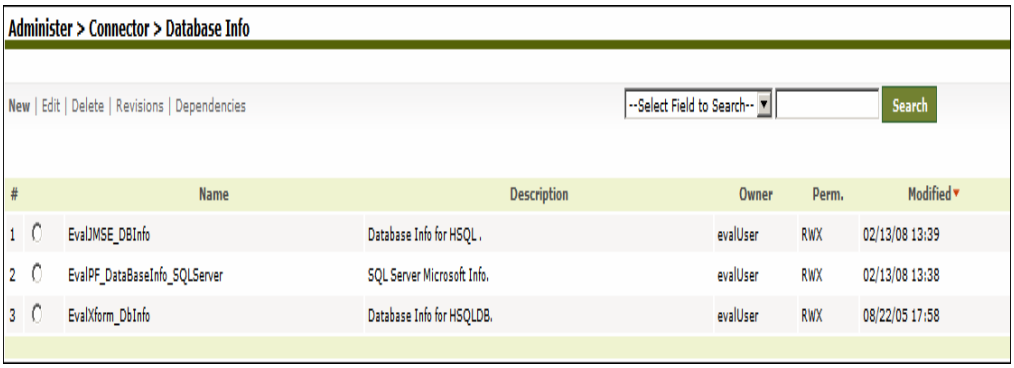



Figure 7.21: Manage Database Info

3. Select the radio button adjacent to *EvalJMSE_DBInfo* activity and then click **Edit** link. This displays the Edit *EvalJMSE_DBInfo* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.22).

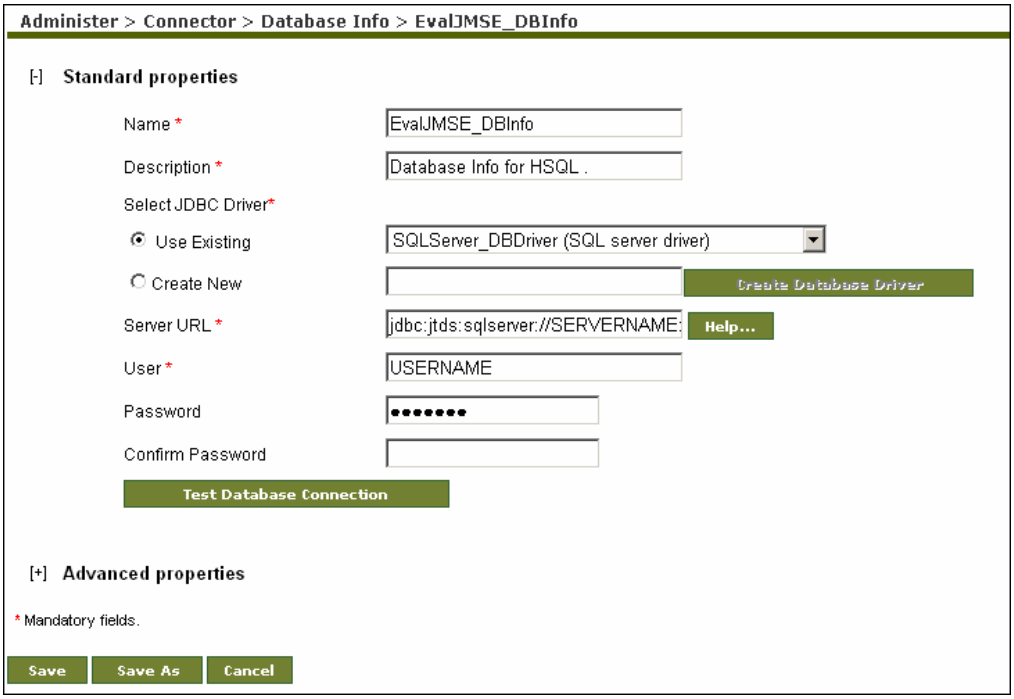



Figure 7.22: Edit *EvalJMSE_DBInfo*

A detailed description of fields on this screen is explicated in the table below.

Table 7.6: Details of Fields on Edit Database Info Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Info |
| Description | Description of the Database Info |
| JDBC Driver | Database Driver is created to connect to the database Server. For more details refer to section **Editing Database Driver**. |

<table>
<tr><td>Server URL</td><td>
Server URL or JDBC URL points to a specific database on a specified database server. There is no standard for JDBC URL. Every JDBC driver uses a slightly different syntax. For Example a JDBC URL for a MySQL database using the <code>com.mysql.jdbc.Driver</code> direct from MySQL might look like this:<br><br>
<code>jdbc:mysql://localhost/databaseName</code>. To specify Server URL, Click the Help button and enter the following information:
<table>
<tr><td>Database Type</td><td>Type of the database</td></tr>
<tr><td>Host Name</td><td>Name of the server on which database server is running</td></tr>
<tr><td>Port</td><td>Port at which database server is running</td></tr>
<tr><td>Database Name</td><td>Name of the database</td></tr>
</table>
</td></tr>
<tr><td></td><td>
Following is the list of Server URL's of different databases:
<table>
<tr><td>Oracle</td><td>jdbc:oracle:thin:@databaseserver:1521:orcl</td></tr>
<tr><td>IBM DB2 (Ver 7.1)</td><td>jdbc:db2://databaseserver:6789/TOOLSDB</td></tr>
<tr><td>IBM DB2 (Ver 8.1)</td><td>jdbc:db2://databaseserver:50000/TOOLSDB</td></tr>
<tr><td>MS SQL</td><td>jdbc:microsoft:sqlserver://databaseserver:1433;DatabaseName=master</td></tr>
<tr><td>SQL JTDS</td><td>jdbc:jtds:sqlserver://databaseserver:1433/master</td></tr>
<tr><td>MS Access</td><td>jdbc:odbc:Driver={MicroSoft Access Driver (*.mdb)}; DBQ=c:/test/db1.mdb</td></tr>
<tr><td>MS Excel</td><td>Jdbc:odbc:ExcelJDBCTest<br>where ExcelJDBCTest is the ODBC object that you need to create using DSN.</td></tr>
<tr><td>HSQL DB</td><td>jdbc:hsqldb:hsql://databaseserver:2476</td></tr>
</table>
</td></tr>
<tr><td></td><td>Here database server is the name of the server where database is running.</td></tr>
</table>

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Info has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Database Info (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the Database Info has been updated successfully.

| | |
|---|---|
| | In this sample Process Flow *HSQLDB* is used as the source database. The data is actually available in *demo.script*, which is located in **`AdeptiaServer/AdeptiaServer4.8/ServerKernel/hsqldb/hsql/demo.script`**.<br><br>If you want to use another database, select the appropriate Server URL for that database. |

## Editing Database Schema (EvalJMSE_DBSchema)

Database Schema defines the structure of a database table. A Database Schema is used to define how records can be read from a database table or can be written into a database table. In this sample process flow, Database Schema is being used at the target end. At the target end, it converts the data from an intermediate XML format into the database specific format. Database Schema uses Database Info activity to connect to the database Server. An important point to note is that the Database Schema does not directly take part in the creation of Process Flow. Since it is selected during the creation of the Database Target activity only this Database Target activity needs to be used in the Process Flow.

***Steps to modify the Database Schema activity***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Database**. The Manage Database Schema screen is displayed with a list of existing Database Schemas (see Figure 7.23).

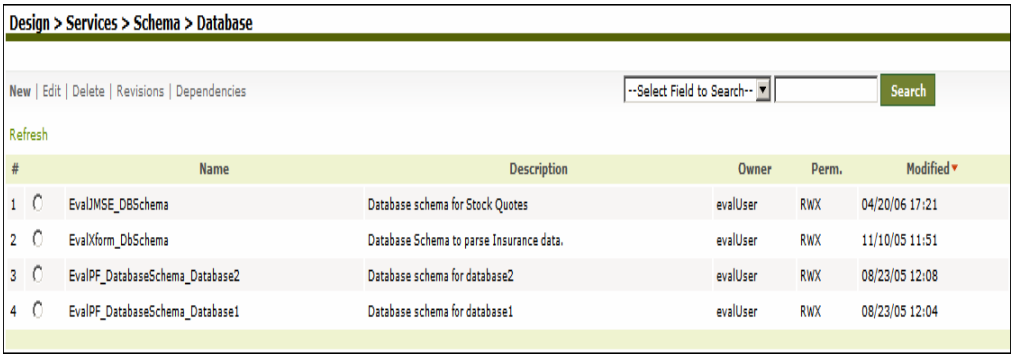



Figure 7.23: Manage Database Schema

4. Select the radio button adjacent to *EvalJMSE_DBSchema* activity and then click **Edit** link. This displays the Edit *EvalJMSE_DBSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.24).

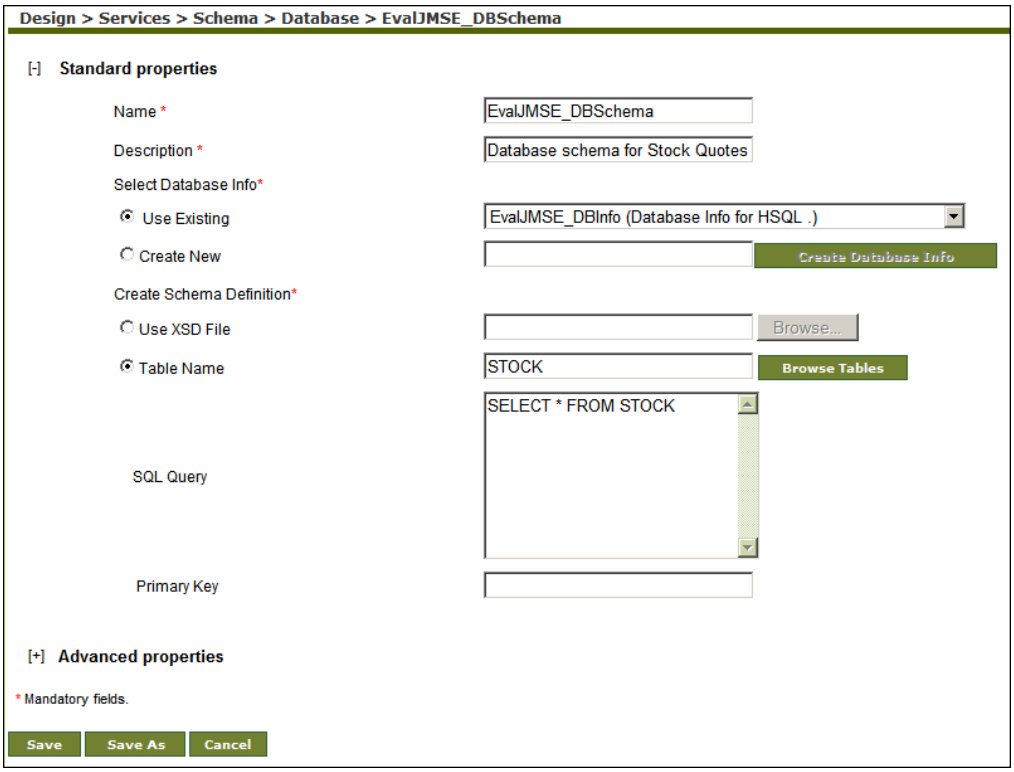



Figure 7.24: Edit *EvalJMSE_DBSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 7.7: Details of Fields on Edit Database Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Schema |
| Description | Description of the Database Schema |
| JDBC Driver | Database Info is created to connect to the specified Database Server. For more details refer to section **Editing Database Info**. |
| Create Schema Definition | Schema Definition can be created using one of the following options:<br>▪ Use XSD File<br>▪ Table Name<br>Database Schema used in this Process Flow is created using second option i.e. Table Name. To select database tables, select **Table Name** radio button and then Click the **Browse Tables**. **Select Table** screen is displayed with the list of database Table. Select the required table and click **Get Columns** button. Click **Close** button to close the **Select Table** screen and return to Database Schema screen.<br>SQL Query box automatically gets populated after selecting database tables. |

5. Make the necessary changes.

6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Schema has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database schema (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

8. Click **OK** to save the comments. This displays a screen confirming that the database schema has been updated successfully.

## Editing Database Target (EvalJMSE_DBTarget)

Database Target is used to insert the data into a database server. Database Target uses Database Info and Database Schema to get list of tables and their fields.

***Steps to update the Database Target***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **Database**. The Manage Database Target screen is displayed with the list of existing Database Target activities (see Figure 7.25).

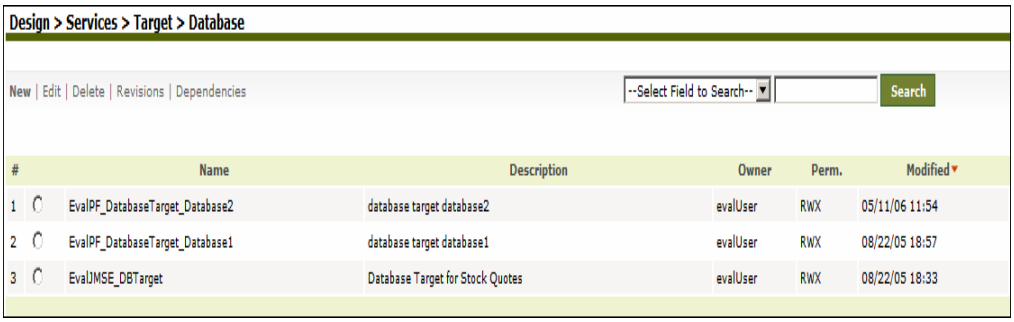



Figure 7.25: Manage Database Target

4. Select the radio button adjacent to *EvalJMSE_DBTarget* activity and then click **Edit** link. This displays the Edit *EvalJMSE_DBTarget* activity screen, with the properties of the activity displayed in their respective fields (see Figure 7.26).

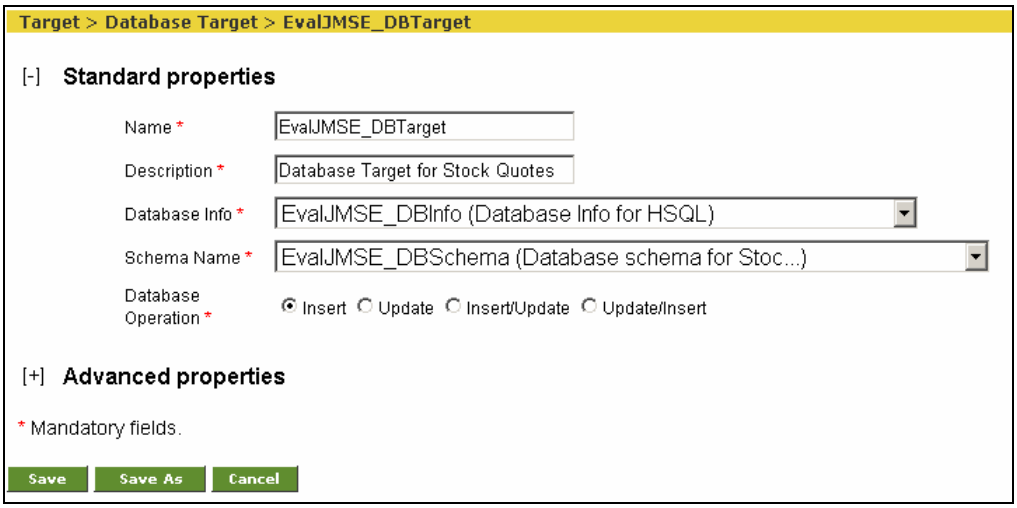



Figure 7.26: Edit *EvalJMSE_DBTarget* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 7.8: Details of Fields on Edit Database Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Database Target |
| Description | Description of the Database Target |
| Database Info | Database Info is created to connect to the specified Database Server. For more details refer to section **Editing Database Info**. |
| Schema Name | Database Schema, which describes the structure of database table. For more details refer to section **Editing Database Schema**. |

| | |
|---|---|
| Database Operation | Database operation specifies how data records are inserted into database tables. Select one of the following database operations:<br>▪ Insert<br>▪ Update<br>▪ Insert/Update<br>▪ Update/Insert<br><br>When **Insert** option is selected, records are inserted into the database tables. If records already exist in the database table, new records are added in the database table along with existing records. When a column (e.g. **CompanyName**) of database table is marked as Primary Key, more than one record cannot exist in the database table for the same Company Name. In this case if data of a company (i.e. XYZ) already exist, insert operation fails and data is not inserted into the database.<br><br>**Update** option is selected when you want to update the existing record. To use updated option a column of database must be marked as Primary Key. When Update option is selected, database target first checks which column of the database table is marked as Primary Key. Suppose CompanyName column is marked as Primary Key. Now database target checks whether data of a company (e.g. XYZ) already exist or not. If data for XYZ already exists, database target updates the existing records. If none of the column is marked as Primary key, Update operation fails.<br><br>When **Insert/Update** option is selected, database target first tries to insert the data into database table. If insert operation fails, database target tries to update the data.<br><br>When **Update/Insert** option is selected, database target first tries to update the database table. If update operation fails, database target tries to insert the data.<br><br>**Note:** To know, how to mark a column of a database table as Primary Key, refer to the documentation of Database Server you are using.<br><br>In this sample Process Flow **Insert** option is used. Every time this Process Flow executes, it adds data along with the existing data. |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Database Target activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the database target (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the database target has been updated successfully.

# CREATING PROCESS FLOW

## (EvalJMSE_ProcessFlow)

A Process Flow is the set of activities arranged in a sequence to perform specific task(s). It is created by combining various activities i.e. Source, Target, Schema or Transformer etc. Process Designer is used to create a Process Flow. Process Designer has a list of activities created. You only need to arrange them in a logical sequence and connect them with the BPMN Flows.

### *Steps to create JMS Driven Process Flow*

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (see Figure 7.27).

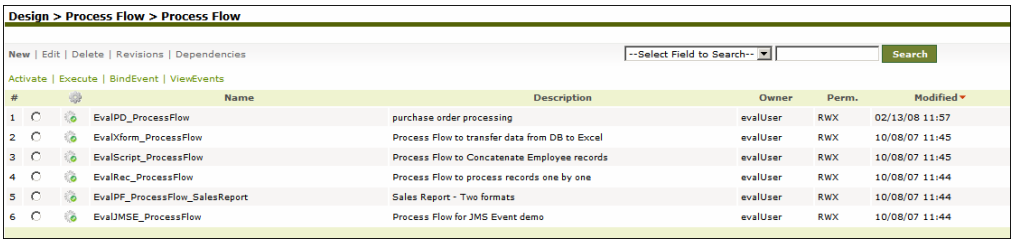



Figure 7.27: Manage Process Flow

3. Click the **New** link. The Create Process Flow screen is displayed (see Figure 7.28).

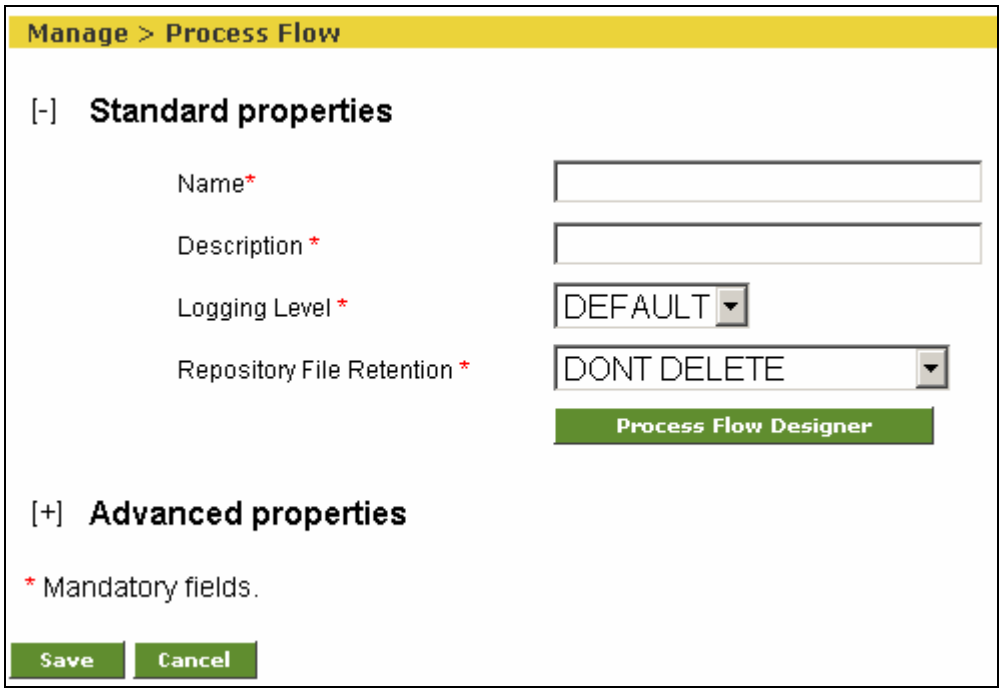



Figure 7.28: Create Process Flow

4. Enter the name and the description of the new Process Flow in the *Name* and *Description* fields respectively.

5. Select the logging level from the *Logging Level* drop-down list.

There are four levels of logging. These are described in the table below.

Table 7.9: Logging Levels

| Level | Description |
|---|---|
| DEBUG | The DEBUG level logs fine-grained informational events that are most useful to debug any problem. Debug level is useful for programmers |
| INFO | The INFO level logs informational messages that highlight the progress of Process Flow execution. In INFO, status (successful or failure) of each activity is shown. |
| ERROR | In ERROR, possible cause of failure of an activity is shown. |
| DEFAULT | If you select Default, logging level, which is set as default in System Configuration, is selected. |

6. Select repository file retention from the *Repository File Retention* option. During execution, the Process Flow creates a temporary repository file to store intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand, sometime these repository files can be helpful in case of the failure of the Process Flow execution. For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files, and target formatted data. By default repository files are being stored in the repository folder of the Adeptia Suite. You can also choose an option to delete them or to archive them in a different location.

There are four options for the Repository File Retention. These are described in the table below.

Table 7.10: Repository File Retention Options

| Retention Option | Description |
|---|---|
| DONTDELETE | Repository files are not deleted after execution of Process Flow. |
| DELETE | Repository files are deleted after the Process Flow is executed. |
| ARCHIVE | Repository files are moved to another location. By default repository files are archived in **C:\repo** folder. |
| DELETE ON SUCCESS | Repository files are deleted only when the process flow is executed successfully and there is no error record. |

7. Click the **Process Designer** button to open Process Designer. The Process Designer screen is displayed (see Figure 7.29).

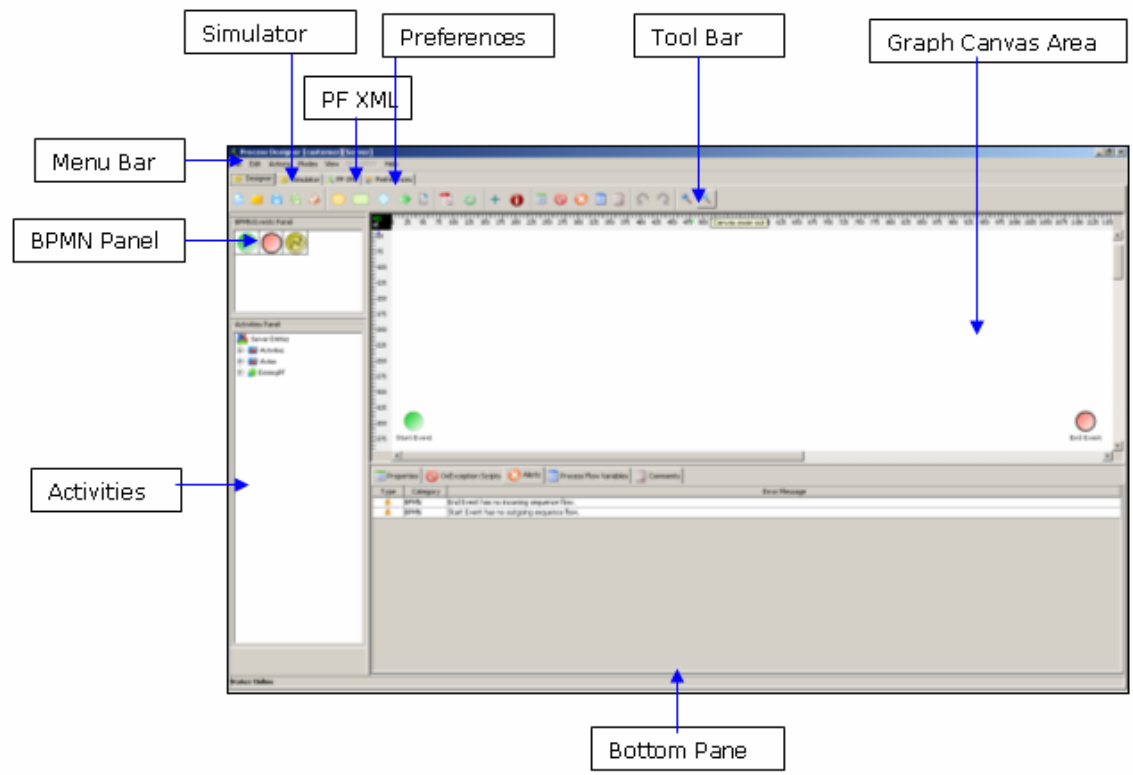



Figure 7.29: Process Designer Applet

The **Activities Panel** has been renamed to **Repository View**. It is not reflected in this.

8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

| | |
|---|---|
|  | Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar. |

9. To create the Context Source, click **[+] Source** in Repository View, to expand the list of Source activities.

10. Click **[+] Context Source** activity to expand the Context Source activity. The Context Source node is displayed.

11. Drag the Context Source node to the Graph Canvas Area (see Figure 7.30).

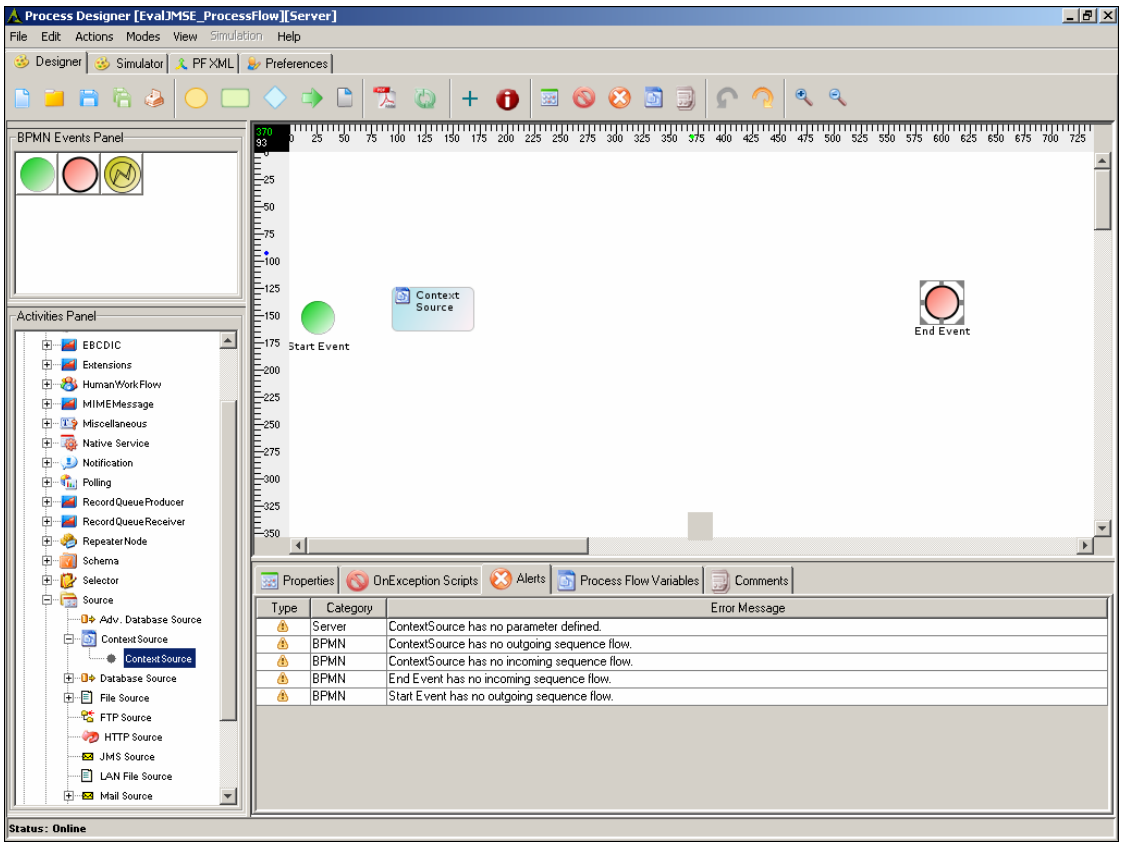



Figure 7.30: Drag Context Source to Graph Canvas

12. Right-click the **Context Source** node and select **View Properties**. Properties of the Context Source activity are displayed in the Bottom Pane (see Figure 7.31).

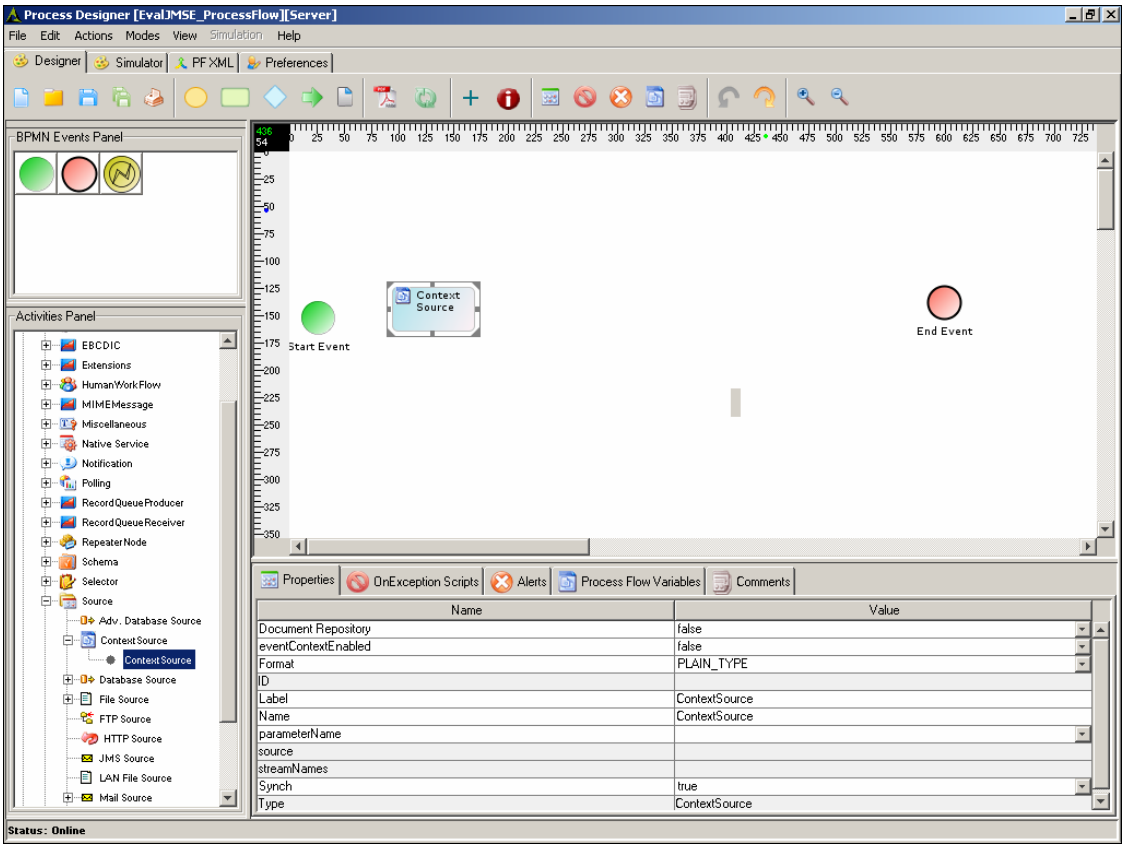



Figure 7.31: View Properties of Context Source Activity

13. Change the Name and Label of the context source as *EvalJMSE_ContextSource.*
14. Change the value of the property *eventContextEnabled* from *false* to *true* (see Figure 7.32).

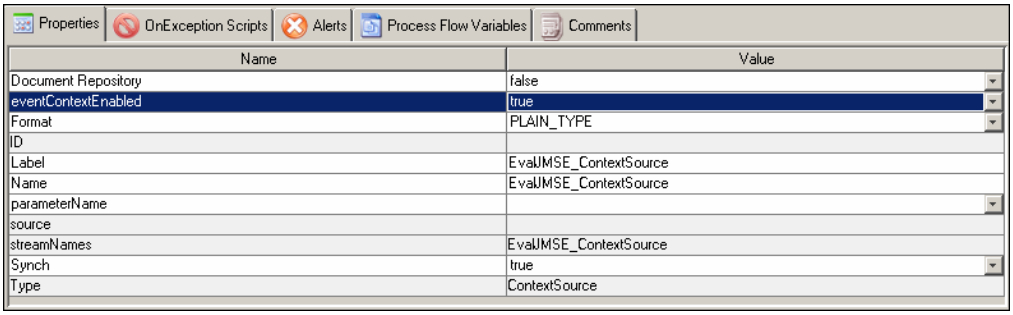



Figure 7.32: Context Source Name

15. Click **[+] Activities** in the Activities Panel to expand the list of services and then click **[+] Schema**. All the items in the **Schema** category are displayed.
16. Click **[+] Text Schema**. A list of existing Text Schema activities is displayed.
17. Select **EvalJMSE_TextSchema** and drag it to the Graph Canvas Area (see Figure 7.33).

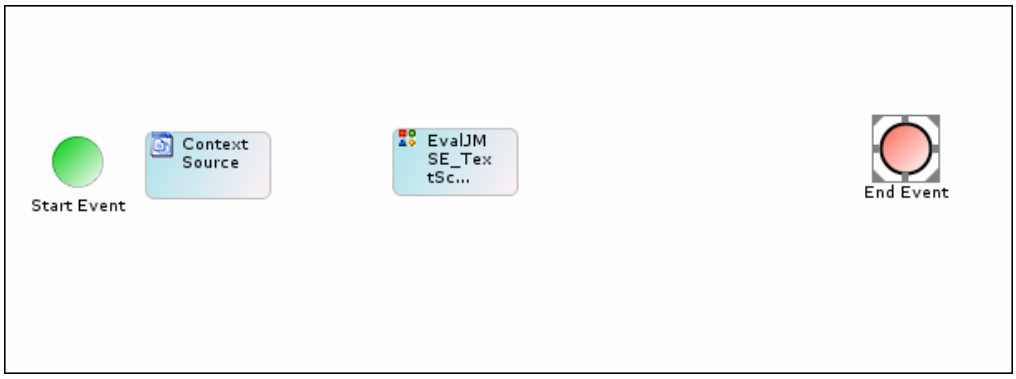



Figure 7.33: Drag Text Schema to Graph Canvas Area

18. Click **[+] DataTransform** and then **[+] Data Mapping**. Select **EvalJMSE_Mapping** activity and drag it to the Graph Canvas Area.
19. Click **[+] Target** and then **[+] Database Target**. Select **EvalJMSE_DBTarget** activity and drag it to the Graph Canvas Area (see Figure 7.34).

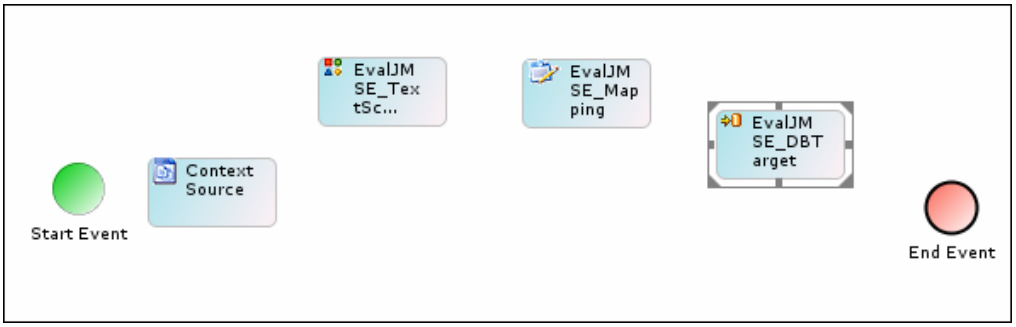



Figure 7.34: Drag Database Target to Graph Canvas Area

20. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
21. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
22. Drag the mouse pointer from *Start Event* to *Context Source* to connect *Start Event* with *EvalJMSE_ContextSource* (see Figure 7.35).

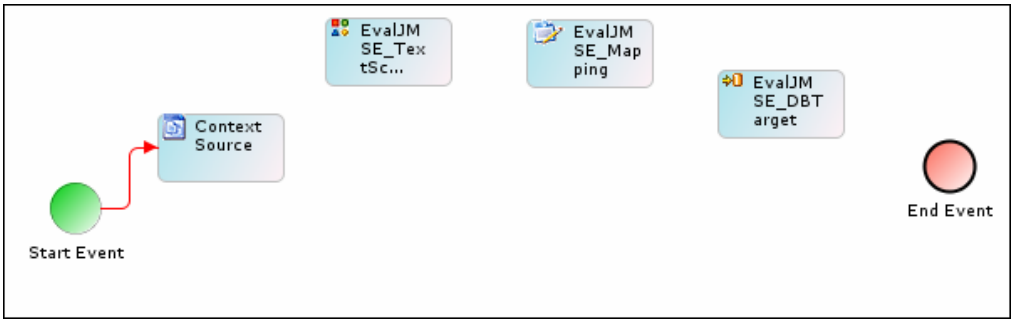



Figure 7.35: Connect Start Event to Context Source

23. Similarly, connect all other activities as shown in Figure 7.36.

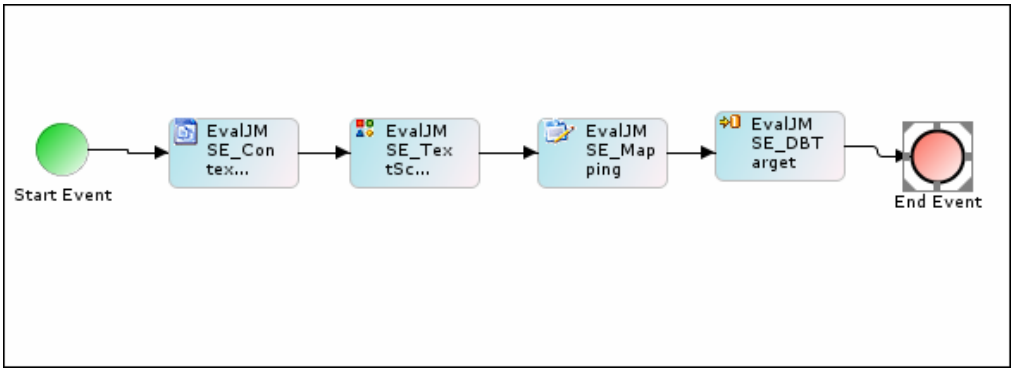



Figure 7.36: Connect all Activities

24. Save the Process Flow by clicking the **File** menu and selecting **Save to Server**. A dialog box is displayed confirming that the *EvalJMSE_ProcessFlow* has been saved successfully. If the *comments* property is enabled, then clicking **Save Process Flow to Server** will display a screen where you need to enter comments related to creating the process flow (see Figure 7.37).

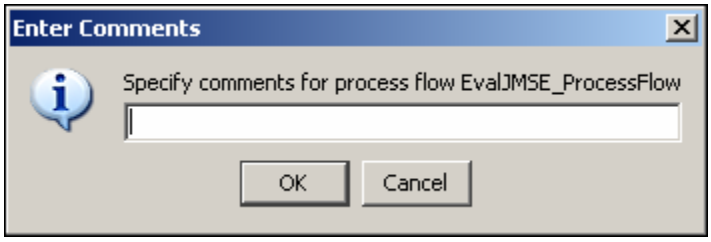



Figure 7.37: Enter Comments (Process Flow)

25. Enter the comments in the *Specify Comments for process flow customer* field.

| | |
|---|---|
| | The comment should be at least 1 character in length. |

26. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.
27. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

# REGISTERING PROCESS FLOW WITH JMS EVENT

After the Process Flow is created it must be registered with the JMS Event. JMS Event triggers the Process Flow when a message is found on the specified JMS Server. To register the Process Flow with the JMS Event, Event Registry activity is created. Event Registry **(EvalJMSE_EventRegistry)** is used to bind **EvalJMSE_JMS** and **EvalJMSE_ProcessFlow** already created. This section describes how to edit the Event Registry.

***Steps to edit the Event Registry***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.

2. Click **[+] Events & Triggers** to expand the tree and then click **Event Registry**. The Manage Event Registry screen is displayed with the list of existing Event Registries (see Figure 7.38).

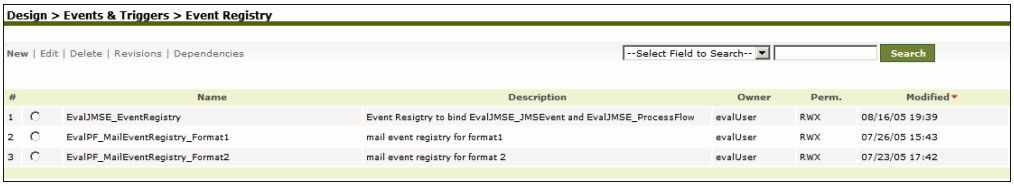



Figure 7.38: Manage Event Registry

3. Select the radio button adjacent to *EvalJMSE_EventRegistry* activity and then click **Edit** link. This displays the Edit *EvalJMSE_EventRegistry* activity screen, with properties of the activity displayed in their respective fields (see Figure 7.39).

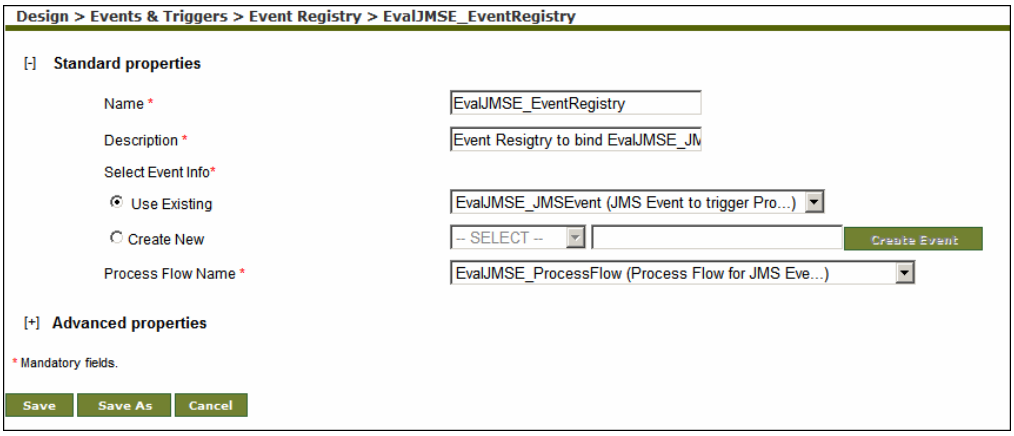



Figure 7.39: Edit *EvalJMSE_Event Registry*

A detailed description of fields on this screen is explicated in the table below.

Table 7.11: Details of Fields on Edit Event Registry Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Event Registry |
| Description | Description of the Event Registry |
| Select Event Info as Existing Event or Create New | Name of the JMS Event, which triggers the Process Flow |
| Process Flow Name | Name of the Process Flow, which is triggered by JMS Event |

4. Make the necessary changes.

5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Event Registry has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Event Registry (refer to Figure 7.10).

6. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

7. Click **OK** to save the comments. This displays a screen confirming that the Event Registry has been updated successfully.

# 8 RECORD TO RECORD SERVICE PROCESS FLOW

This section describes the Record to Record Service Process Flow.

In the Adeptia Suite this process flow is available in:

| BPM Suite | Workflow Suite | Integration Suite | ETL Suite |
|---|---|---|---|
| √ | | √ | √ |

## INTRODUCTION

This Process Flow is used to process data of a positional file. Data of the positional file is processed record by record. One record is taken at a time, processed and finally sent to JMS server. After the record is inserted to the JMS server at the target end, an email is sent for acknowledgement and next record is taken for processing. The whole process continues till all the records of positional file at the source end are processed and inserted to the JMS server.

## SERVICES USED IN THIS SAMPLE PROCESS FLOW

This sample Process Flow illustrates the use of following services of Adeptia Suite:

- File Source
- Positional Schema
- Record to Record Service
- JMS Target
- Mail Notification
- Process Flow Variable

## DESCRIPTION

This sample Process Flow can be outlined as below:

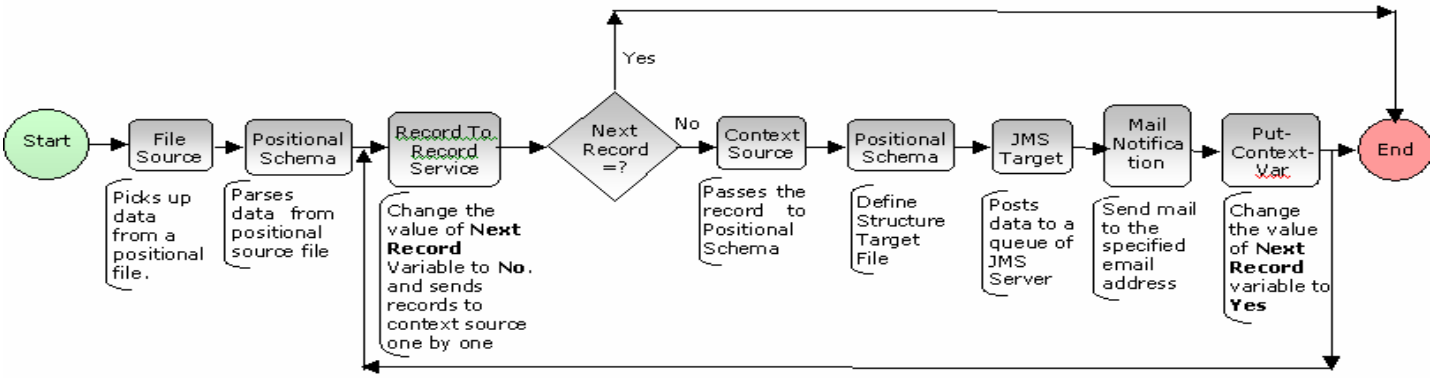



Figure 8.1: Flow Chart to show Process Flow

### File Source (EvalRec_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow, a positional file is specified as source.

### Positional Schema (EvalRec_PositionalSchema)

Positional Schema is used to read data from or write data to a positional file. In this Process Flow two positional schemas are used, one at the source end to read data from the positional source file and the other at the target end to write data to the JMS Target.

### Record to Record Service (EvalRec_Record2Record)

Record to Record Service is used to process input data record by record according to the defined Java Script and provide the output data record by record. In this Process Flow, Record to Record service is used to process input data one record at a time and provide the output in a similar fashion.

### JMS Target (EvalRec_JMSTarget)

JMS Target is used to publish target data in a topic or to post data in a queue of the JMS server. In this Process Flow JMS Target is used to post the data in a queue of the JMS Server.

### Mail Notification (EvalRec_MailNotification)

Mail Notification is used to send mail to users about the execution status of a Process Flow. In this Process Flow mail notification is used to acknowledge the processed records to the users.

## USAGE SCENARIO

This Process Flow can be used, whenever you want to process records one by one.

## DATA DESCRIPTION

Data used in this Process Flow contains records of employees of different departments of a company. Records of the employees are stored in a positional file.

The structure of the positional file being used as source is displayed in the table below.

Table 8.1: Structure of Positional File used as Source

| Field Name | Description | Data Type |
|---|---|---|
| NAME | Name of the employee | String |
| ADDRESS | Address of the employee | String |
| EMAIL_ID | Email address of the employee | String |
| PHONE_NO | Phone number of the employee | Number |
| DOB | Date of Birth of the employee | Date |
| DEPT | Department of the employee | String |
| SALARY | Salary of the employee | Number |
| DOJ | Date of joining of the employee | Date |
| DESIGNATION | Designation of the employee | String |
| AGE | Age of the employee | Number |

Name of the fields of the target file are same as shown in the above table.

## PREREQUISITES

- OpenJMS must be installed and running.
- Queue1, which is the default queue for OpenJMS, is used. So make sure that Queue is available in OpenJMS

## EXECUTING AND MONITORING

This section describes the execution of sample Process Flow and monitoring Process Flow execution.

***Steps to execute the Process Flow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Select the radio button adjacent to *EvalRec_ProcessFlow* Process Flow and then click **Execute** link. The View Process Flow Monitor screen is displayed (refer to Figure 3.2).
4. Click the link **Process Flow Logs** to view the status of the Process Flow execution. The Process Flow Logs is displayed (see Figure 8.2).

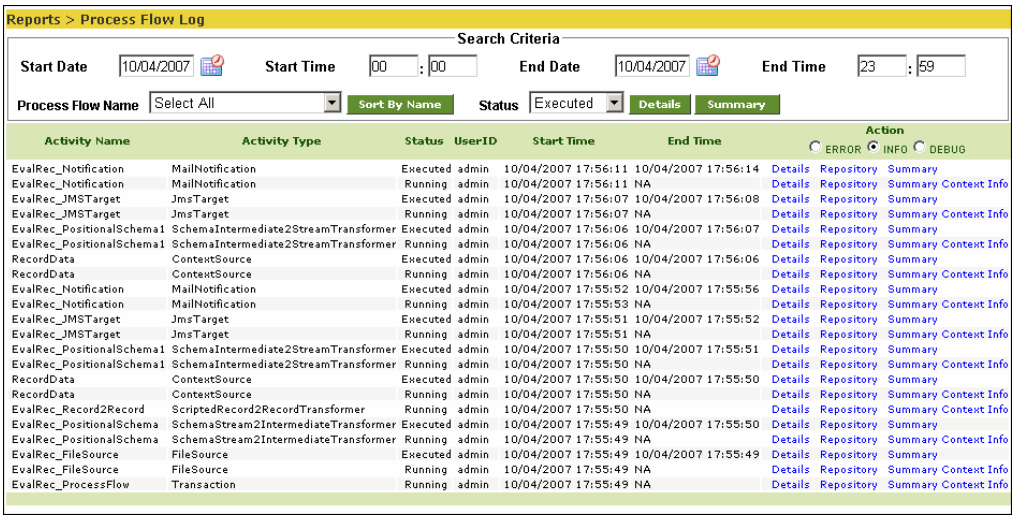



Figure 8.2: View Process Flow Logs

To view the summary of all instances of the process flow execution, click the **Summary** button.

5. Click **Details** under the **Action** column of the activity of which you want to view execution details. This displays the Detailed Process Flow Activity Execution screen (see Figure 8.3).

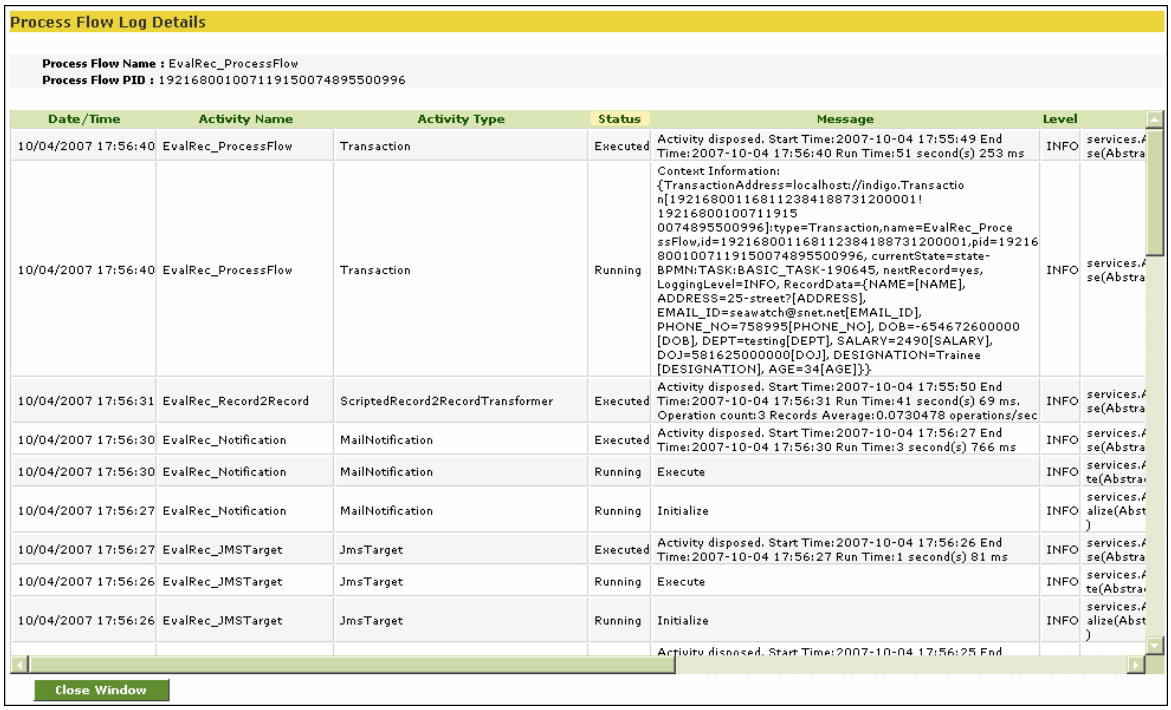



Figure 8.3: View Process Flow Log Details

# EDITING ACTIVITIES

Activities used in this sample Process Flow are pre-created. This section describes how to edit these activities.

## File Source

## (EvalRec_FileSource)

File Source activity is used to specify a file, which is stored in local drive, as source. File Source picks up the specified file and passes it to the other activity. In this Process Flow a positional file **(EvalRecPositionalFile.txt)** is specified as source. This file is stored in **../../Solutions/Demo/EvalRec/** directory.

***Steps to edit the File Source:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **File**. The Manage File Source screen is displayed with the list of existing File Source activities (refer to Figure 4.4).
4. Select the radio button adjacent to *EvalRec_FileSource* activity and then click **Edit** link. This displays the Edit *EvalRec_FileSource* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.4).

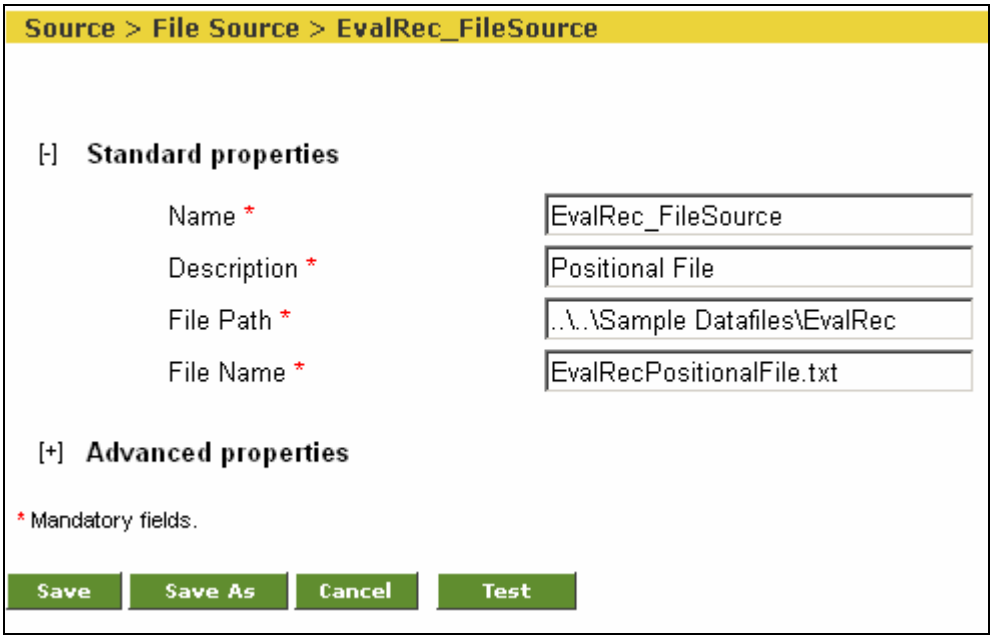



Figure 8.4: Edit *EvalRec_File Source* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.2: Details of Fields on Edit File Source Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the File Source |
| Description | Description of the File Source |
| File Path | Path of the source file.<br>For example:<br>**../../Solutions/Demo/EvalRec/** |
| File Name | Name of the source file. For example:<br>**EvalRecFileSource.txt** |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the File Source Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the file source (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the file source has been updated successfully.

You can verify the file source activity at design time. For this, click **Test Connection**. This verifies the values in the *File Path* and *Filename* fields and checks whether the file actually exists in the specified location.

## Editing Positional Schema (EvalRec_PositionalSchema)

Positional Schema is used to read data from or write data to a positional file. In this Process Flow same positional schema is used in two places, one at the source end to read data from the positional source file and the other at the target end to write data to the JMS Target.

***Steps to edit the Positional Schema:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Positional**. The Manage Positional Schema screen is displayed with the list of existing Positional Schema activities (see Figure 8.5).

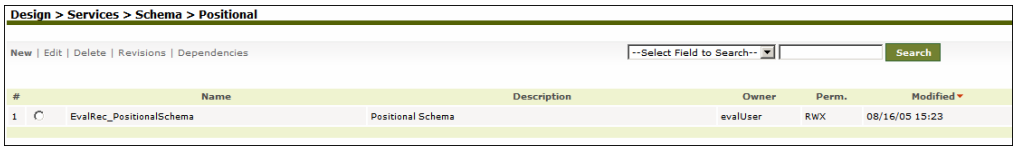



Figure 8.5: Manage Positional Schema

4. Select the radio button adjacent to *EvalRec_PositionalSchema* activity and then click **Edit** link. This displays the Edit *EvalRec_PositionalSchema* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.6).

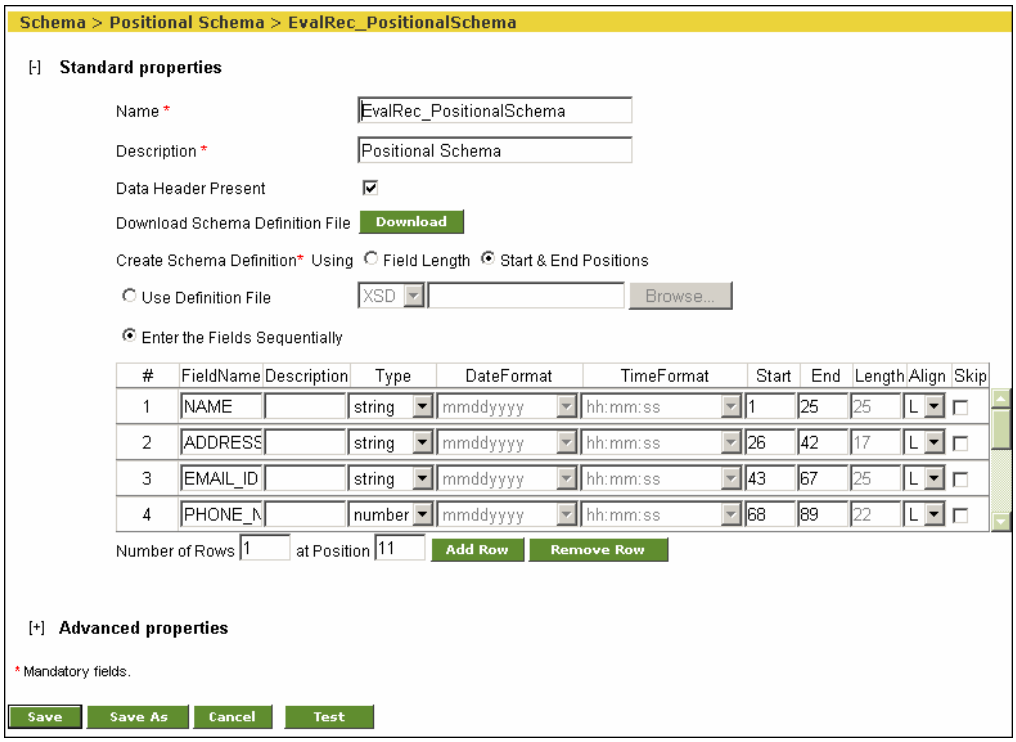



Figure 8.6: Edit *EvalRec_PositionalSchema* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.3: Details of Fields on Edit Positional Schema Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Positional Schema |
| Description | Description of the Positional Schema |
| Data Header Present | Data Header contains the titles of the fields in positional file. If data header is present in the positional file, check the **Data Header Present** checkbox. |
| Download Schema Definition File | To download existing schema definition file click Download button. Else, you can create a new schema definition. |
| Create Schema Definition | Schema can be defined using one of the following options:<br>▪ Use Definition File |

| | |
|---|---|
| | ▪ Enter the Field Sequentially<br>Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially |
| Field Name | Name of the Fields |
| Type | There are three data types:<br>String — String can be used for any type of data.<br>Number — Contains numbers<br>Date — Contains Date and Time |
| Start Position | Start position of the field |
| End Position | End position of the field |
| Length | Length of the field |
| Alignment | Alignment of the Filed<br>**L** if the field is left aligned.<br>**R** if the field is right aligned. |
| Skip | Skip the field while parsing the data from source file to XML |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Positional Schema activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Positional Schema activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the Positional Schema activity has been updated successfully.

## Testing Positional Schema (EvalRec_PositionalSchema)

You can verify the positional schema activity at design time.

***Steps to verify schema activity***

1. Click **Test** button on the Edit Positional Schema screen. The Test Schema screen is displayed (refer to Figure 7.14).
2. Select the type of schema to test, from the *Type* drop-down list. By default, *Source* is selected.
3. Enter the full path (with file name and extension) of the source file in the *Source File Name* field.
4. Enter the full path of the XML target file, where it will be generated, in the *Target File Name* field.

| | |
|---|---|
|  | If you want to test this schema for the target end, select *Target* from the *Type* drop-down list. Now the source file will be an XML file that is generated by the mapping activity used in the process flow and target file will be the full path (with file name and extension) of target file. |

5. Enter the full path of the XML file where errors will be stored, in the *Error File Name* field.
6. Click **Submit** button. This tests the validity of the positional schema.

## Editing Record to Record Service (EvalRec_Record2Record)

Record to Record Service is used to process input data record by record according to the defined Java Script and provide the output data record by record. In this Process Flow Record to Record service is used to process input data one record at a time and provide the output in a similar fashion. Sample Java Code used in the scripted service is displayed in Figure 8.7.

```
// Setting Process Flow variable (nextRecord) value
context.put("nextRecord","no");

// Setting one record into Process Flow Context
context.put("RecordData",record);

// Wait for Process Flow variable(nextRecord) value to be changed to 'yes'
while(true)
{
        try
        {
                String recordExist = (String)context.get("nextRecord");

                if(recordExist.equalsIgnoreCase("no"))
                {
                        Thread.sleep(1000);

                }
                else
                {
                        break;
                }

        }
        catch (InterruptedException e)
        {
                e.printStackTrace();
        }
}
```

Figure 8.7: Sample JAVA Code

***Steps to edit the Record to Record Service:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Record to Record**. The Manage Record to Record screen is displayed with the list of existing Record to Record activities (see Figure 8.8).

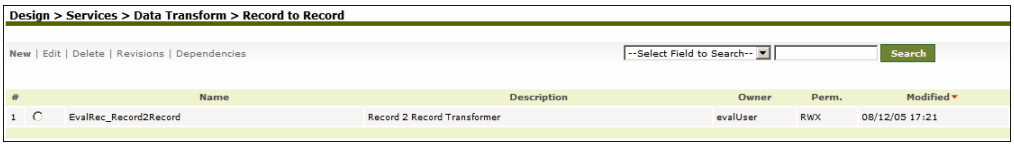



Figure 8.8: Manage Record to Record

4. Select the radio button adjacent to *EvalRec_Record2Record Service* activity and then click **Edit** link. This displays the Edit *EvalRec_Record2Record Service* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.9).

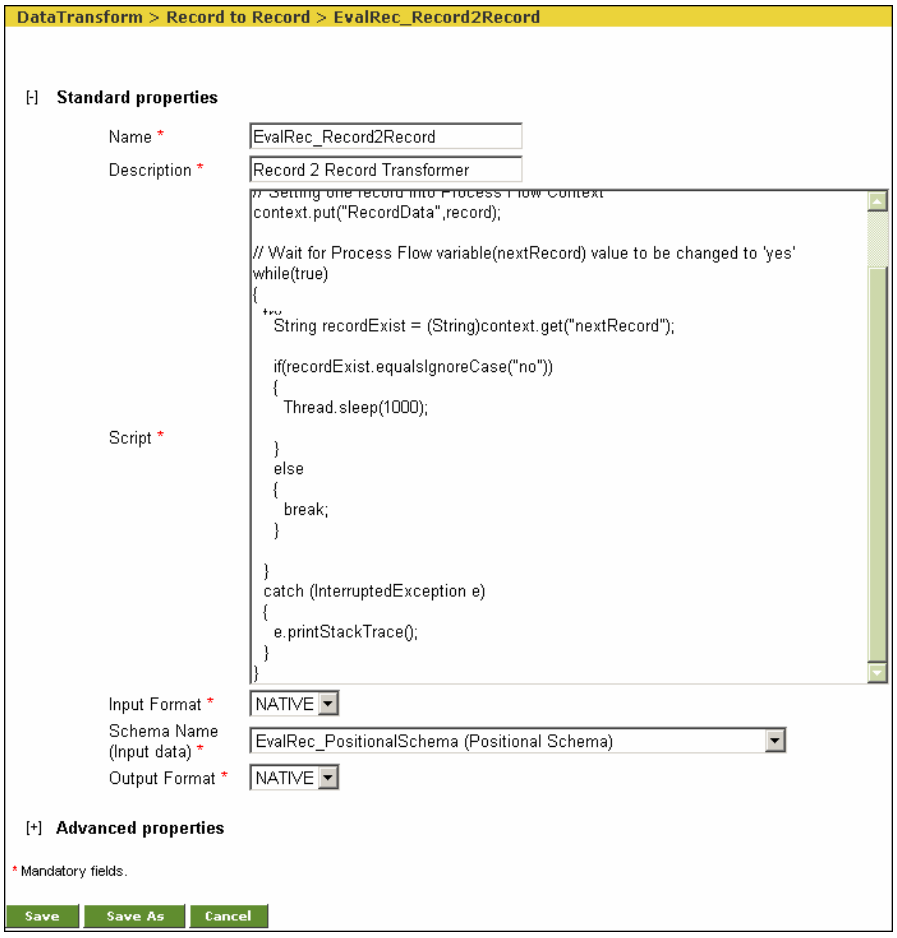



Figure 8.9: Edit *EvalRec_Record2Record* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.4: Details of Fields on Edit Record to Record Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Record to Record activity |
| Description | Description of the Record to Record activity |
| Script | Java Code that you want to run to process the records |
| Input Format | Format of the input data whether Native or XML |
| Output Format | Format of the output data whether Native or XML |
| Schema Name | Name of the Schema activity used to parse the data |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Record to Record Service activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Record to Record Service activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the Record to Record Service activity has been updated successfully.

## Editing JMS Provider

## (EvalRec_JMSProvider)

JMS Provider is used to connect to JMS Server. While creating JMS Provider, you need to specify the Provider Jar files, which are used to connect to JMS server. JMS Event further uses JMS Provider. This should be noted that JMS Provider does not directly take part in the creation of Process Flow. It just helps the JMS Target to connect to the JMS Server.

***Steps to edit JMS Provider:***

1. Click [+] **Administer** to expand the tree and then click **[+] Connector**. All the items in the **Connector** category are displayed.
2. Click **JMS Provider**. The Manage JMS Provider screen is displayed with the list of existing JMS Providers (refer to Figure 7.8).
3. Select the radio button adjacent to *EvalRec_JMSProvider* activity and then click **Edit** link. This displays the Edit *EvalRec_JMSProvider* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.10).

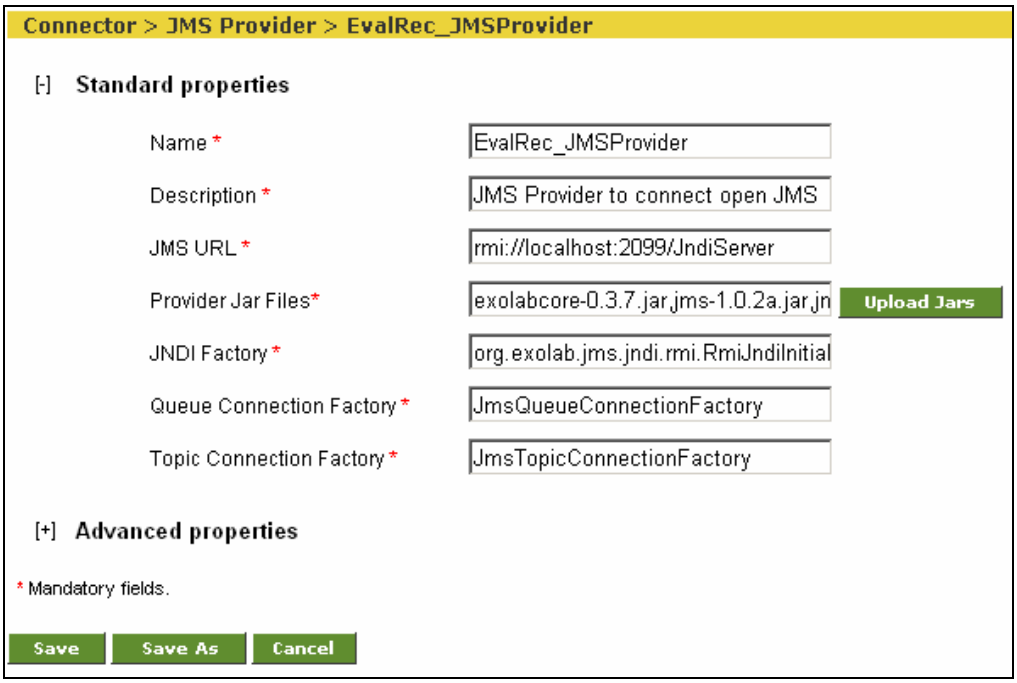



Figure 8.10: Edit *EvalRec_JMS Provider* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.5: Details of Fields on Edit JMS Provider Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the JMS Provider |
| Description | Description of the JMS Provider |
| JMS URL | URL of the JMS Server |
| Provider Jar Files | Jar files, which are used to connect to JMS servers. Click **Upload Jars** button to browse and upload Jar files. |
| JNDI Factory | The factory name used to access the external JMS JNDI name service. |
| Queue Connection Factory | Queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. |
| Topic Connection Factory | JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. |

4. Make the necessary changes.
5. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the JMS Provider has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the JMS Provider (refer to Figure 7.10).
6. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

7. Click **OK** to save the comments. This displays a screen confirming that the JMS Provider has been updated successfully.

| | |
|---|---|
|  | JMS provider, pre-created with the Adeptia Suite, is configured for OpenJMS server. If you want to use another JMS server, load appropriate Provider Jar files and change other parameters. |

## Editing JMS Target (EvalRec_JMSTarget)

JMS Target is used to publish target data in a topic or to post data in a queue of the JMS server. In this Process Flow JMS Target is used to post the data in a queue of the JMS Server.

***Steps to edit the JMS target:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **JMS**. The Manage JMS Target screen is displayed with a list of existing JMS Target activities (see Figure 8.11).

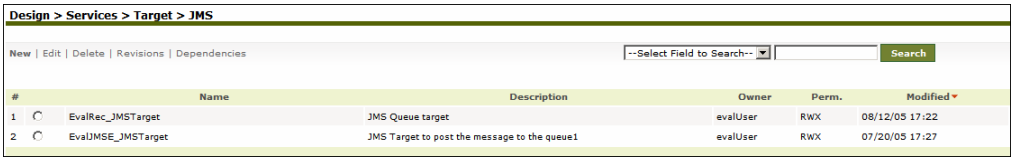



Figure 8.11: Manage JMS Target

4. Select the radio button adjacent to *EvalRec_JMSTarget* activity and then click **Edit** link. This displays the Edit *EvalRec_JMSTarget* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.12).

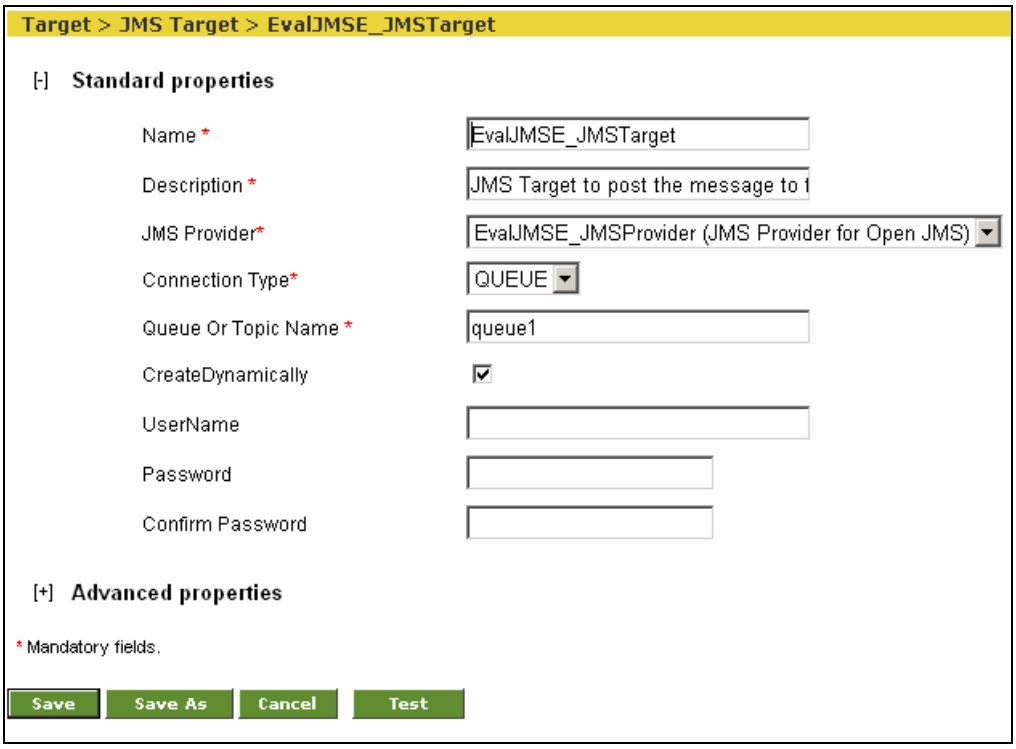



Figure 8.12: Edit *EvalRec_JMSTarget* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.6: Details of Fields on Edit JMS Target Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the JMS Target activity |
| Description | Description of the JMS Target Activity |
| JMS Provider | JMS Provider used to connect to JMS Server. For more details, refer to the section **Editing JMS Provider**. |
| Connection Type | JMS Connection type, either TOPIC or QUEUE<br>TOPIC — Used for one to many messaging. It supports publish subscribe model of messaging.<br>QUEUE — Used for one-to-one messaging. It supports Point-to-Point Messaging. |
| Queue or Topic Name | Name of the Queue or Topic from which JMS Event receives data |
| Create Dynamically | Creates Queue or Topic specified above if it does not already exists in the specified JMS Server |
| Username | Username required to connect to JMS Server |
| Password | Password required to connect to JMS Server |
| Confirm Password | Re-enter the Password |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the JMS Target Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save**

will display a screen where you need to enter comments related to updating the JMS target (refer to Figure 7.10).

7. Enter the comments in the *Add Comments* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

8. Click **OK** to save the comments. This displays a screen confirming that the JMS target has been updated successfully.

| | |
|---|---|
|  | You can verify the JMS target activity at design time. For this, click **Test Connection**. |

## Editing Mail Notification Activity

## (EvalRec_MailNotification)

Mail Notification is used to send mail to users about the execution status of a Process Flow. In this Process Flow mail notification is used to acknowledge the processed records to the users.

***Steps to edit Mail Notification:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Notification** to expand the tree, and then click **Mail Notification**. The Manage Mail Notification screen is displayed with a list of existing Mail Notification activities (see Figure 8.13).

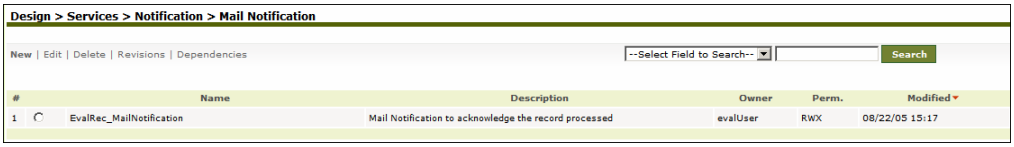



Figure 8.13: Manage Mail Notification

4. Select the radio button adjacent to *EvalRec_MailNotification* activity and then click **Edit** link. This displays the Edit *EvalRec_MailNotification* activity screen, with the properties of the activity displayed in their respective fields (see Figure 8.14).

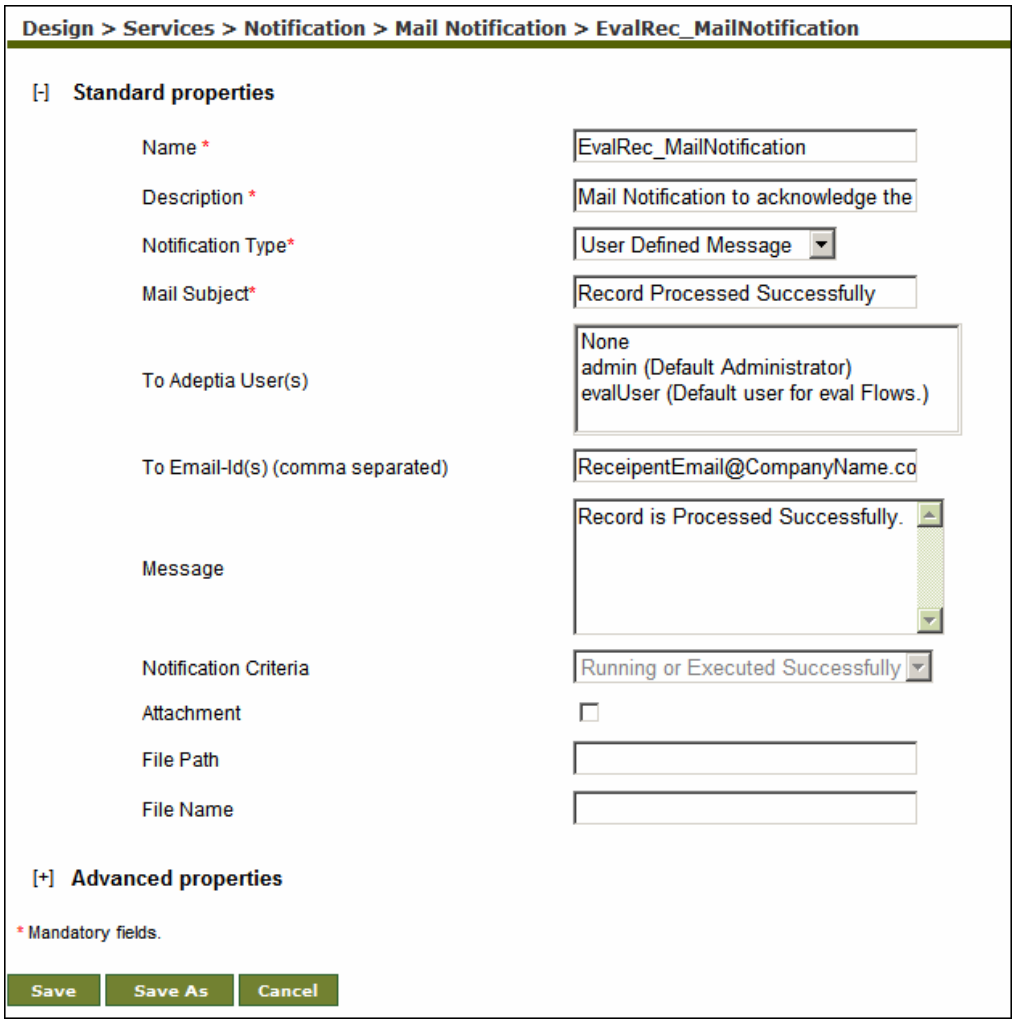



Figure 8.14: Edit *EvalRec_MailNotification* Activity

A detailed description of fields on this screen is explicated in the table below.

Table 8.7: Details of Fields on Edit Mail Notification Screen

| Field Name | Field Description |
|---|---|
| Name | Name of the Mail Notification |
| Description | Description of the Mail Notification |
| Notification Type | There are two types of Mail Notification:<br>▪ User Defined<br>▪ Process Flow Summary<br>In this Process Flow, User Defined notification type is selected. |
| Mail Subject | Enter the subject of the mail, which is sent to specified email address |

| | |
|---|---|
| To Adeptia User(s) | Select user(s) to whom you want to send notification email. You can either select user(s) or you can specifiy email Id(s) in the *To Email-Id(s)* field. |
| To Email-Id(s) (comma separated) | Email address(s) of the recipient (s) |
| Message | Message that is send with the mail |
| Notification Criteria | There are three types of Notification Criteria:<br>▪ Running or Executed Successfully<br>▪ Failure<br>▪ Always<br>Notification criteria are only applicable, when the Notification Type is Process Flow Summary. |
| Attachment | If Attachment checkbox is enabled, the specified Message is send as attachment of the mail |
| File Path | Path of the file in case Attachment checkbox is enabled |
| File Name | Name of the file incase Attachment checkbox is enabled |

5. Make the necessary changes.
6. Once you have made the required changes, click the **Save** button to save the changes. A screen is displayed confirming that the Mail Notification Activity has been updated successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to updating the Mail Notification activity (refer to Figure 7.10).
7. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the Mail Notification activity has been updated successfully.

# CREATING PROCESS FLOW

## (EvalRec_ProcessFlow)

Process Flow is the set of activities arranged in a sequence to perform a specific task(s). Process Flow is created by combining various activities such as Source, Target, Schema or Transformer activities in a logical sequence. Process Designer is used to create a Process Flow. Process Designer has list of activities created. You only need to arrange them in a logical sequence and connect them with BPMN Flows.

***Steps to create EvalRec_ProcessFlow:***

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. The Manage Process Flow screen is displayed with the list of existing Process Flows (refer to Figure 7.27).
3. Click the **New** link. The Create Process Flow screen is displayed (refer to Figure 7.28).
4. Enter the name and the description of the new Process Flow in the *Name* and *Description* fields respectively.
5. Select the logging level from the *Logging Level* drop-down list. There are four level of logging. They are described in Table 7.9.
6. Select repository file retention from the *Repository File Retention* option. During execution Process Flow creates a temporary repository file to store the intermediate data. These repository files can cause unnecessary disk space usage and you may want to delete them after execution of the Process Flow. On the other hand sometime these repository files can be helpful in case of failure of Process Flow execution. For each instance of the Process Flow execution a unique repository folder is created that contains Source, intermediate XML data files and target formatted data. By default repository files are being stored in the repository folder of the Adeptia Suite. You can also choose an option to delete them or to archive them in a different location. There are four options for the Repository File Retention. They are described in Table 7.10.
7. Click the **Process Designer** button to open Process Designer. The Process Designer Screen is displayed (refer to Figure 7.29).
8. Once the Process Designer screen is displayed, a message to synchronize the list of activities and Process Flow with the Adeptia Suite is displayed. Click **OK** to synchronize.

| | |
|---|---|
|  | Alternately, you can synchronize the list of activities and Process Flow with the Adeptia Suite by clicking the **Synchronize** (  ) button displayed on the Tool Bar. |

9. Click **[+] Activities** in Repository View, to expand the list of services and then click **[+] Source**. All the items in the **Source** category are displayed.
10. Click **[+] File Source**. A list of existing File Source activities is displayed.
11. Select **EvalRec_FileSource** and drag it to the Graph Canvas Area (see Figure 8.15).

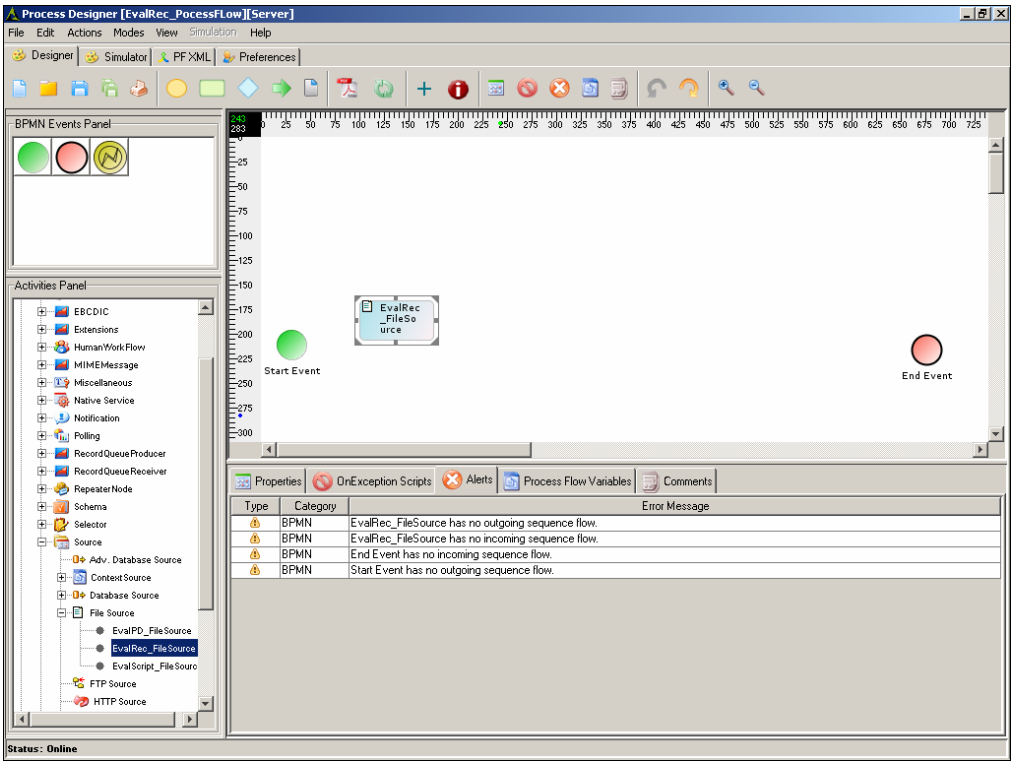



Figure 8.15: Select File Source Activity

12. Similarly, click **[+] Schema** and then **[+] Positional Schema**. Select **EvalRec_PositionalSchema** activity and drag it to the Graph Canvas Area.

13. Click **[+] Datatransform** and then **[+] Record to Record**. Select **EvalRec_Record2Record** activity and drag it to the Graph Canvas Area.

14. To select a BPMN Gateway, click the **BPMN Gateway** (  ) icon in the Palette and drag it to the Graph Canvas Area (see Figure 8.16). In this Process Flow, Gateway is used to check the value of *NextRecord* variable and to decide which path to choose.

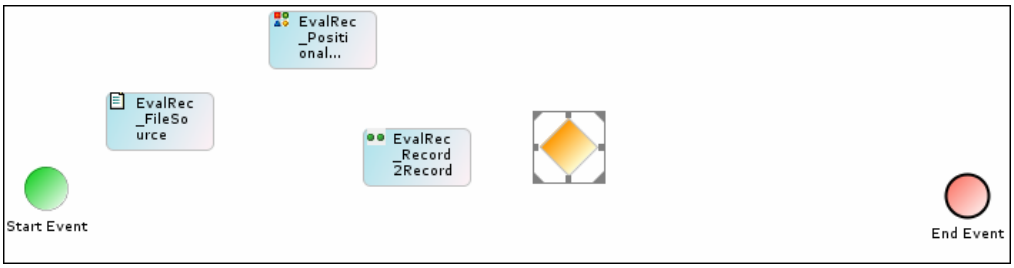



Figure 8.16: Drag BPMN Gateway to Graph Canvas Area

15. To create Process Flow Variable, click *Process Flow Variables* tab in the bottom pane. The *Process Flow Variables* panel is displayed in bottom pane. (see Figure 8.17 )

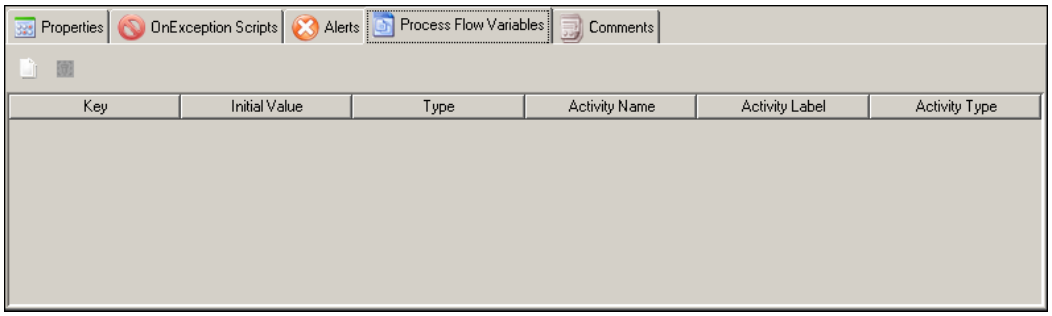



Figure 8.17: Process Flow Variables Panel

16. Click **New Process Flow Variable** ( ) button. The Process flow variable entry dialog box is displayed (see Figure 8.18).

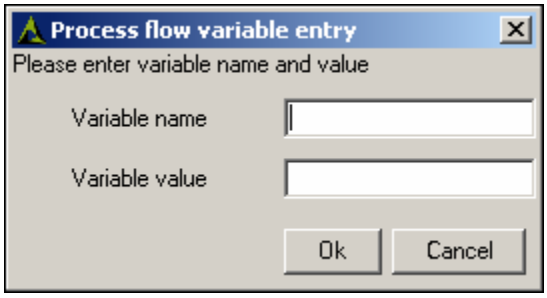



Figure 8.18: Create Process Flow Variable

17. Enter *RecordData* in the *Variable Name* field and click **Ok**. A process flow variable with name *RecordData* is created.
18. Similarly create another process flow variable with name *next Record* with the variable value *yes*.
19. Click **[+] Source** in Repository View, to expand the list of Source activities.
20. Click **[+] Context Source** activity to expand the Context Source activity. The Context Source node is displayed.
21. Drag the Context Source node to the Graph Canvas Area (see Figure 8.19).

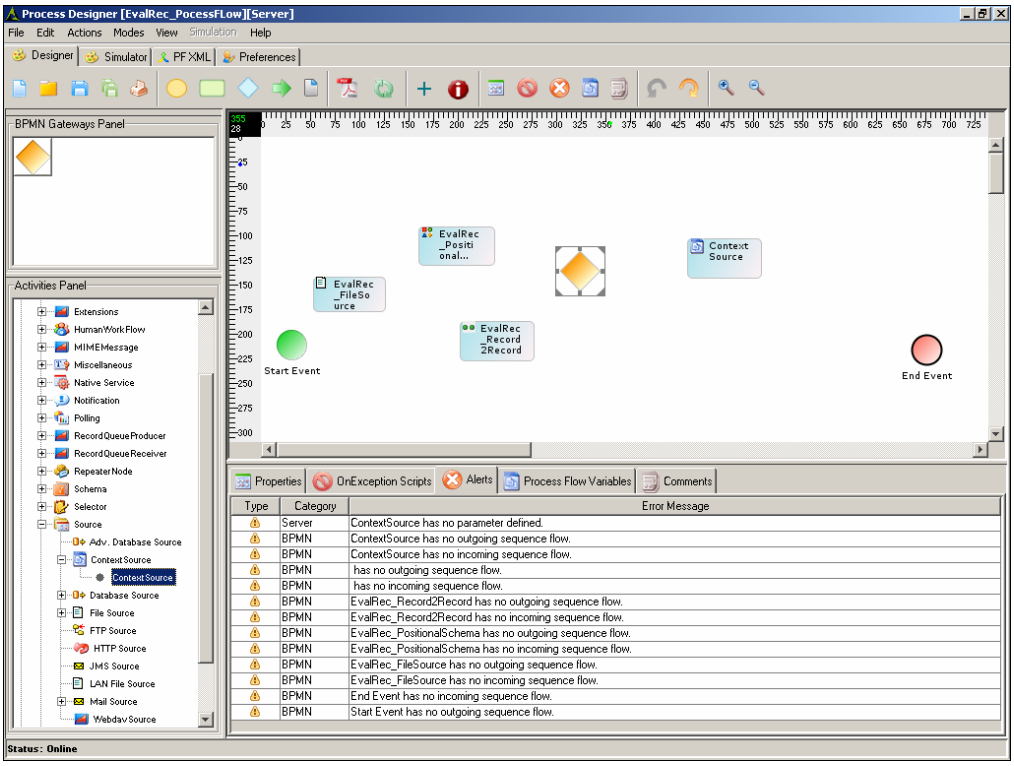



Figure 8.19: Drag Context Source to Graph Canvas Area

22. Right-click the context source variable and select **View Properties**. Properties of the context source variable are displayed in the Bottom Pane (see Figure 8.20).

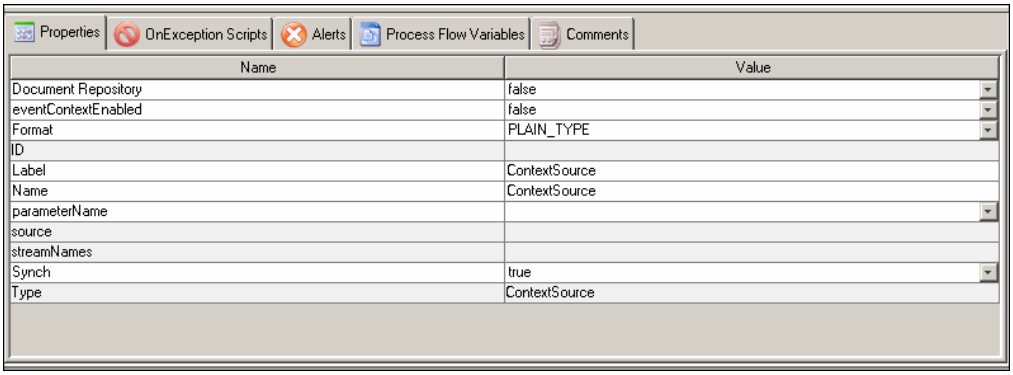



Figure 8.20: Drag Context Source to Graph Canvas Area

23. Change the Name and Label of the context source as *RecordData.*

24. Select *RecordData* from the *parameterName* drop-down list (see Figure 8.21).

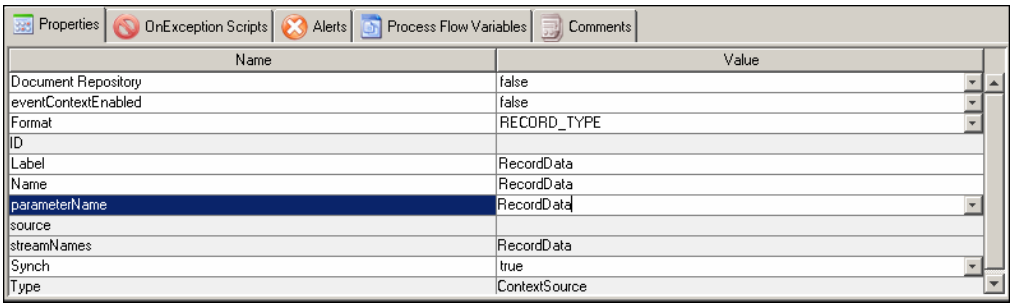



Figure 8.21: Enter Context Source Name

25. Click **[+] Schema** and then **[+] Positional Schema**. Select **EvalRec_PositionalSchema** activity and drag it to the Graph Canvas Area. The Change Activity name dialog box is displayed (see Figure 8.22).

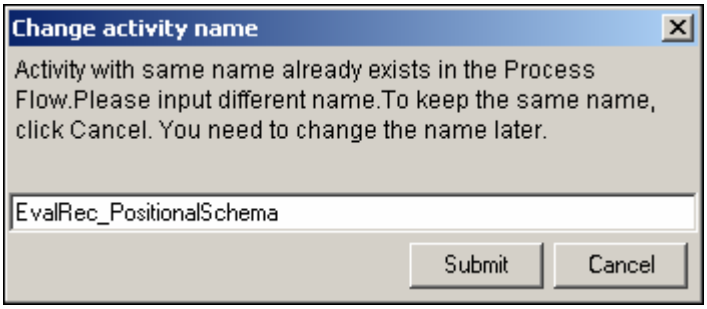



Figure 8.22: Change Activity Name

26. Enter **EvalRec_PostionalSchema1** and click the **Submit** button. The *EvalRec_PositionalSchema* activity is displayed in the Graph Canvas Area.
27. Click **[+] Target** and then **[+] JMS Target**. Select **EvalRec_JMSTarget** activity and drag it to the Graph Canvas Area.
28. Click **[+] Notification** and then **[+] Mail Notification**. Select **EvalRec_MailNotification** activity and drag it to the Graph Canvas Area.
29. Click **[+] Action** in the Repository View, to expand the list of Actions.

| | |
|---|---|
|  | An Action name is always unique. |

30. Select **Put-Context-Var** and drag it to the Graph Canvas Area (see Figure 8.23).

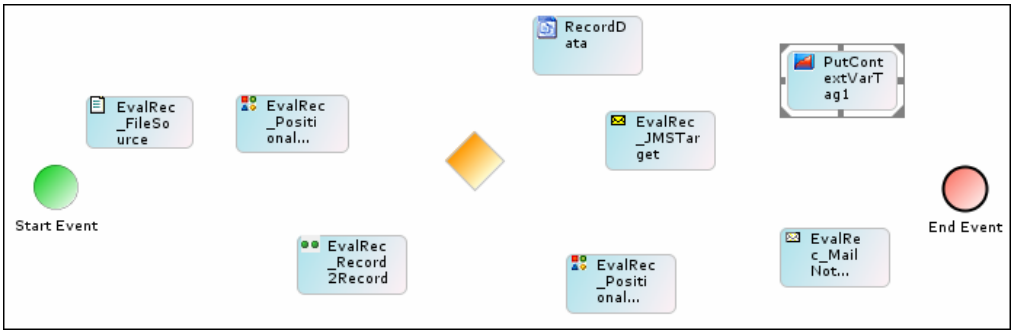



Figure 8.23: Select Put-Context -Var

31. Right-click **Put-Context-Var** in the Graph Canvas Area, and select **View Properties**. Properties of *Put-Context-Var* are shown in the Properties Panel (see Figure 8.24).

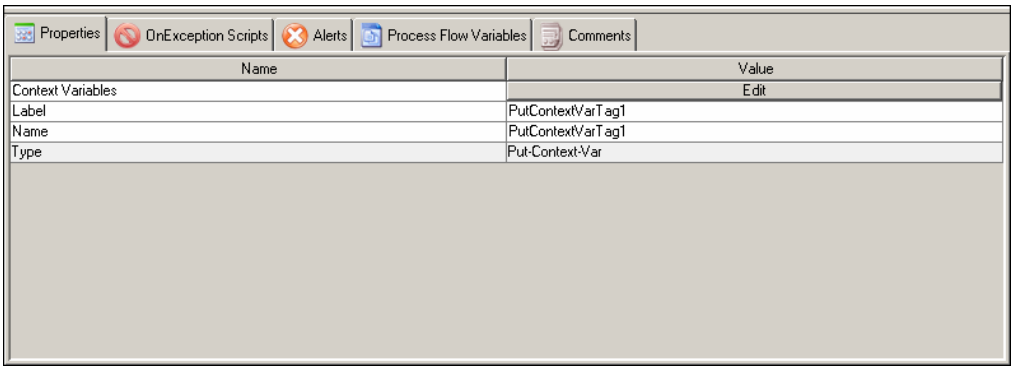



Figure 8.24: View Properties of Put-Context-Var

32. Click *Edit* from the value field of the *context Variable* properties. The Edit Context Variable screen is displayed (see Figure 8.25).

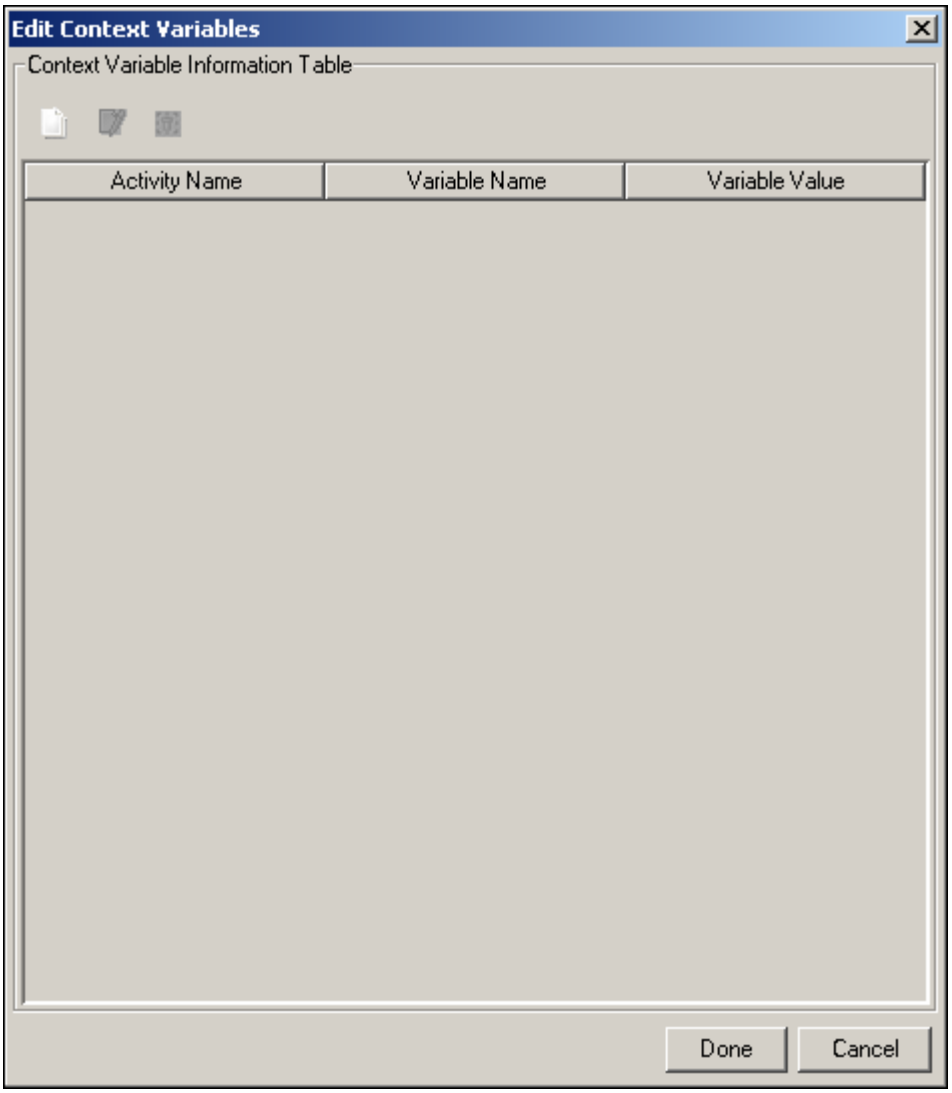



Figure 8.25: Edit Context variable

33. Click **New variable** ( ) button. The Context Variable Information dialog box is displayed (see Figure 8.26).

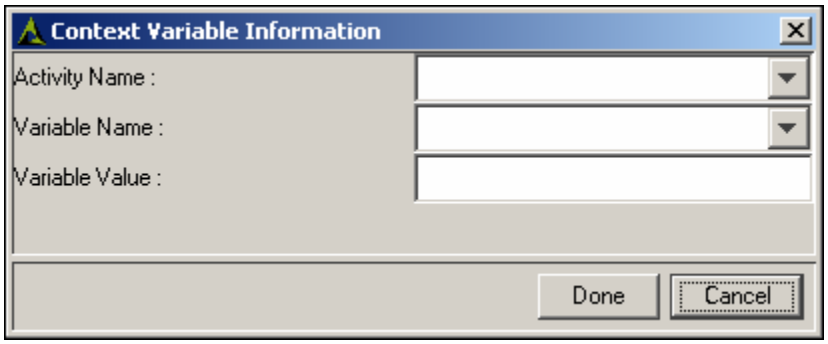



Figure 8.26: Add Context variable

34. Select *nextRecord* from the *Variable Name* drop-down list. *nextRecord* is the name of a Process Flow variable.

35. Enter *yes* in the *Variable Value* field.
36. In this Process Flow, *Put-Context-Var* action is used to change the value of *nextRecord* variable from *no* to *yes*.
37. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
38. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
39. To connect *Start Event* with *EvalRec_FileSource,* drag mouse pointer from *Start Event* to *EvalRec_FileSource* (see Figure 8.27).

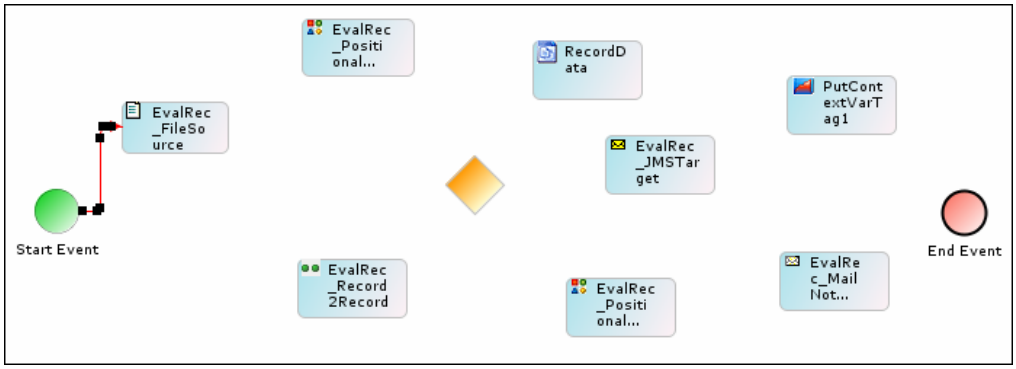



Figure 8.27: Connect Start Event to File Source Activity

40. Similarly, connect all other activities as shown in Figure 8.28.

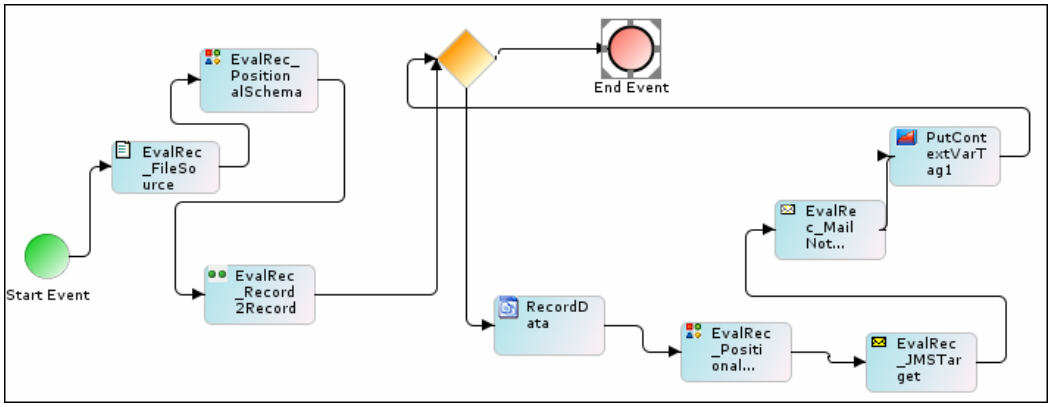



Figure 8.28: Connect all Activities

41. To define the decision criteria, right-click the Control Flow between *Gateway* and the *End Event*, and select **View Properties**. Properties of the selected control flow are shown (see Figure 8.29).

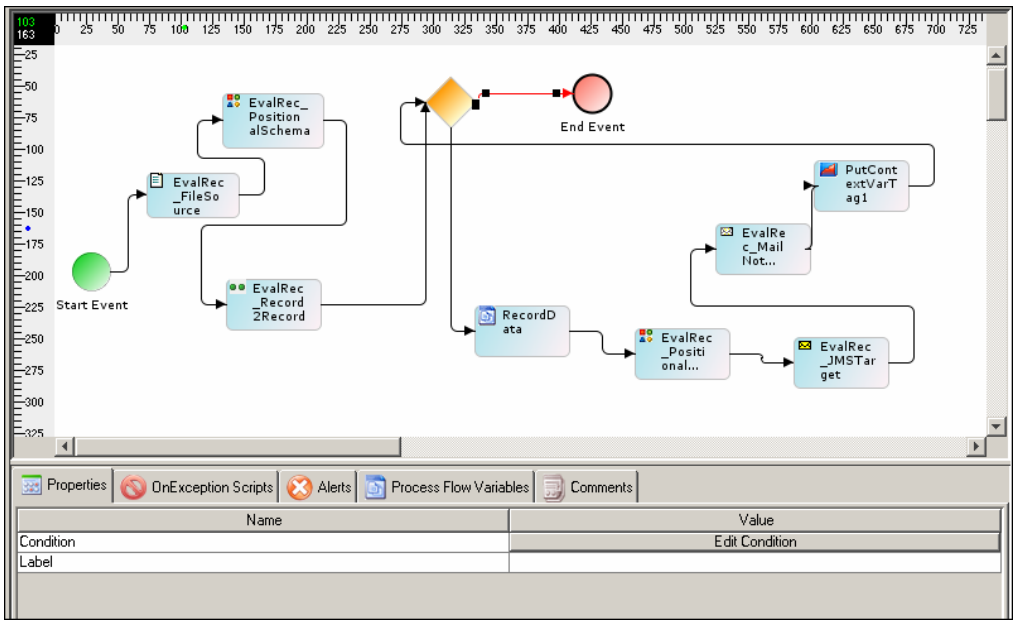



Figure 8.29: View Properties of Control Flow between Gateway Element and End Event

42. Click **Edit Condition**. The Condition Wizard is displayed (see Figure 8.30).

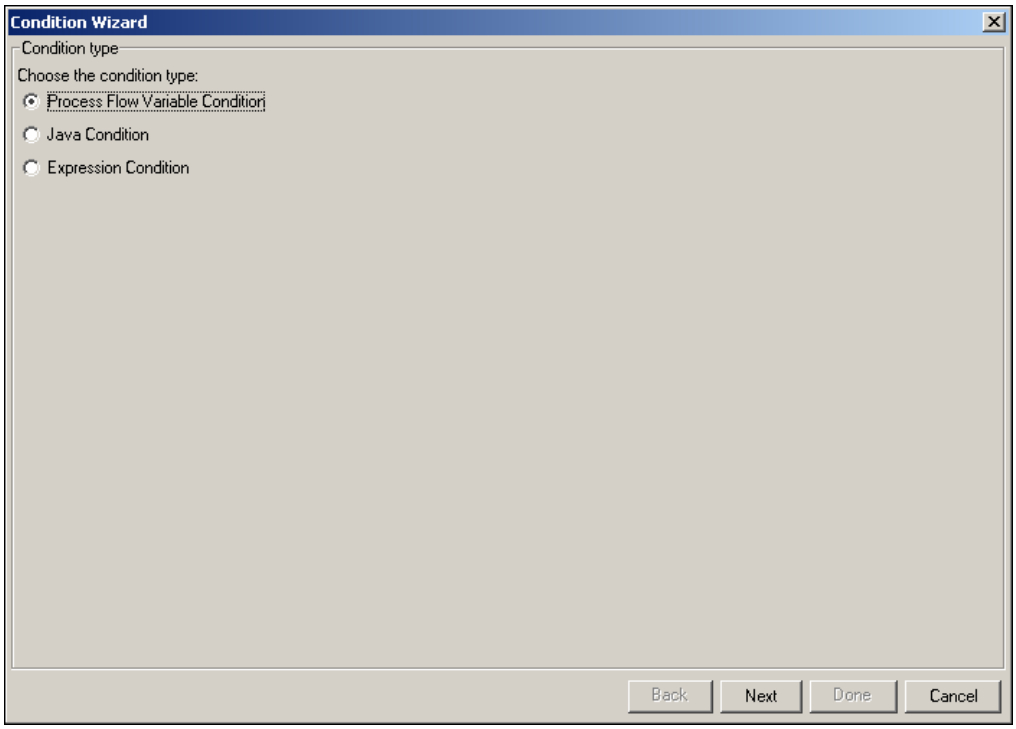



Figure 8.30: Condition Wizard

43. Select **Java Condition** and then click the **Next** button. The Java Condition box is displayed (see Figure 8.31).

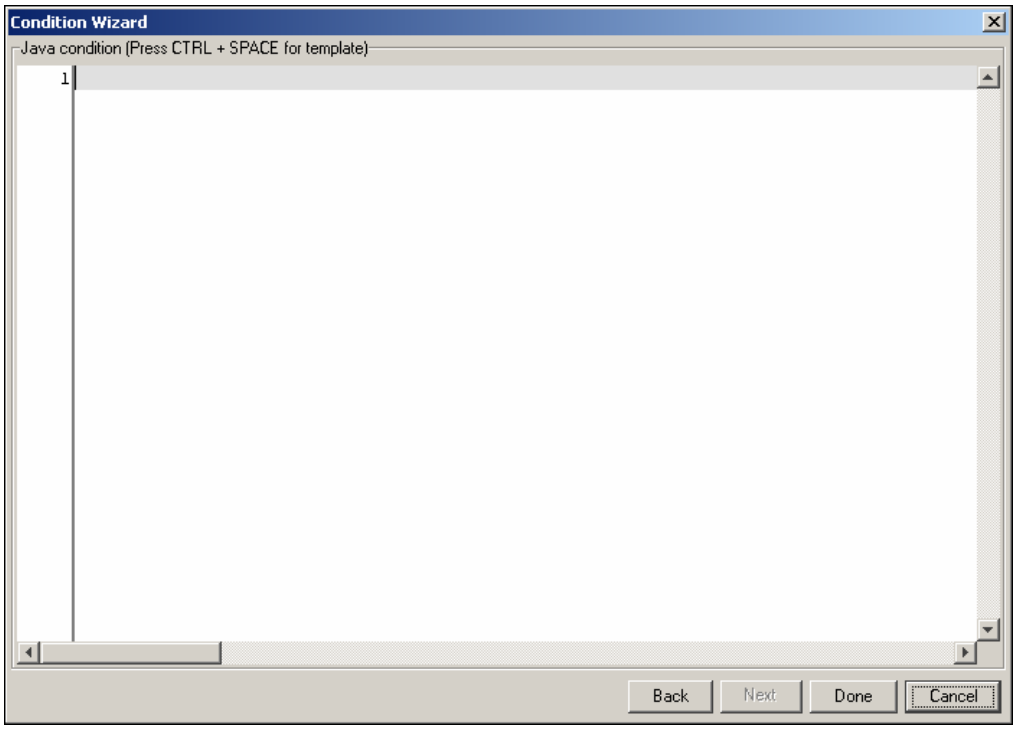



Figure 8.31: Enter JAVA Condition

44. Enter the sample Java Code in the Java Condition box (see Figure 8.32) and click the **Done** button.

```
//Getting Process Flow variable(nextRecord) value
String next= (String)context.get("nextRecord");
if( next.equals("yes") ){
      //wait for 10 seconds before end of Process Flow and recheck Process Flow variable(nextRecord) value
      Thread.sleep(10000);
        next= (String)context.get("nextRecord") ;
      if(next.equals("yes")){
         return true ;
        }
}
return false;
```

Figure 8.32: Enter JAVA Code

45. In Graph Canvas Area, right-click the BPMN Gateway element and select **Sequence Flow Ordering** option. The Sequence Flow Ordering dialog box is displayed (see Figure 8.33).

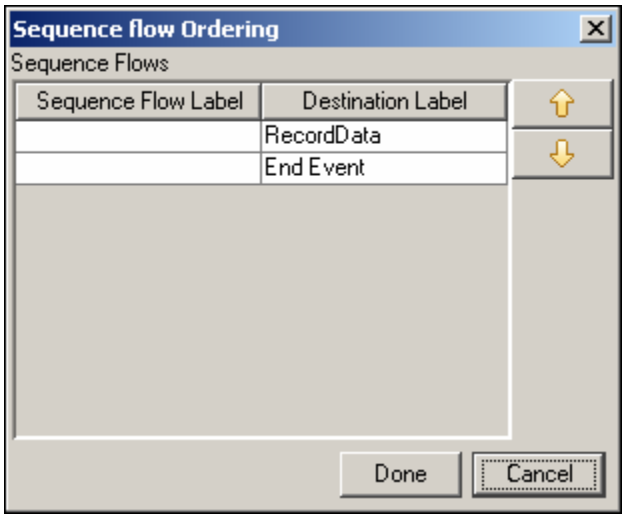



Figure 8.33: Sequence Flow Ordering

46. Select the control, which is connecting **Gateway** and **End Event** and click **UP Arrow** ( ).

47. Click the **Done** button to close the Sequence Flow Ordering Dialog box.

| | |
|---|---|
| | A default outgoing sequence flow is added for the Gateway. While executing a process flow, if none of the specified conditions are met, then the default gateway is executed. If there is no default gateway specified, then an error occurs at the Process Flow design level. |

48. Right click the **EvalRec_Record2Record** activity and select **Properties** option. Properties of *EvalRec_Record2Record* are shown in the Properties Panel of the bottom pane (see Figure 8.34).

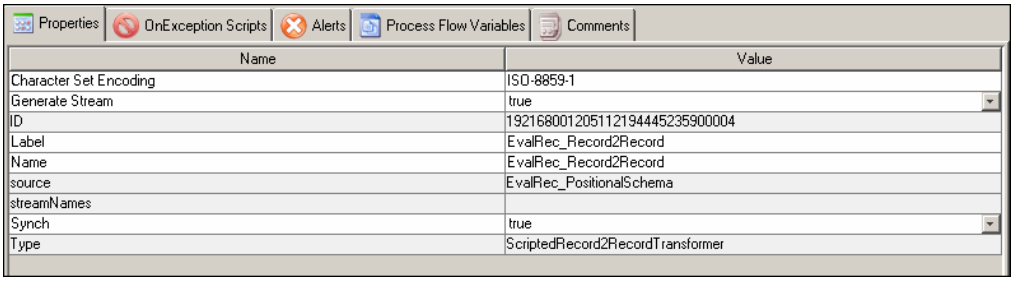



Figure 8.34: View Properties of *EvalRec_Record2Record* Activity

49. Change the value of property *Synch* from *true* to *false*.

50. Change the value of property *Generate Stream* from *true* to *false*.

51. Save the Process Flow by clicking the **File** menu and selecting **Save Process Flow to Server**. A dialog box is displayed confirming that the *EvalRec_ProcessFlow* has been saved successfully. If the *comments* property is enabled, then clicking **Save Process Flow to Server** will display a screen where you need to enter comments related to creating the process flow (refer to Figure 7.37).

52. Enter the comments in the *Specify Comments for process flow customer* field.

| | |
|---|---|
|  | The comment should be at least 1 character in length. |

53. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.
54. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

# 9 APPENDIX A: SETTING UP OPENJMS

This appendix describes how to install and configure OpenJMS server.

## INSTALLING OPENJMS

***Steps to download and install OpenJMS:***

1. Download the OpenJMS version 0.7.6.1 from any of the following URLs:
   - http://openjms.sourceforge.net/downloads.html
   - http://sourceforge.net/project/showfiles.php?group_id=54559
2. Unzip the downloaded file on the same machine where you have installed Adeptia Suite.
3. Go to *C:\openjms-0.7.6.1\config* folder and open *openjms.xml* file in a text editor.
4. Copy the following content (see Figure 9.1) in *openjms.xml* file, just below the Server Configuration tag.

```
  <RmiConfiguration embeddedRegistry="true"
                registryHost="localhost"
                registryPort="2099">
    </RmiConfiguration>


  <JndiConfiguration>
        <property name="java.naming.factory.initial"
                 value="com.sun.jndi.rmi.registry.RegistryContextFactory"/>
         <property name="java.naming.provider.url"
value="rmi//localhost:2099" />
 </JndiConfiguration>
```

Figure 9.1: Content to be copied in *openjms.xml* file

5. After copying the above content the *openjms.xml* file will look like as shown below (see Figure 9.2).

```
<!-- Optional. This represents the default configuration  -->
  <ServerConfiguration host="localhost" embeddedJNDI="true"   />

<RmiConfiguration embeddedRegistry="true"
                registryHost="localhost"
                registryPort="2099">
    </RmiConfiguration>


  <JndiConfiguration>
        <property name="java.naming.factory.initial"
                 value="com.sun.jndi.rmi.registry.RegistryContextFactory"/>
         <property name="java.naming.provider.url"
value="rmi//localhost:2099" />
```

```
</JndiConfiguration>
```

Figure 9.2: Content copied in *openjms.xml* file

6. Save the file and close it.
7. Set following parameter as System environment variable:

*JAVA_HOME= "C:\Program Files\adeptiajre\jre"*

*OPENJMS_HOME="C:\openjms-0.7.6.1"*

# STARTING OPENJMS

***Steps to start OpenJMS Server:***

1. To start the OpenJMS server, on command prompt type the following command:

   `CD %OPENJMS_HOME%\bin`

2. Now enter the command:
   `Startup`

3. OpenJMS server is started (see Figure 9.3).

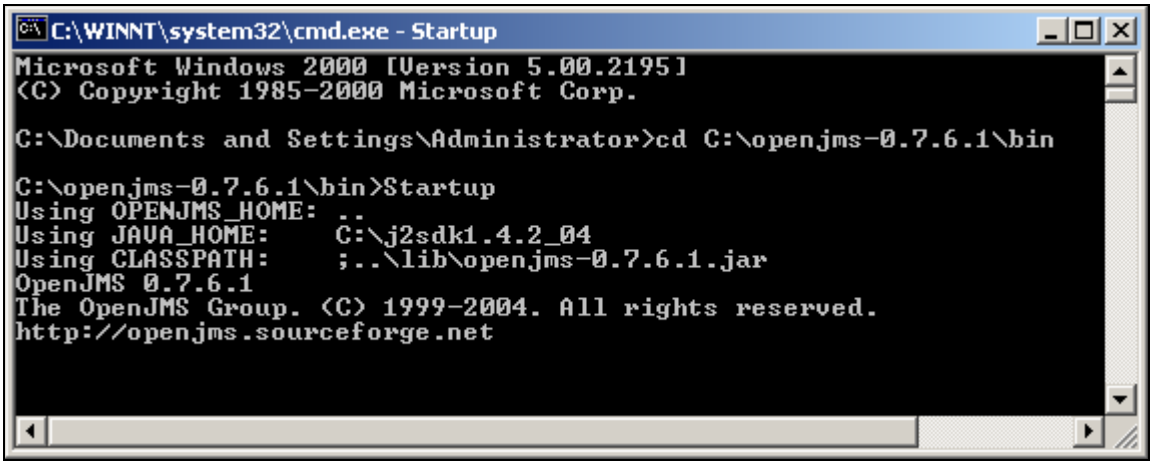



Figure 9.3: Start OpenJMS Server

To stop the OpenJMS Server, press <**Ctrl> +<C>**

# OPENING OPENJMS

***Pre-Requisites:***

OpenJMS server must be started before opening the Admin GUI.

***Steps to open OpenJMS Admin GUI:***

1. To open the OpenJMS admin screen, open another command prompt and type the following command:

   ```
   CD  %OPENJMS_HOME%\bin
   ```

2. Now enter the command:

   ```
   Admin
   ```

3. OpenJMS admin screen is displayed (see Figure 9.4).

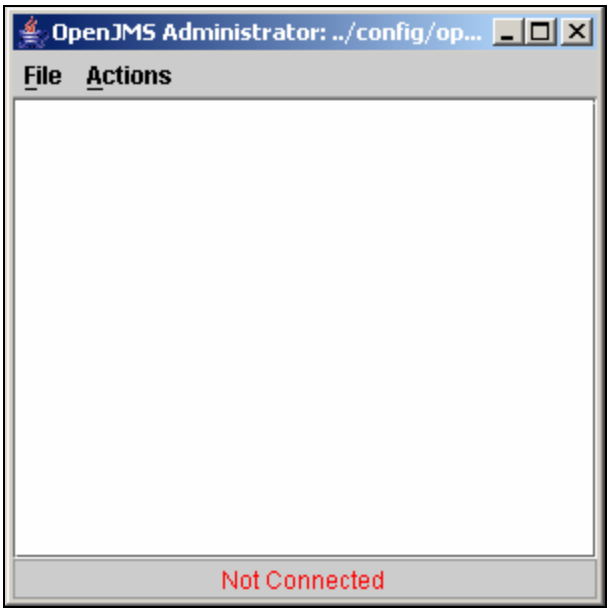



Figure 9.4: Start OpneJMS Admin GUI

4. Click **Actions** menu, go to **Connections** and then select **Online**. Default Queues and topics are displayed (see Figure 9.5).

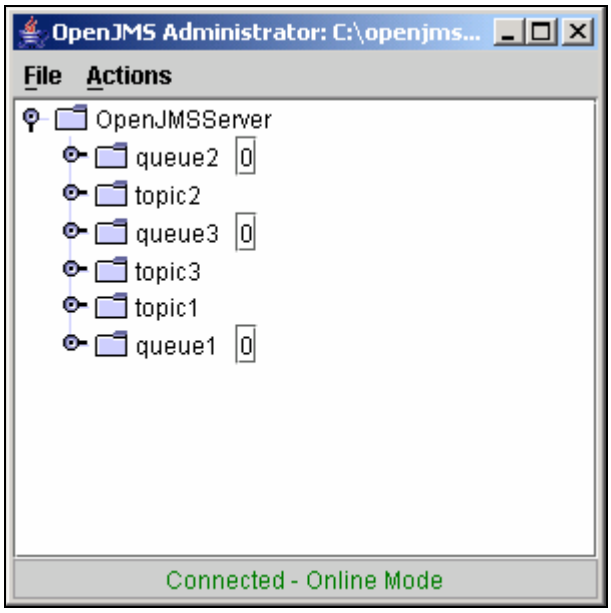



Figure 9.5: Connect to OpenJMS Server

5. To close the Admin screen, click **File** menu and select **Exit**.

# 10 ABOUT ADEPTIA INC.

Adeptia, an enterprise software company headquartered in Chicago, Illinois, provides business process integration technology to easily and quickly automate business processes using industry-specific standards. Adeptia's unique product combines business process management with business-to-business integration. Adeptia's reusable and highly scalable technology has been deployed by Fortune 1000 companies. For more information, visit **www.adeptia.com**.

Adeptia Headquarters
443 North Clark St, Suite 350
Chicago, IL 60654
USA
Email: info@adeptia.com

Adeptia India R&D Centre
D-74, Sector 63,
Noida, U.P. - 201301
India

# 11 TABLE OF FIGURES

# 12 TABLE OF TABLES