Adeptia Suite 6.0 Evaluation Guide

Version 1.0

Release Date March 25, 2013

Adeptia Inc.
443 North Clark Ave,
Suite 350
Chicago, IL 60610, USA
Phone: (312) 229-1727

DOCUMENT INFORMATION

Adeptia Suite

Evaluation Guide

Adeptia Suite Version 6.0

Printed February 2013

Printed in USA

Adeptia Support Information

For support queries, please contact us at *support@adeptia.com*.
Access the Adeptia Web site at the following URL:

www.adeptia.com

Copyright

Trademarks

Confidentiality

Disclaimer

TABLE OF CONTENTS

1

PREFACE

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.

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.

Pre-requisites

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

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**.

Other resource materials

The following other resource materials are available.

Title	Description
Installation Guide	This guide provided all the details of installing Adeptia Suite.
Getting Started Guide	This guide is intended as a reference for those working with Adeptia Suite for the first time.
Administration Guide	This guide provides a detailed description of the Administrative features of Adeptia Suite. It guides you to seamlessly manage the functioning, design and integration of business processes using these administrative features.
Business User Guide	This guide is intended for business users. They can use this guide to perform all the functions of Adeptia Suite.
Developer Guide	This guide covers a detailed description of all activities and services of Adeptia Suite that are available to a developer. It acts as a guideline to use these services seamlessly and use them in a design environment using Adeptia Suite.

HOW IS THIS GUIDE ORGANIZED?

This guide is organized into the following sections:

Section	Description
Preface	Introduction to this document
Data Transformation Flow	Use of Database Driver and Database Info, Database Schema, Database Source, Different Mapping function, Excel Schema and File Target
Scripted Service Process Flow	Use of File Source, Scripted Service, Text Schema, Mapping, Excel Schema, Compression, Put-Context-Var and File Target
Process Designer Process Flow	Use of File Source, Context Target, Context Source, Repeater, File target, Decision Node and Mail Target
Process Flow to Process Excel Data	Use of Mail Event, Mail Source, Excel Schema, Mapping, Database Schema, Database Target and Process Designer
JMS Event Driven Process Flow	Use of JMS Event, Context Source, Text Schema, Mapping, Database Schema, Database Driver, Database Info and Database Target
Record to Record Service Process Flow	Use of File Source, Positional Schema, Record to Record Service, JMS Target, Mail Notification and Process Flow Variable

CONVENTIONS

The following tables list the various conventions used in Adeptia documentation. We follow these conventions to help you quickly and easily identify particular elements, processes, and names that occur frequently in documents.

Abbreviations Used

This guide uses the following abbreviations:

Convention	Description
EBIM Suite	Enterprise Business Integration Management Suite

Convention	Description
BPM Suite	Business Process Management Suite
ESB Suite	Enterprise Service Bus Suite
ETL Suite	Extract, Transform and Load Suite
JMS	Java Messaging Service
CDO	Collaboration Data Object

Typographical conventions

This guide uses the following typographical conventions:

Convention	Description
Bold text	Indicates one of the following: ▪ Screen element ▪ New terminology ▪ A file or folder name ▪ A control in an application's user interface ▪ A registry key ▪ Important information
Italic text	Indicates a reference or the title of a publication.
`Monospaced text`	Indicates code examples or system messages.
`Monospaced bold text`	Indicates system commands that you enter.
Hyperlink	Indicates an Internet link to target material.

Graphical conventions

This guide uses the following graphical conventions:

Convention	Description

Indicates additional information that may be of interest to the reader.

Indicates cautions that, if ignored, can result in damage to software or hardware.

CONTACTS/REPORTING PROBLEMS

These sections present contact information for a variety of situations.

Sales

In case of any sales queries, please contact us at *sales@adeptia.com*.

Support

For support queries, please contact us at *support@adeptia.com*.

Latest updates and information

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

Adeptia Web site

Access the Adeptia Web site at the following URL:
www.adeptia.com

2

DATA TRANSFORMATION FLOW

This section describes the Data Transformation Process Flow.

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

DESCRIPTION

This sample Process Flow can be outlined as below:

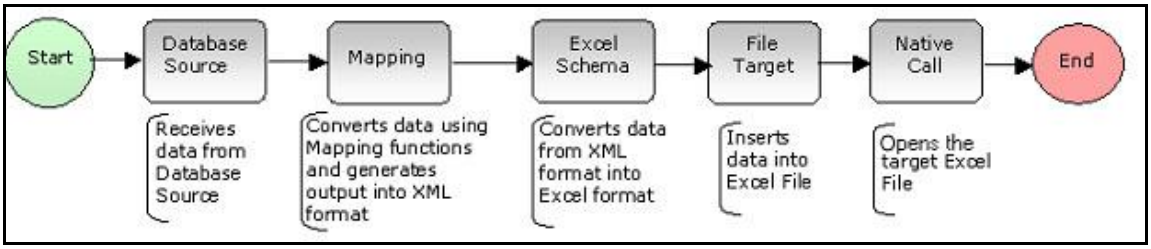

Figure 1: Data Transformation Process Flow

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 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

Field Name	Description	Data Type
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 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

1. 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.

2. 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-6.0/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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Process > Process Flow. The *Manage Process Flow* screen is displayed (see Figure 2).

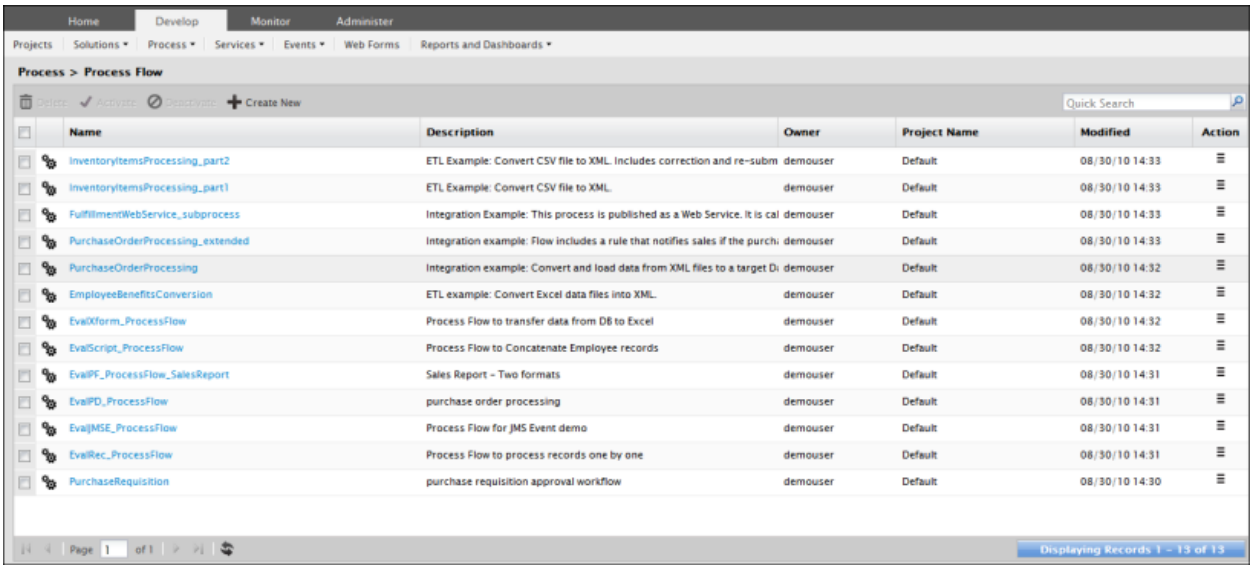

Figure 2: Process Flow Manage Page

3. Click the *Execute* button, which is adjacent to *EvalXform_ProcessFlow*. A confirmation dialog box is displayed (see Figure 3).

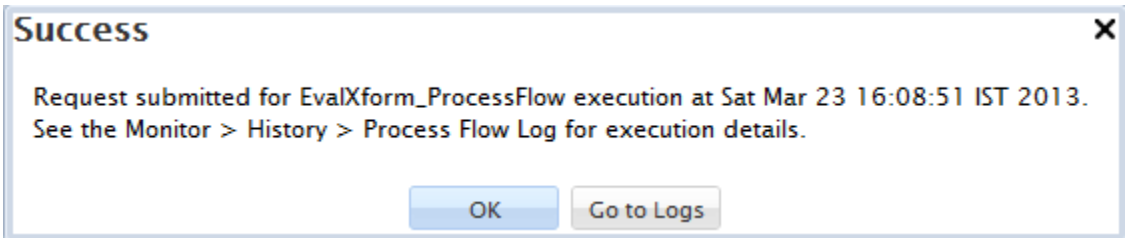

Figure 3: View Process Log

4. Click the button **Go to Logs** to view the status of the Process Flow execution. The Process Flow Logs are displayed (see Figure 4).

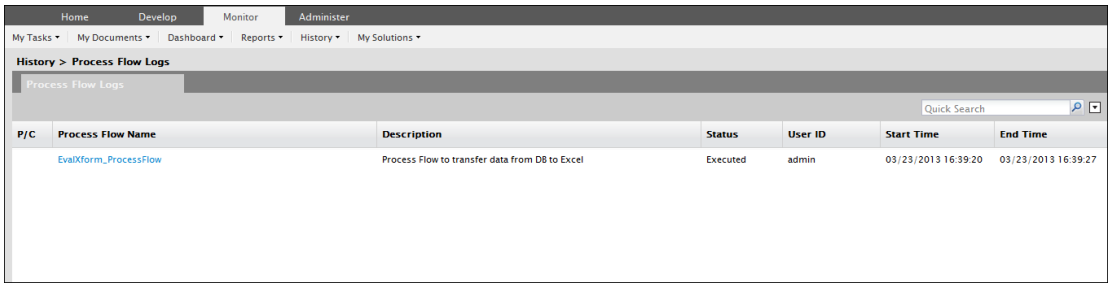

Figure 4: Process Flow Logs

5. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 5).

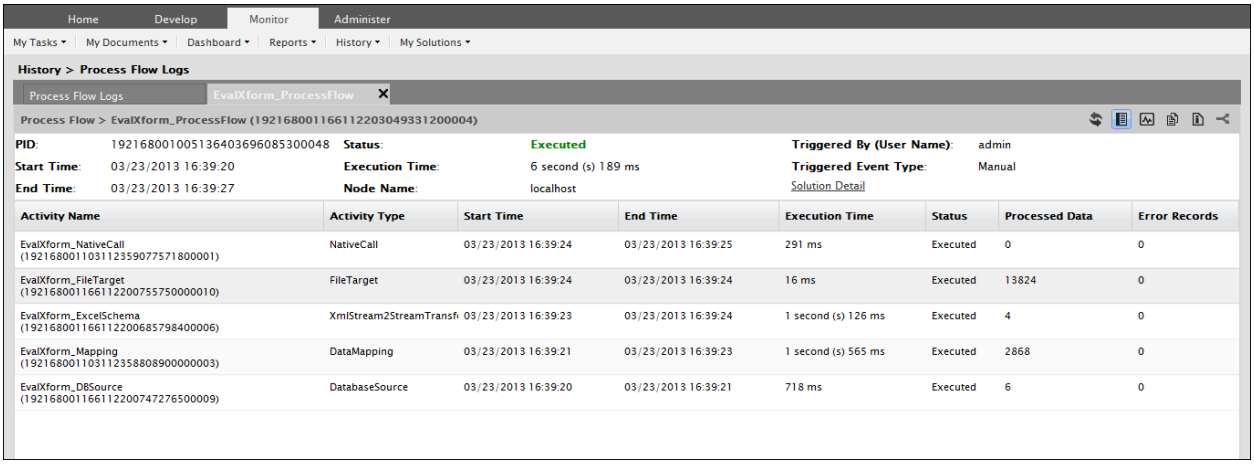

Figure 5: Process Flow Log Details

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-6.0/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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **Database Driver**. The *Manage Database Driver* screen is displayed with the list of existing database drivers (see Figure 6).

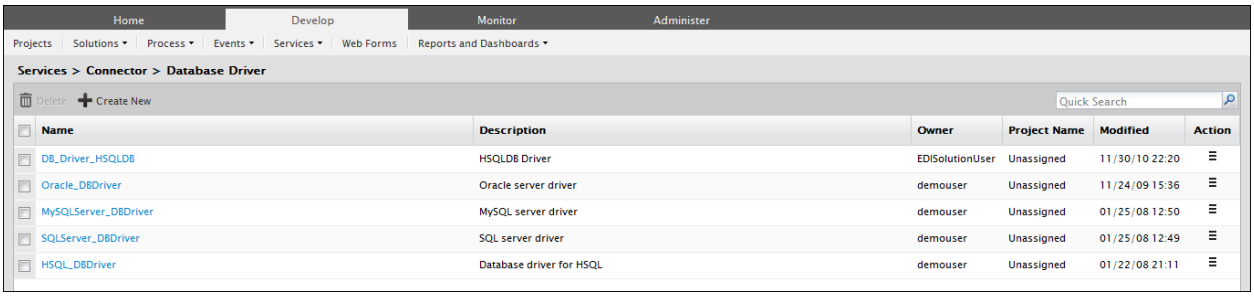

Figure 6: Manage Database Driver

4. Select the required Database Driver activity and right-click it to view the **More Actions** menu.
5. Click **Edit** link. This opens the respective Database Driver activity in the edit mode (see Figure 7).

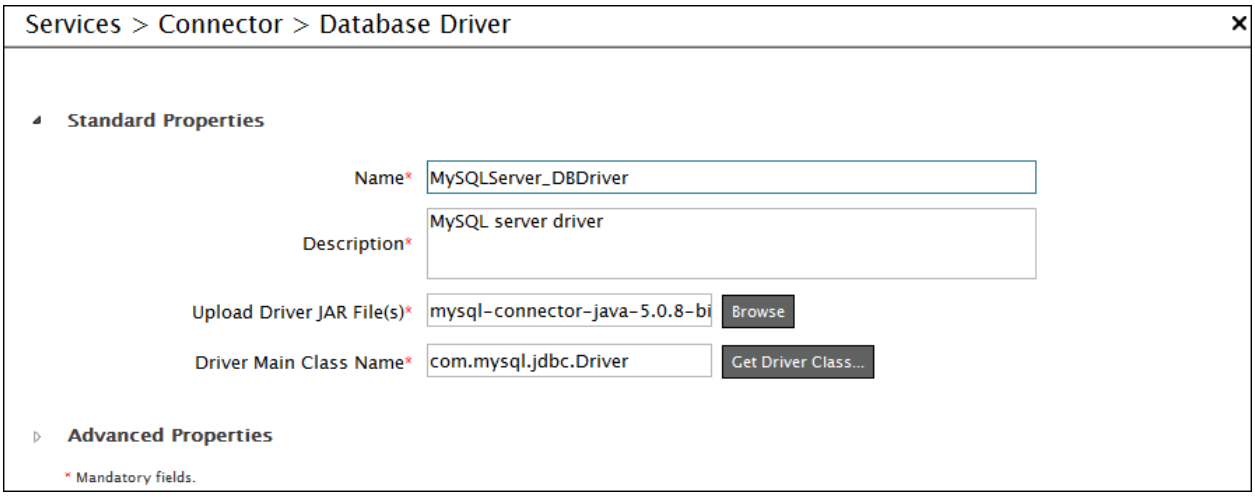

Figure 7: Edit *EvalXForm_DBDriver*

The following tables define the fields on this screen.

Table 3: Details of Fields on Edit Database Driver Screen

Field Name	Description

Field Name	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: Oracle: OJDBC5.jar MS SQL:sqljdbc4.jar JTDS- SQL Server: Jtds.jar HSQL DB: hsqldb-1.8.0.7.jar MySQL: mysql-connector-java-5.0.8-bin.jar IBM DB2 (Ver 7.1): db2java.zip (7.1 version) IBM DB2 (Ver 8.1): db2jcc.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 dropdown list. Following is the list of Driver Main Class Name of different databases: Oracle: oracle.jdbc.driver.OracleDriver 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 IBM DB2 (Ver 7.1): COM.ibm.db2.jdbc.net.DB2Driver IBM DB2 (Ver 8.1): com.ibm.db2.jcc.DB2Driver

6. Make the necessary changes.

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

The comment should be at least 1 character in length.

9. 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 the specified Database Server.

Steps to edit the Database Information

1. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **Database Info**. The *Manage Database Info* screen is displayed with the list of existing Database Info (see Figure 8).

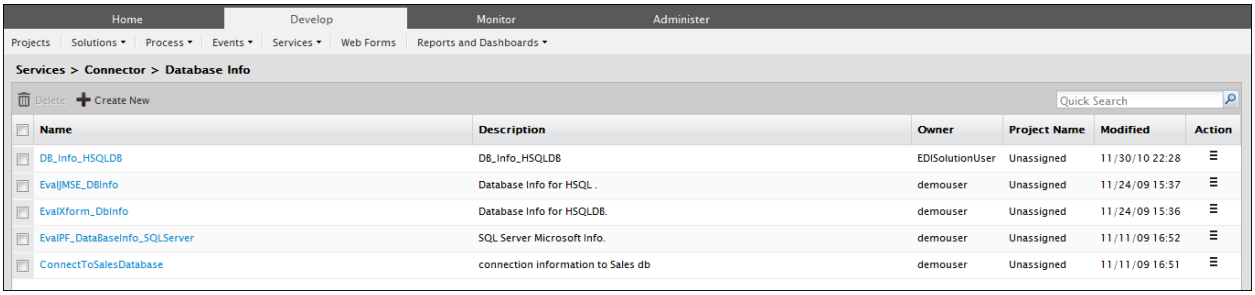

Figure 8: Manage Database Info

4. Select the required Database Info activity and right-click the same to view the **More Actions** menu.
5. Click **Edit** to open the respective Database Info activity in edit mode (see Figure 9).

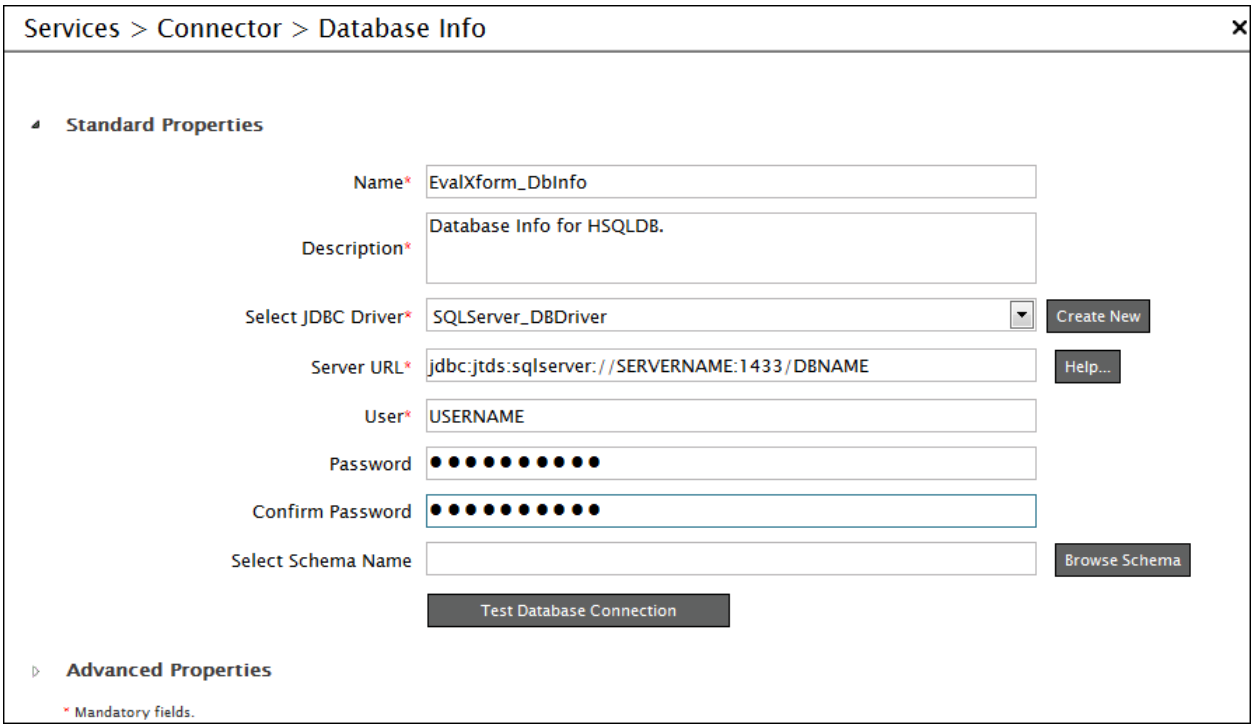

Figure 9: Edit *EvalXForm_DBInfo*

The following tables define the fields on this screen.

Table 4: Details of Fields on Edit Database Info Screen

Field Name	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	Following is the list of Server URL's of different databases: Oracle: jdbc:oracle:thin:@<databaseServerName>:<port>:<databaseName> MS SQL: jdbc:sqlserver://<databaseServerName>:<port>/databaseName SQL JTDS: jdbc:jtds:sqlserver://<databaseServerName>:<port>/<databaseName> MS Access: jdbc:odbc:Driver={MicroSoft Access Driver (*.mdb)}; DBQ=<Path of .mdb file> MS Excel: Jdbc:odbc:ExcelJDBCTest where ExcelJDBCTest is the ODBC object that you need to create using DSN. HSQL DB: jdbc:hsqldb:hsql://<databaseServerName>:<port>/<databaseName> IBM DB2 (Ver 7.1): jdbc:db2://<databaseServerName>:<port>/databaseName IBM DB2 (Ver 8.1): jdbc:db2://<databaseServerName>:<port>/databaseName Here: <databaseServerName> is the name/IP address of the machine where database server is running. <port> is the port at which database server is running. <databaseName> is the name of the database to which you want to connect.

6. Make the necessary changes.
7. 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.
8. Enter comments in the **Add Comments** field.

> The comment should be at least 1 character in length.

9. 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Schema** and then click **Database**.

   The *Manage Database Schema* screen is displayed (see Figure 10).

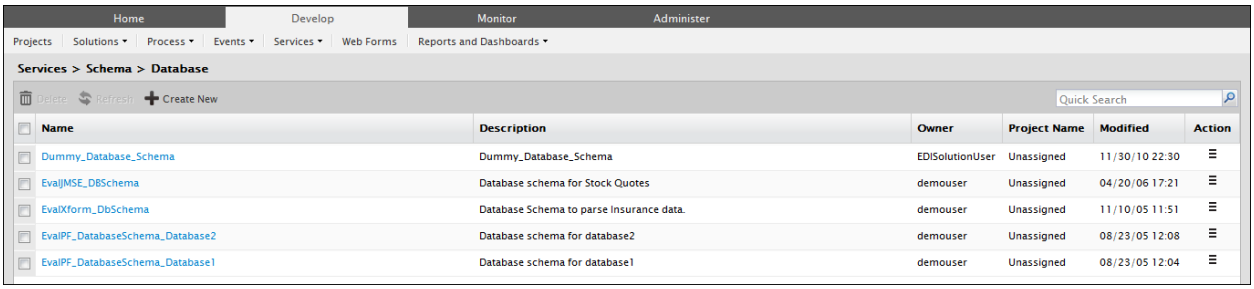

Figure 10: Manage Database Schema

3. Select the required database activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Database schema activity in edit mode (see Figure 11).

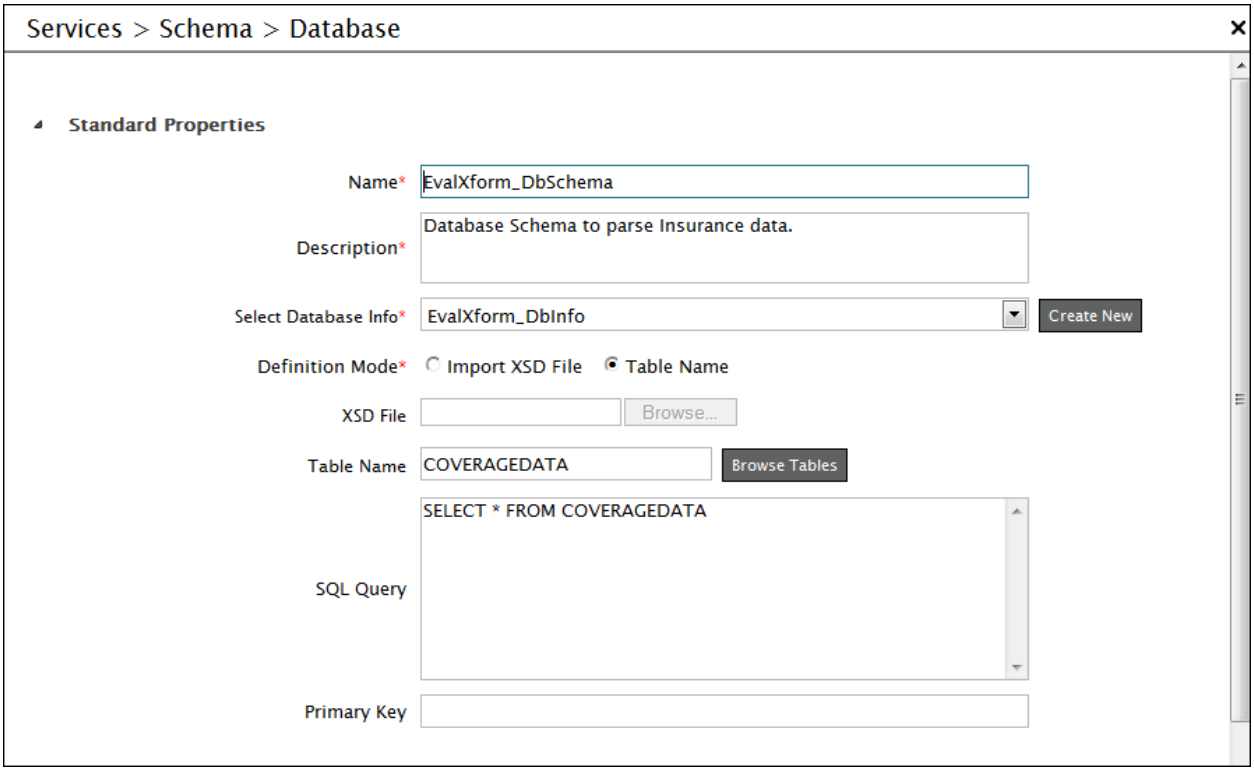

Figure 11: Edit *EvalXForm_DBSchema* Activity

The following table defines the fields on this screen.

Table 5: Details of Fields on Edit Database Schema Screen

Field Name	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.

Field Name	Description
Create Schema Definition	Schema Definition can be created using one of the following options: • Use XSD File • Table Name 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. 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.
7. Enter 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 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Source and then click Database.

   The *Manage Database Source* screen is displayed (see Figure 12).

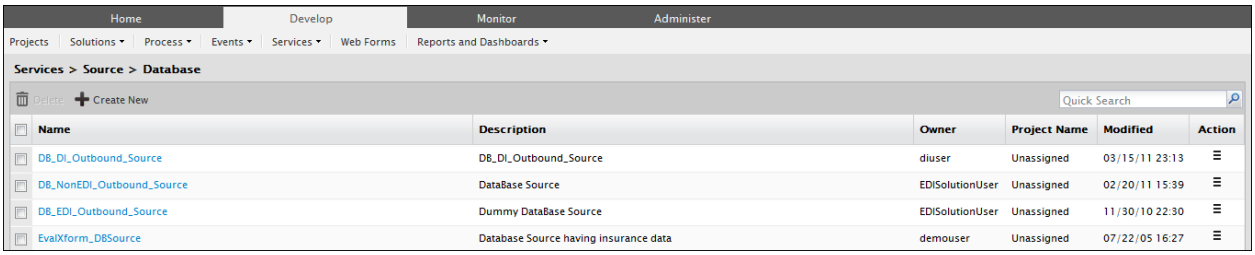

Figure 12: Manage Database Source

3. Select the required database activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Database source activity in edit mode (see Figure 13).

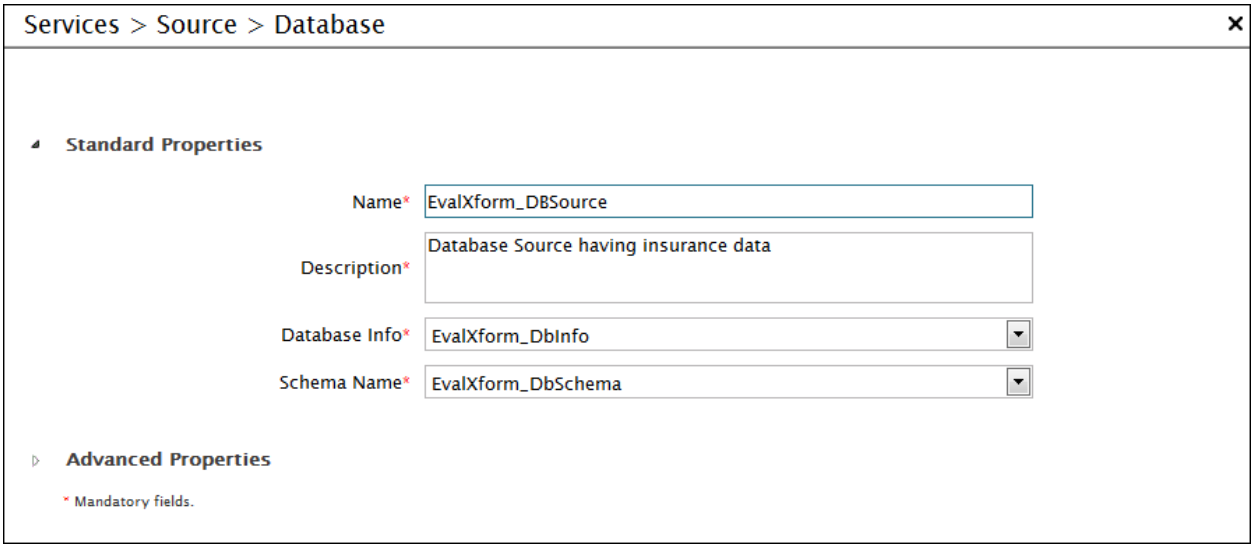

Figure 13: Edit *EvalXForm_DBSource* Activity

The following tables defines the fields on this screen.

Table 6: Details of Fields on Edit Database Source Screen

Field Name	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.
7. Enter 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation and then click Data Mapping.

   The *Manage Data Mapping* screen is displayed with the list of existing mapping activities (see Figure 14).

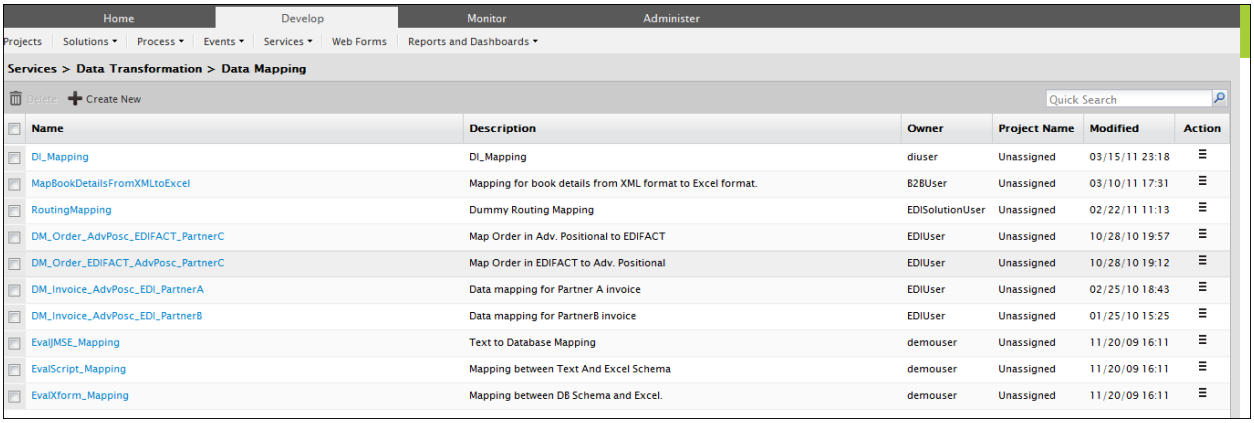

Figure 14: Manage Data Mapping

3. Select the required Data Mapping activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Data Mapping activity in edit mode (see Figure 15).

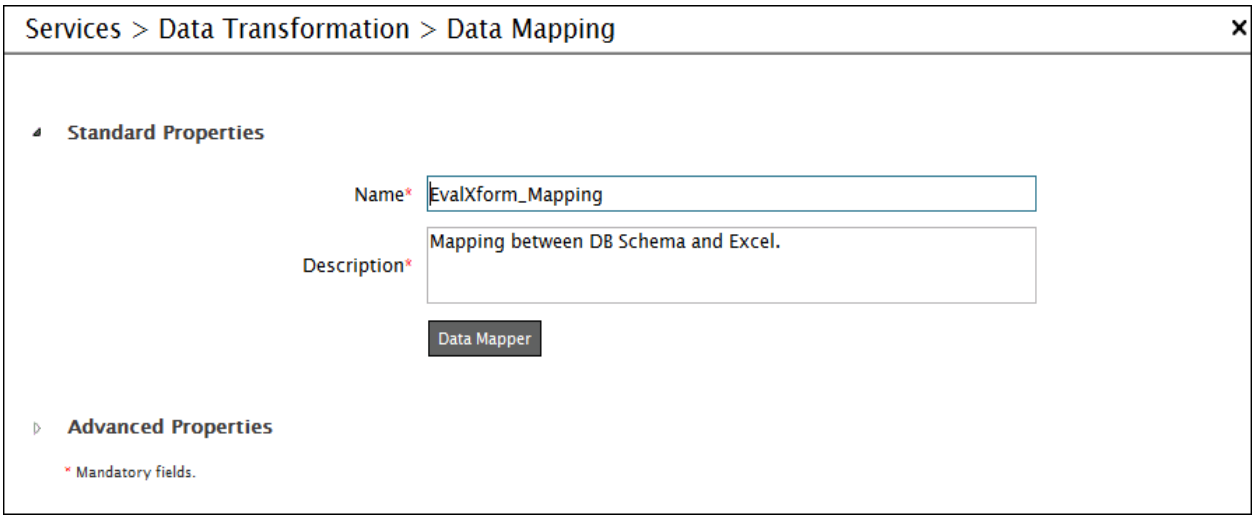

Figure 15: Edit *EvalXForm_Mapping* Activity

5. Click **Data Mapper** button**.** The Data Mapper applet is displayed showing mapping between data fields of source and target schema (see Figure 16).

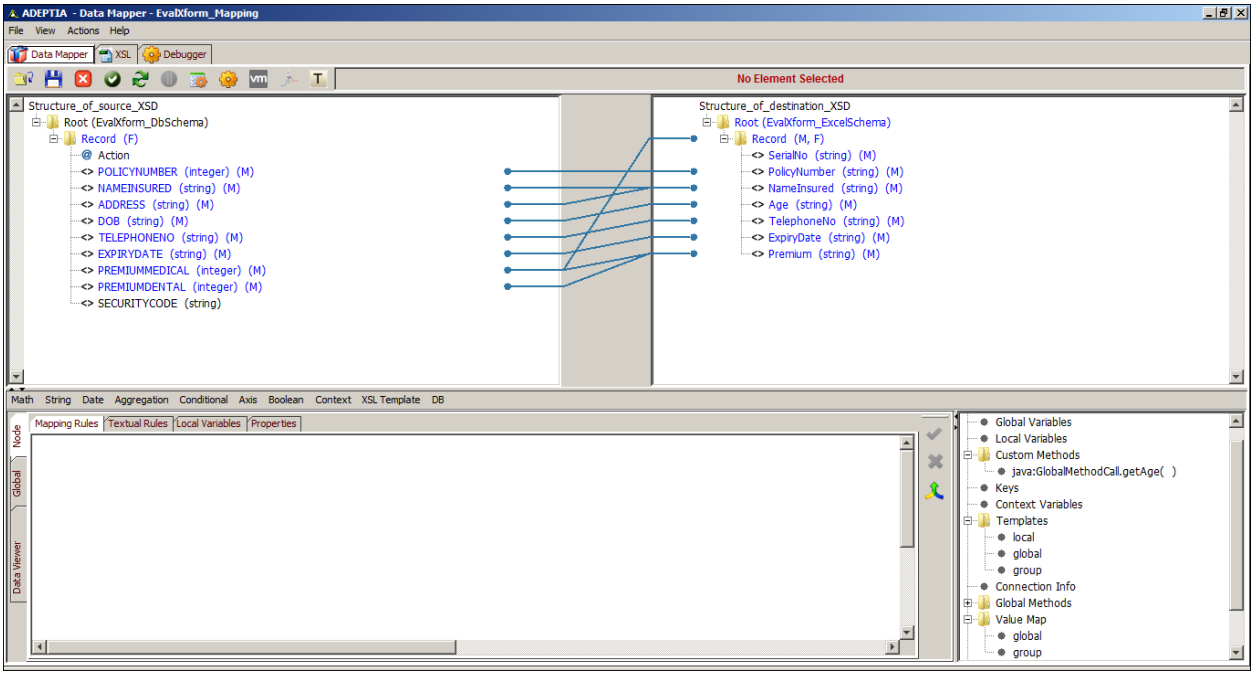

Figure 16: *EvalXForm_Mapping* Activity in the Data Mapper Applet

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 17).

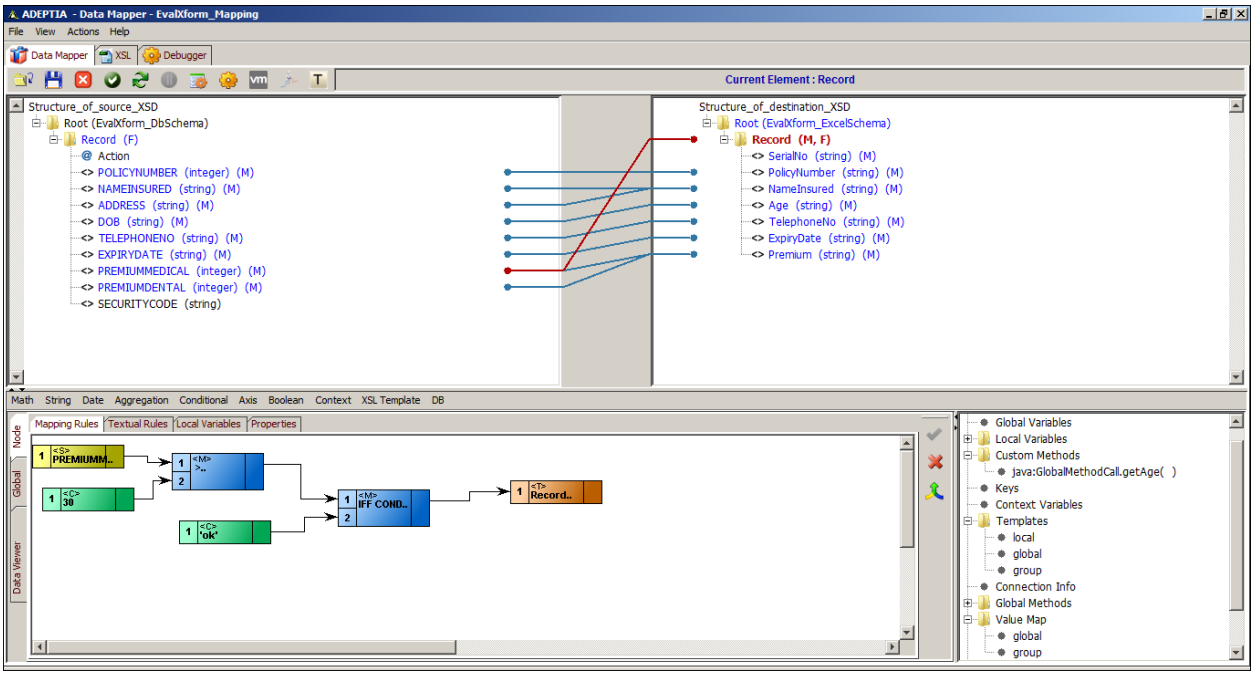

Figure 17: View Mapping Functions used in *EvalXForm_Mapping* Activity

7. To analyze the impact of mapping function on output records, right-click any of function node (*IFF Condition*) and select **Information** (see Figure 18).

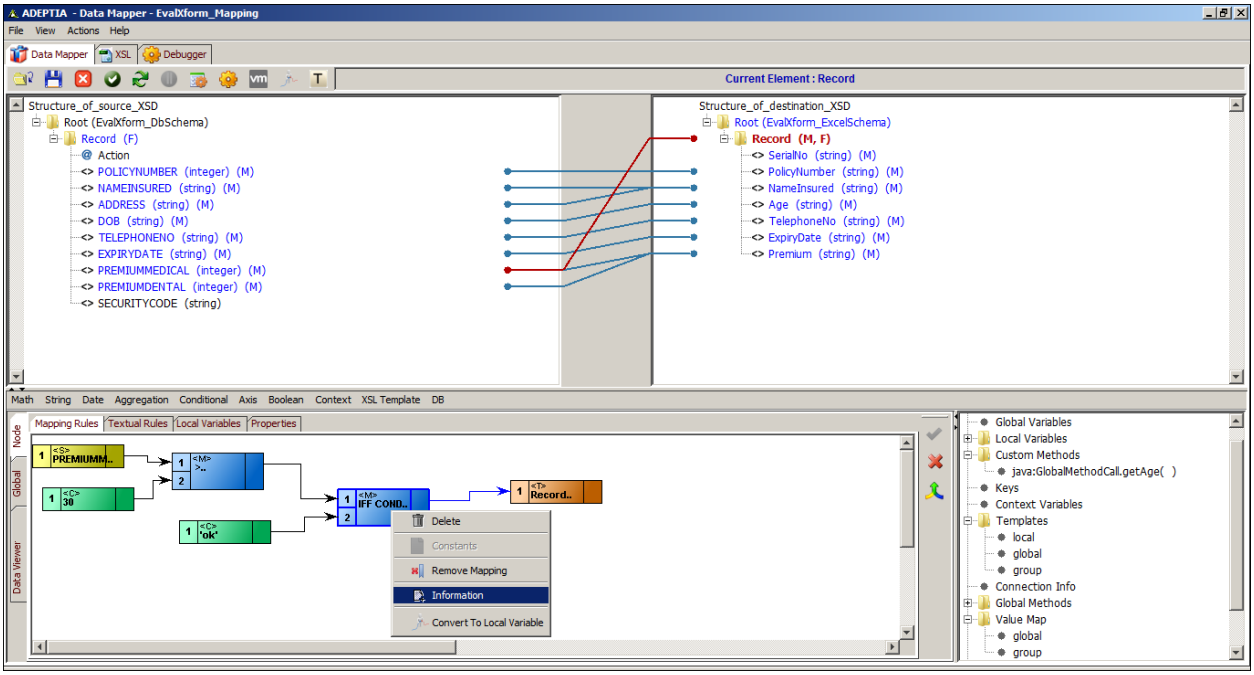

Figure 18: Select Mapping Function Information

8. A dialog box is displayed that shows the information about that mapping function (see Figure 19).

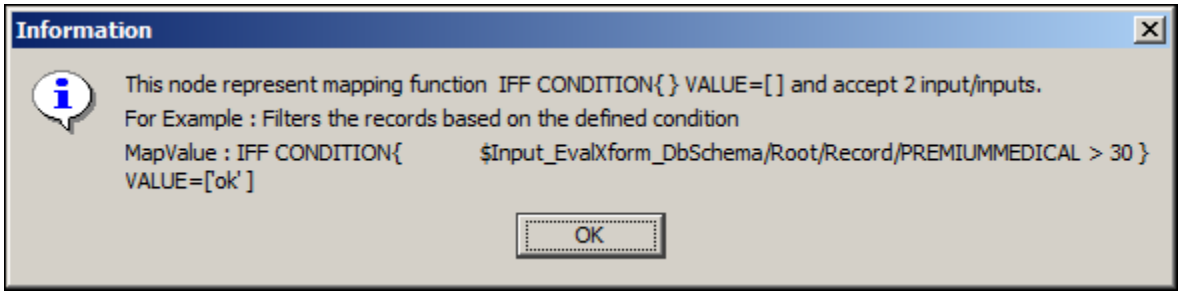

Figure 19: View Mapping Function Information

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.
13. Enter the comments in the Specify comments for mapping object (object name) field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to > **Services** > **Schema**, and then click **Excel**. The *Manage Excel Schema* screen is displayed with the list of existing Excel Schemas (see **Error! Reference source not found.**).

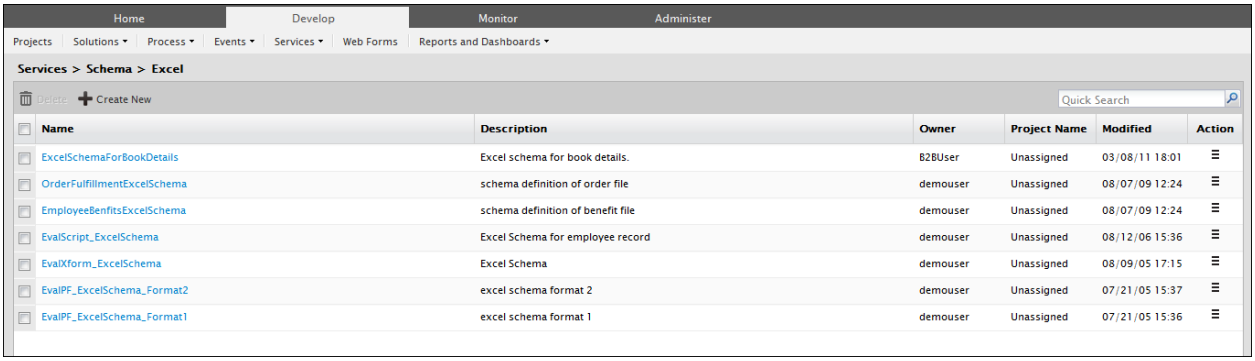

Figure 20: Manage Excel Schema

3. Select the required Excel schema activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Excel schema activity in edit mode (see Figure 21).

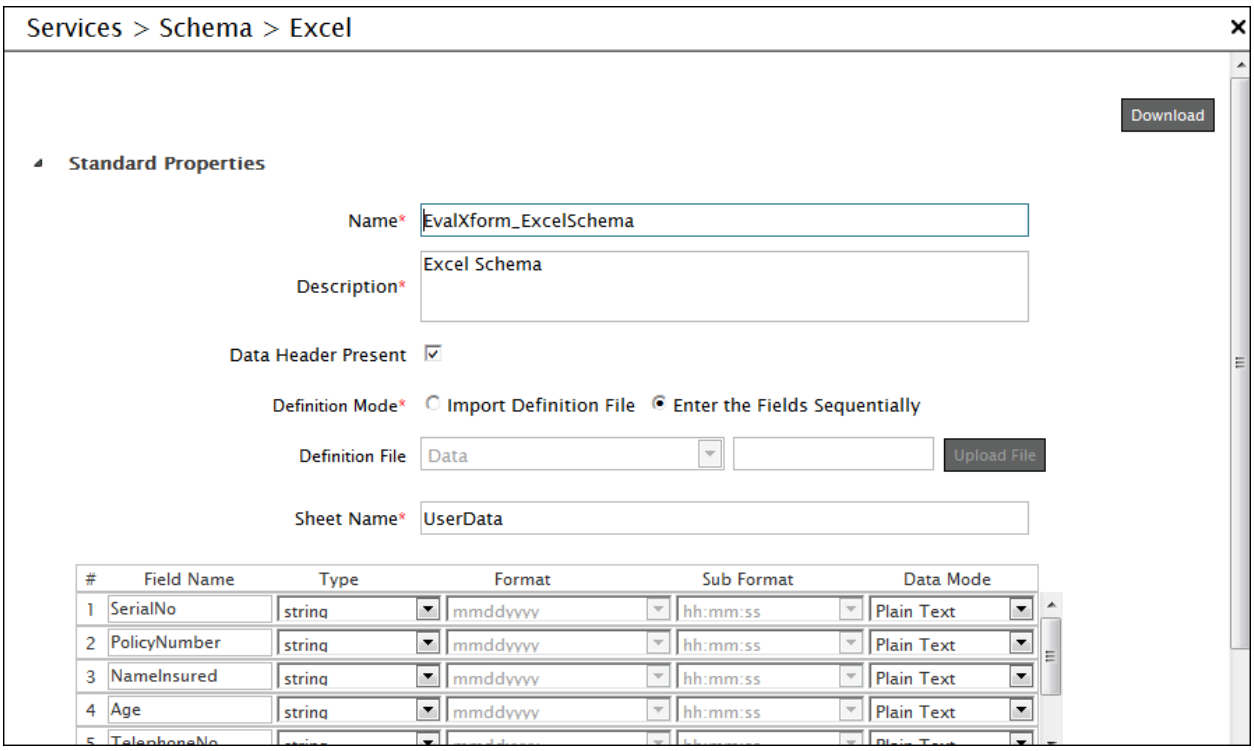

Figure 21: Edit *EvalXForm_ExcelSchema* Activity

The following table defines the fields on this screen.

Table 7: Details of Fields on Edit Excel Schema Screen

Field Name	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: • Use Definition File • Enter the Field Sequentially Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially

Field Name	Description
Field Name	Name of the Fields
Type	There are following 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* dropdown list.
SubFormat	If data type is *Date*, select time format from *SubFormat* dropdown 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.
7. Enter 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.

Testing Excel Schema (EvalXForm_ExcelSchema)

You can verify the excel schema activity at design time.

Steps to verify schema activity

1. Click the **Test** button on the *Edit Excel Schema* screen. The *Test Schema* screen is displayed (refer to Figure 178).
2. Select the type of schema to test, from the *Type* dropdown list. By default, *Source* is selected. Since this Excel schema is used at target end, select Target from *Type* dropdown 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-6.0/Sample Datafiles/EvalXform** directory.

Steps to edit the File Target

1. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **File**.

   The *Manage File Target* screen is displayed (see Figure 22).

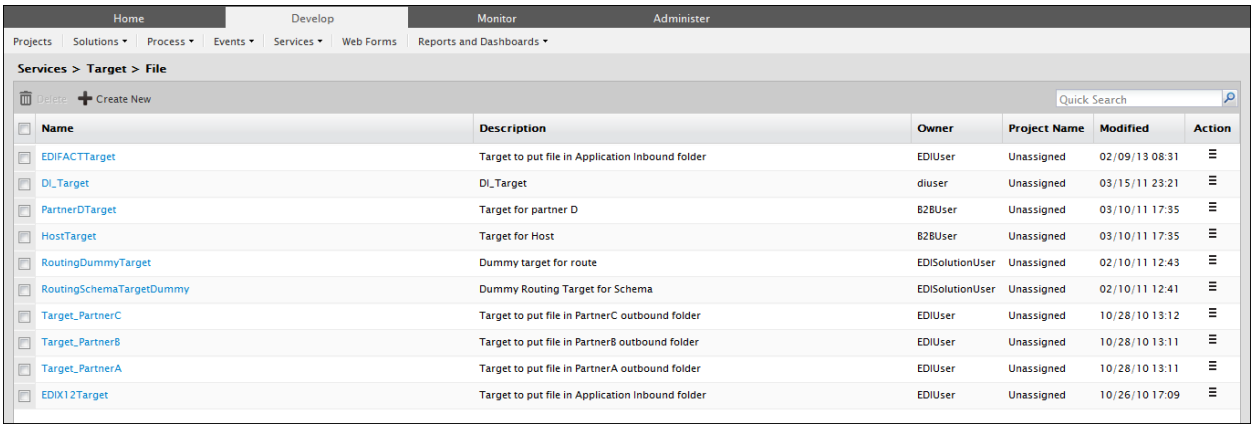

Figure 22: Manage File Target

3. Select the required File target activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective File target activity in edit mode (see Figure 23).

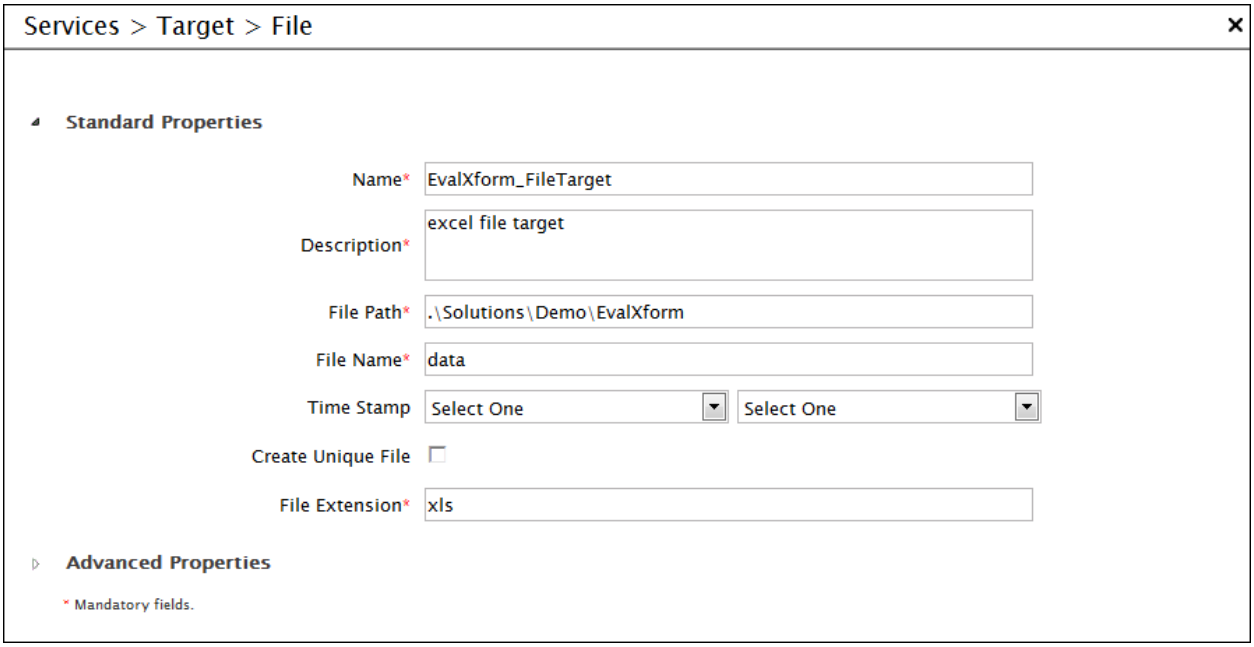

Figure 23: Edit *EvalXForm_FileTarget* Activity

The following table defines the fields on this screen.

Table 8: Details of Fields on Edit File Target Screen

Field Name	Description
Name	Name of the File Target
Description	Description of the File Target
File Path	Path of the target file. For example: ../../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 activity.
7. Enter 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Extensions** and then click **Native Call**. The *Manage Native Call* screen is displayed with the list of existing Native Calls (see Figure 24).

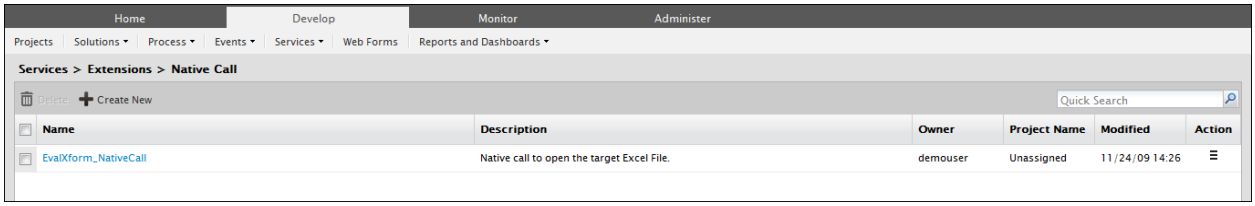

Figure 24: Manage Native Call

3. Select the required Native Call activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Native Call activity in edit mode (see Figure 25).

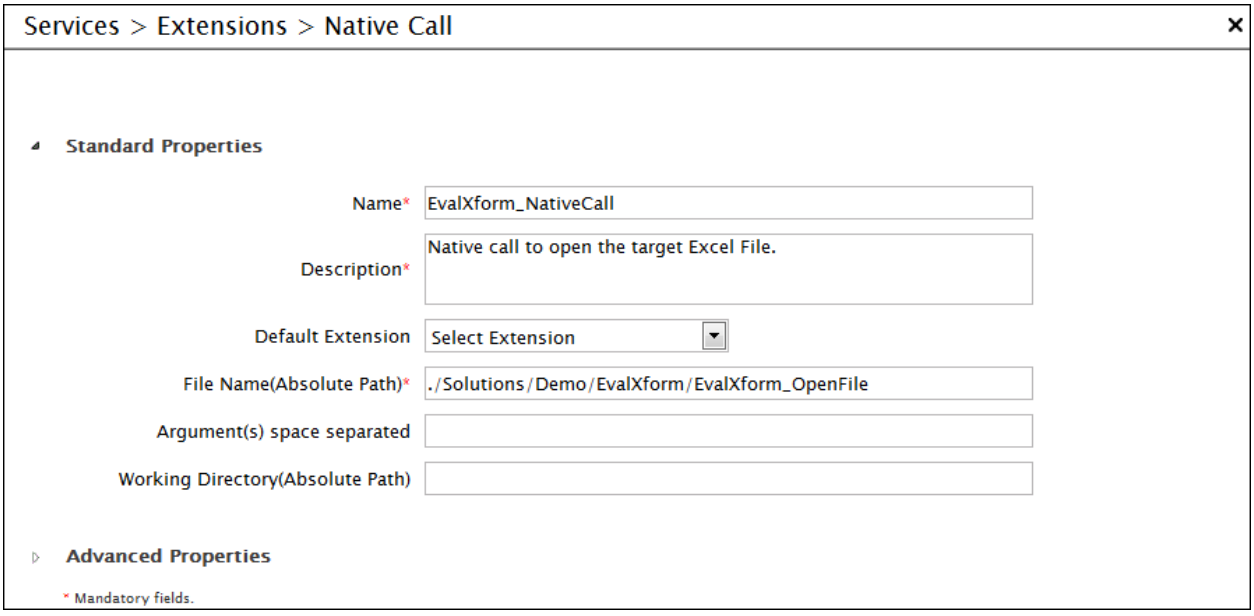

Figure 25: Edit *EvalXForm_NativeCall* Activity

The following tables defines the fields on this screen.

Table 9: Details of Fields on Edit Native Call Screen

Field Name	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 For example: ../../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.
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation and then click Data Mapping.

   The *Manage Data Mapping* screen is displayed with the list of existing activities (see Figure 26).

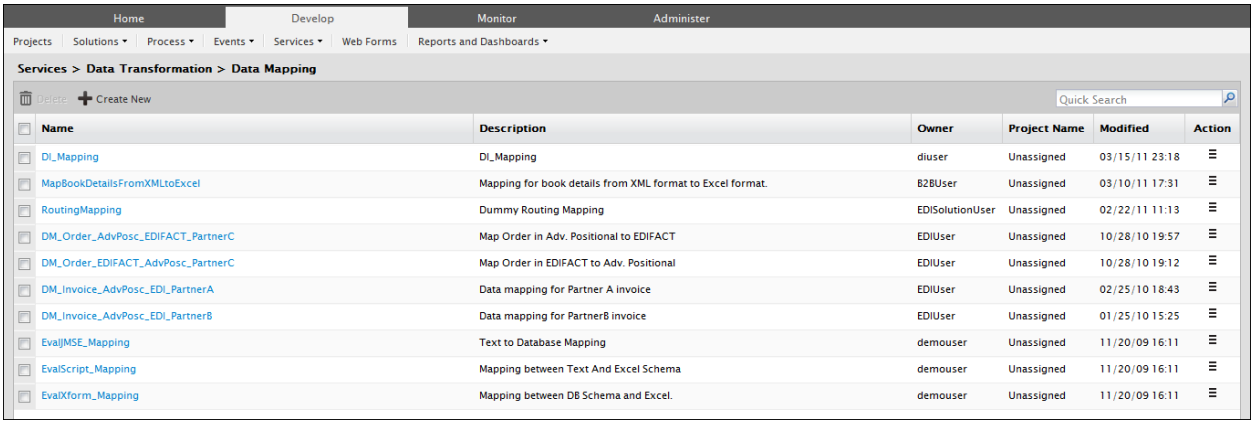

Figure 26: Manage Data Mapping

3. Click the **Create New** link. The *Create Data Mapping* screen is displayed (see Figure 27).

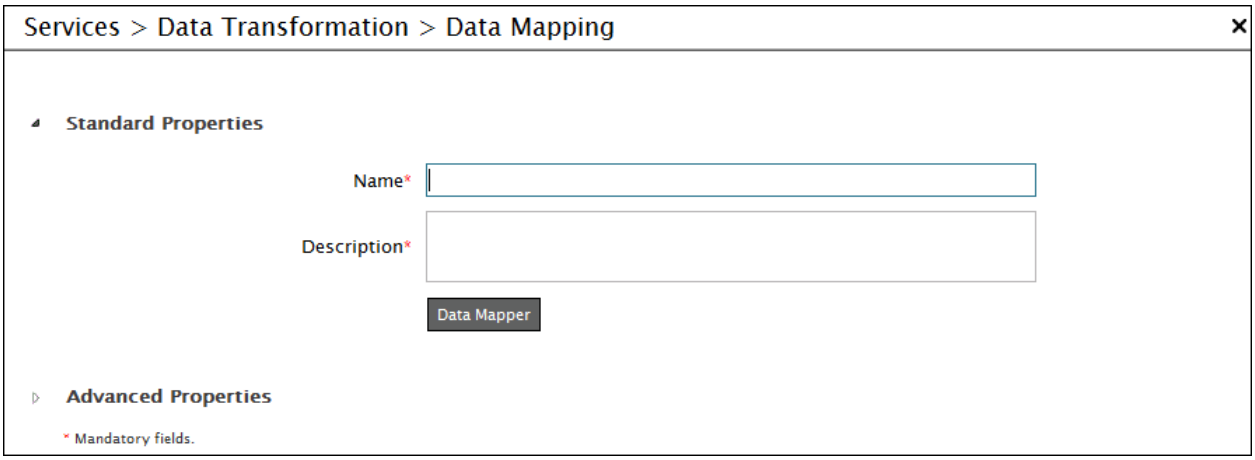

Figure 27: Create Data Mapping

4. Enter the name and description of the new mapping activity in the textboxes **Name** and **Description** respectively.

5. Click the **Data Mapper** button. This displays the Data Mapper applet (see Figure 28)

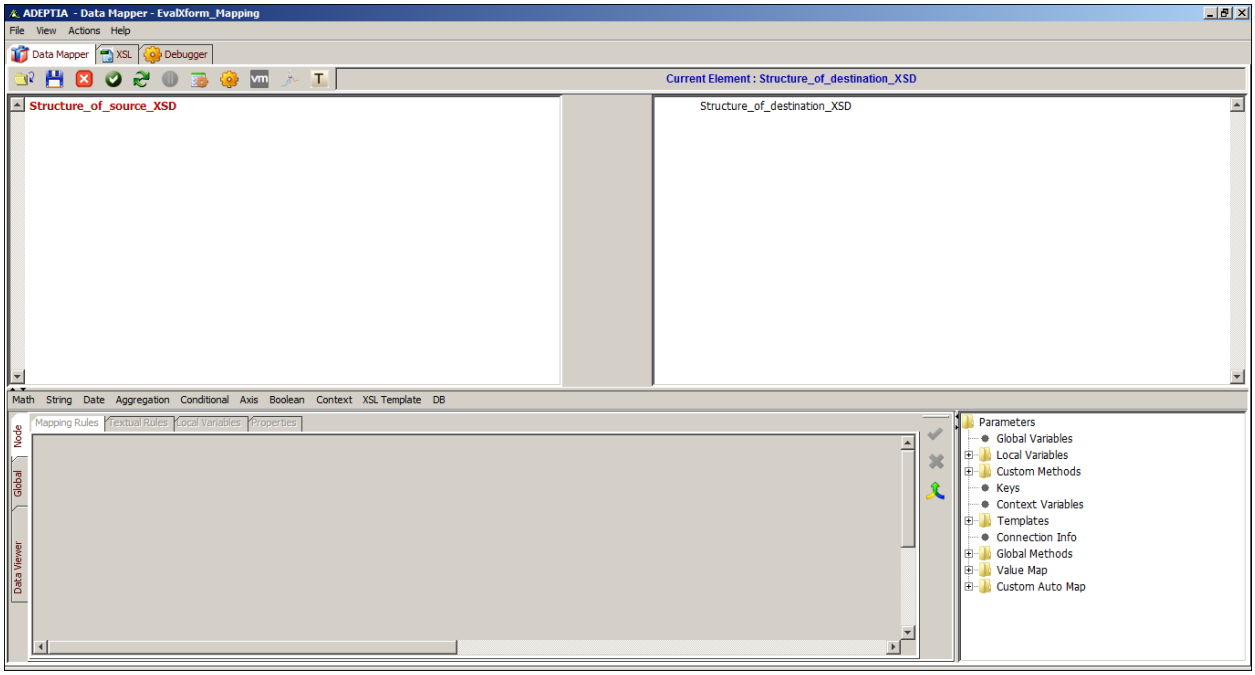

Figure 28: Data Mapper Applet

6. Click **Open Source Schema** button to load the source schema. This displays the **Select Schema** screen (see Figure 29).

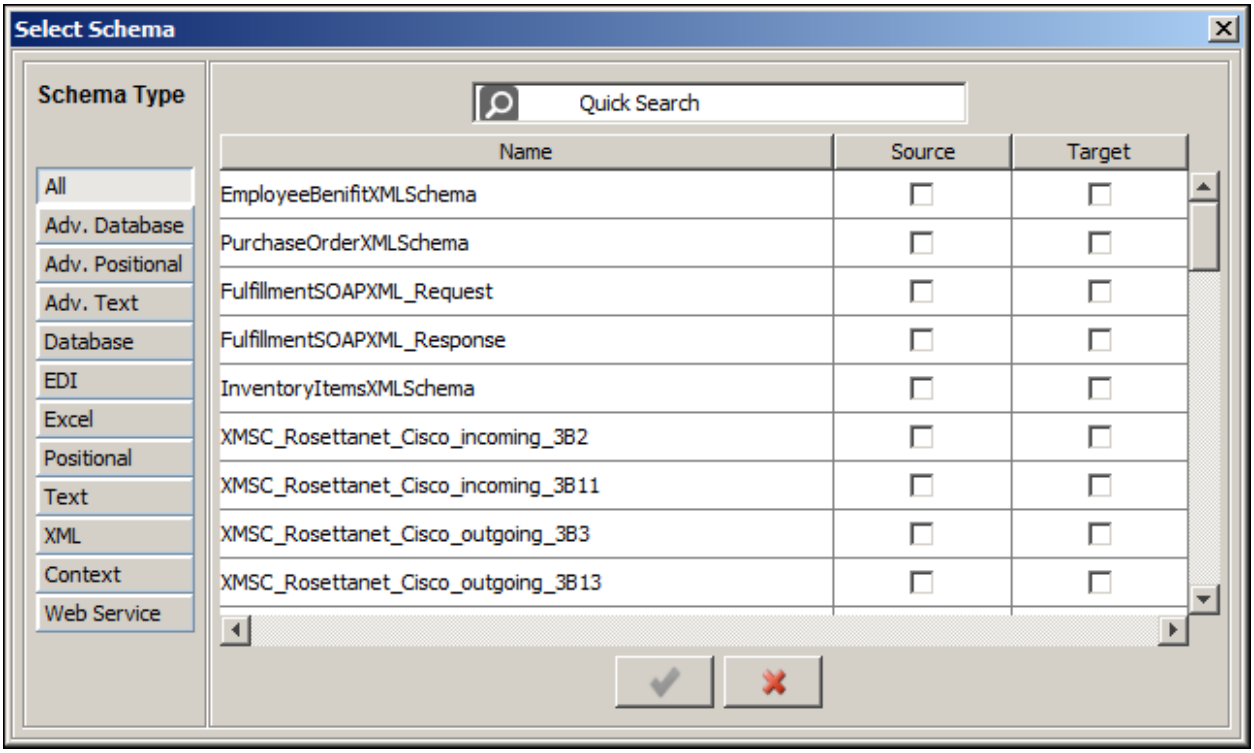

Figure 29: Select Schema

7. Select the type of schema that you want to load as the source schema. For example, select **Database** from the **Schema Type** column. All the Database schema types will be displayed.

8. Select the *EvalXform_DBSchema* schema and in this row select the checkbox under the Source column if you want to select this schema as the Source schema (see Figure 30).

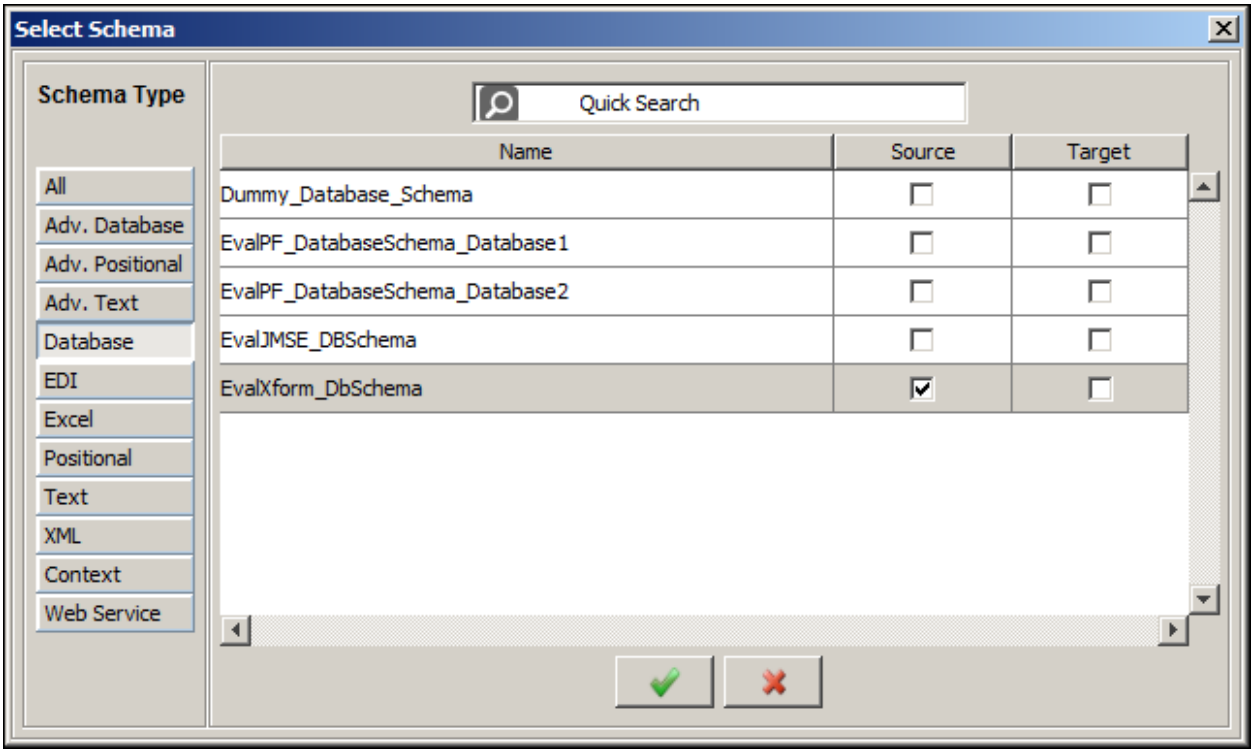

Figure 30: Select Source Schema

9. Select the type of schema that you want to load as the load schema. For example, select **Excel** from the **Schema Type** column. All the Excel schema types will be displayed.

10. Select the *EvalXform_ExcelSchema* schema and in this row select the checkbox under the Target column if you want to select this schema as the Target schema (see Figure 31).

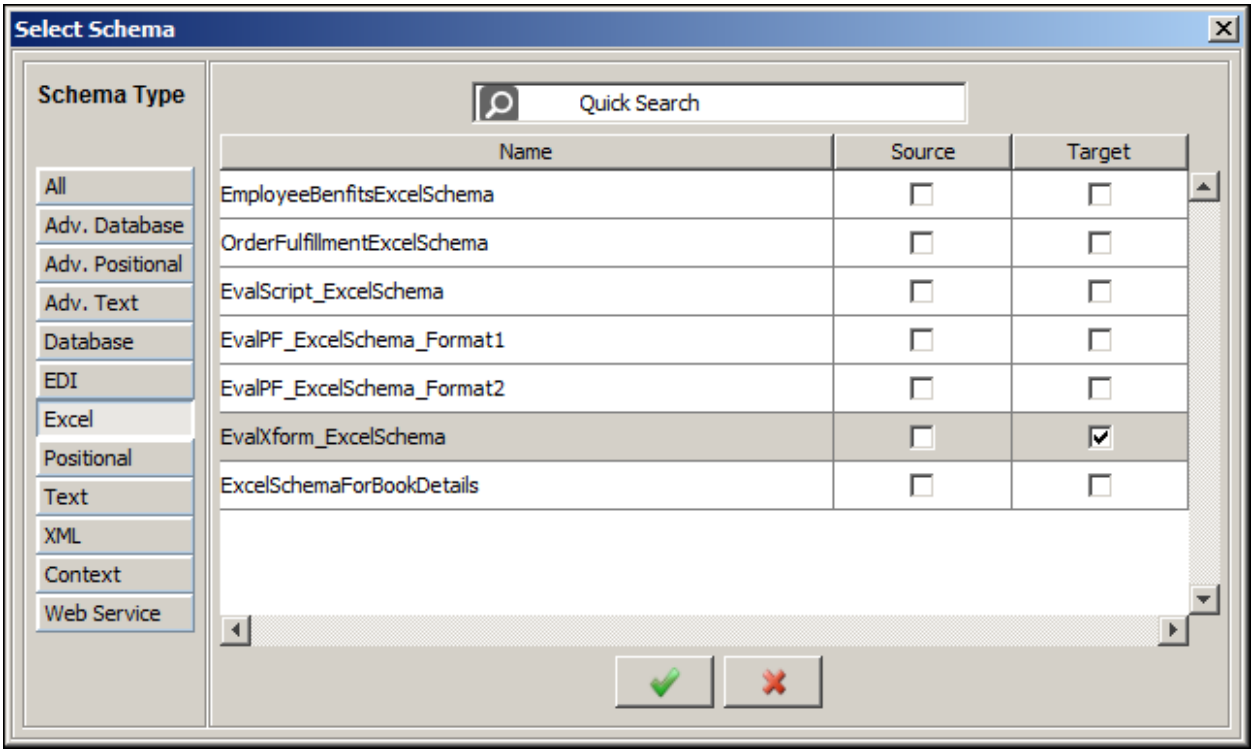

Figure 31: Select Load Schema

11. Click  button to load the schemas. This loads the selected schemas in the Source and Target Panel of the Data Mapper applet.

12. Expand the tree structure for Source Schema and the Target Schema and display their elements (see Figure 32).

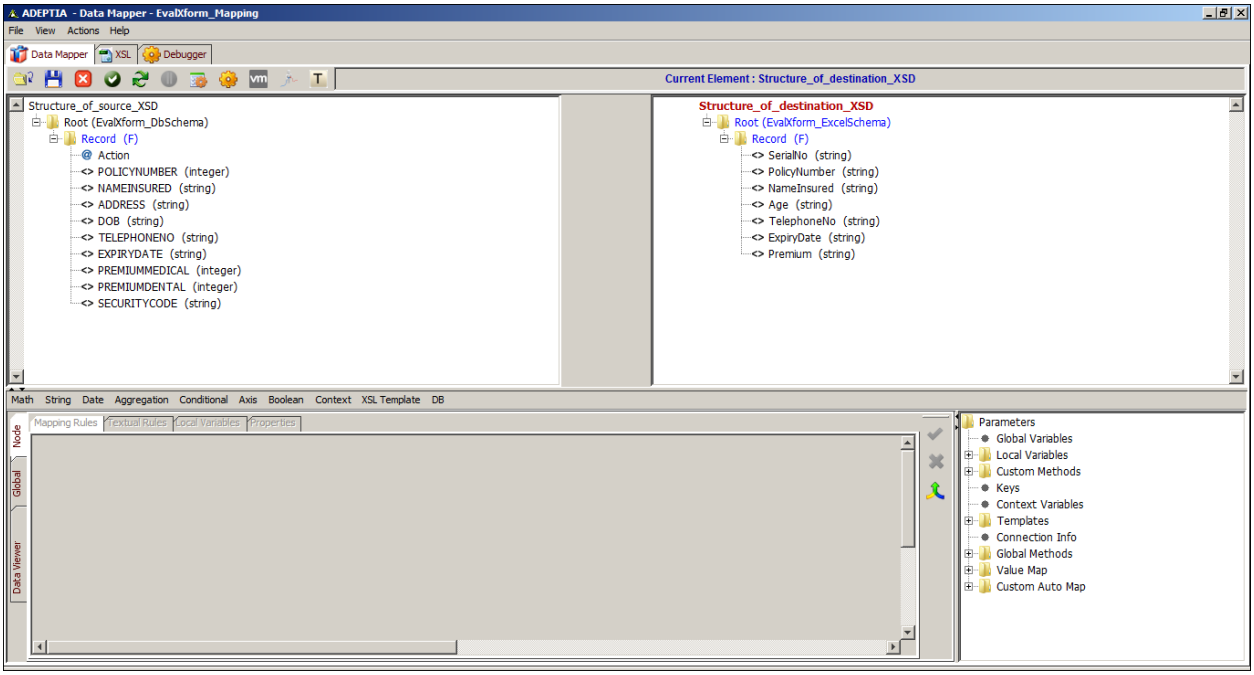

Figure 32: Source and Target Elements

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

14. 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 33).

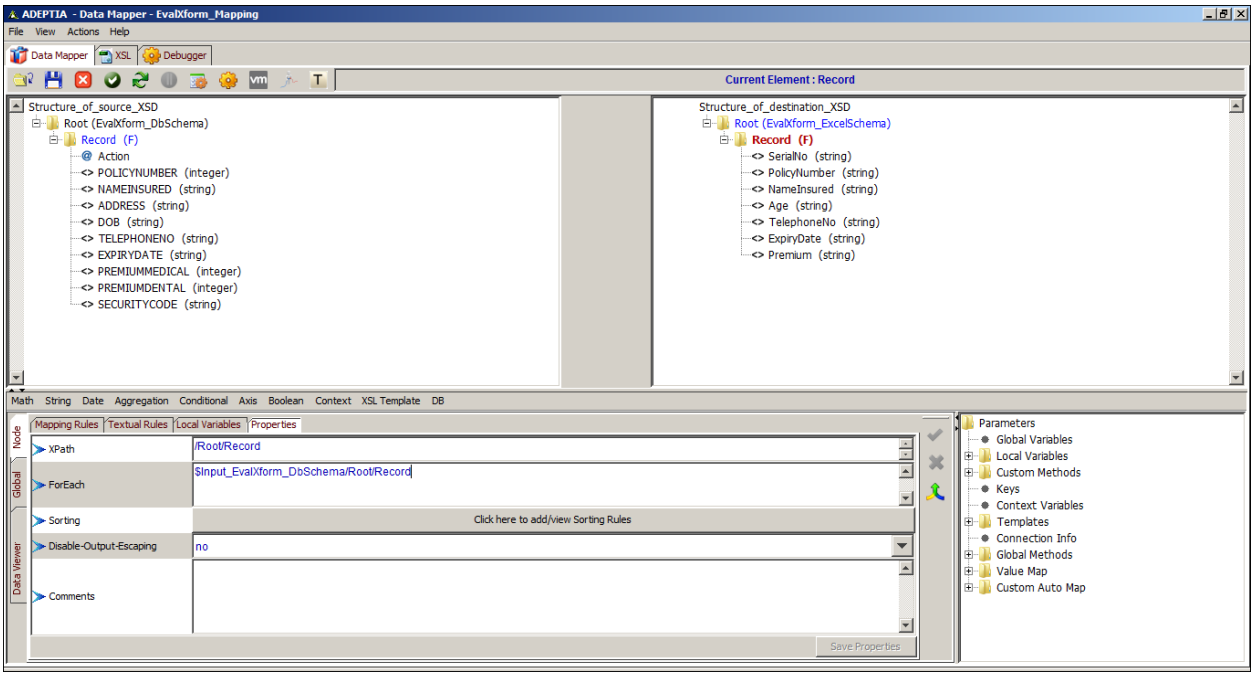

Figure 33: Apply *For Each* Property

15. Click the node **SerialNo.** on the Target Panel. The node *SerialNo.* is displayed in the Mapping Graph Area (see Figure 34).

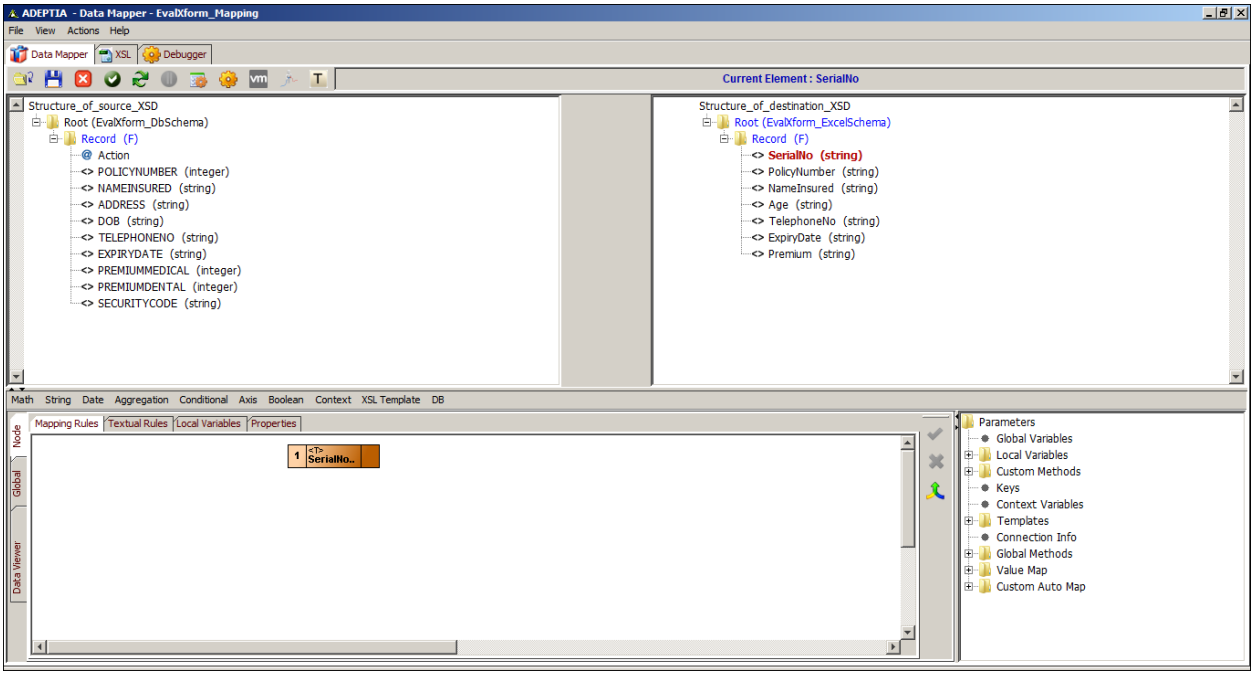

Figure 34: Select Target Node

16. 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 35).

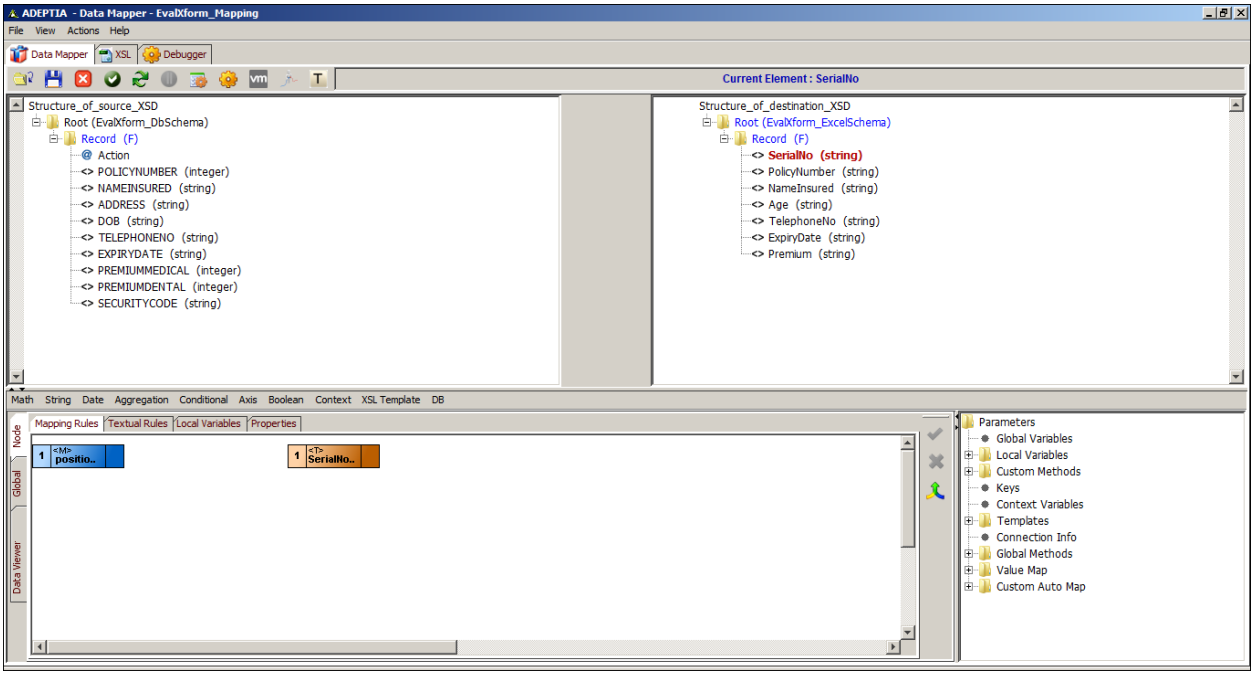

Figure 35: Select Position Function

17. 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 36).

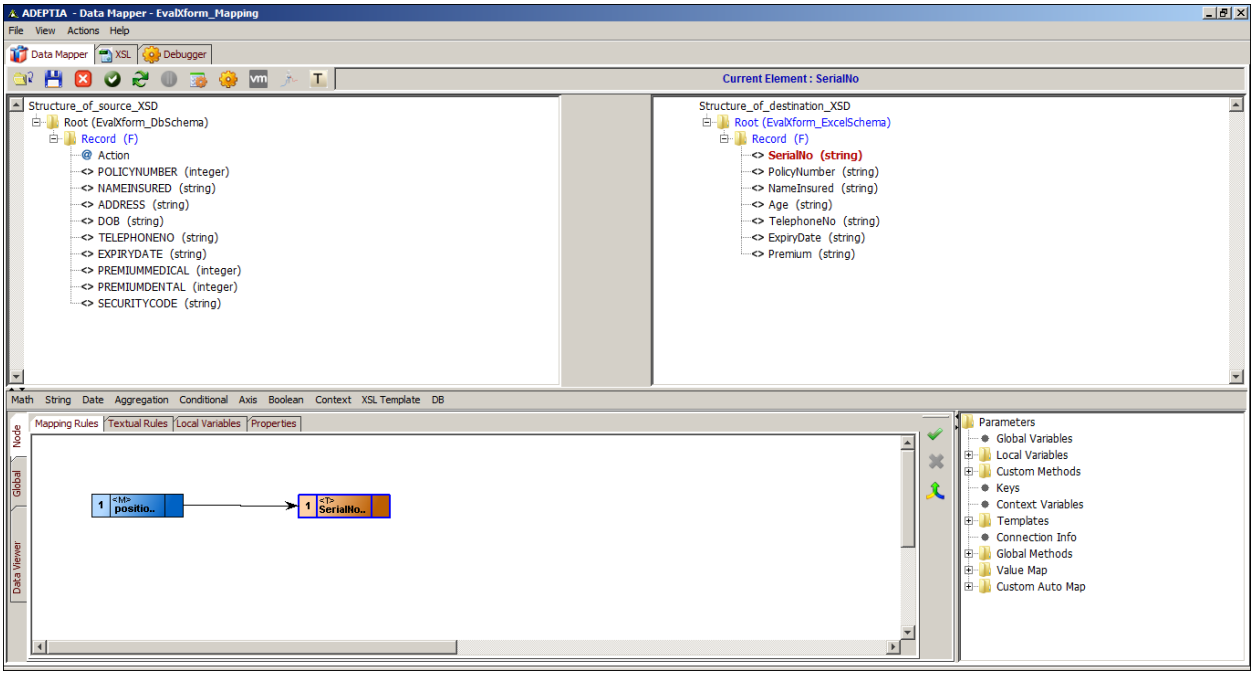

Figure 36: Map Position Function to Serial No.

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

19. 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 37).

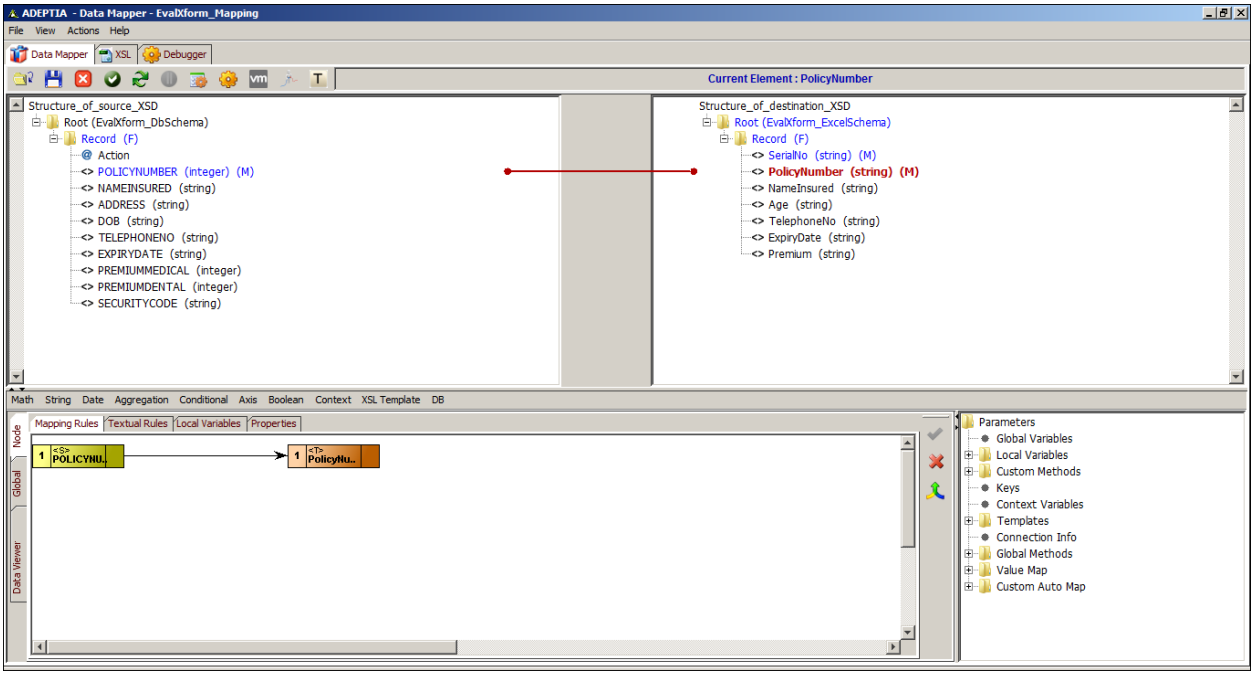

Figure 37: Map Policy Numbers

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

20. 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 38).

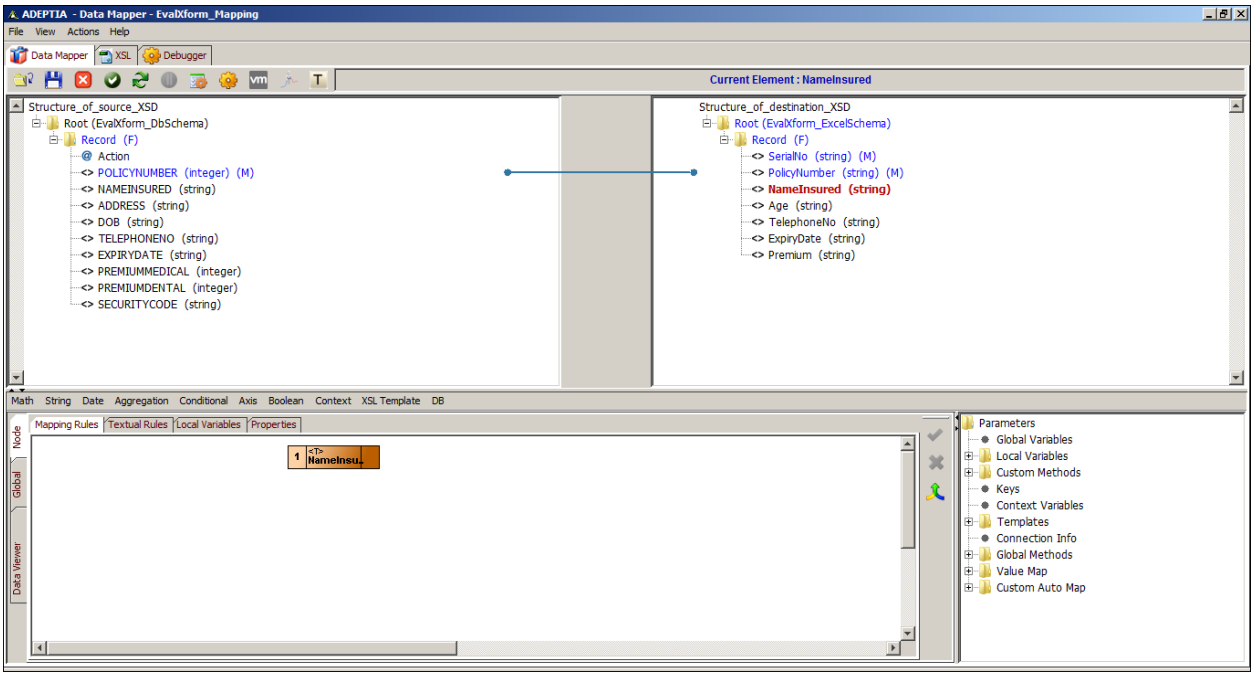

Figure 38: Select Name Insured

21. 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 39).

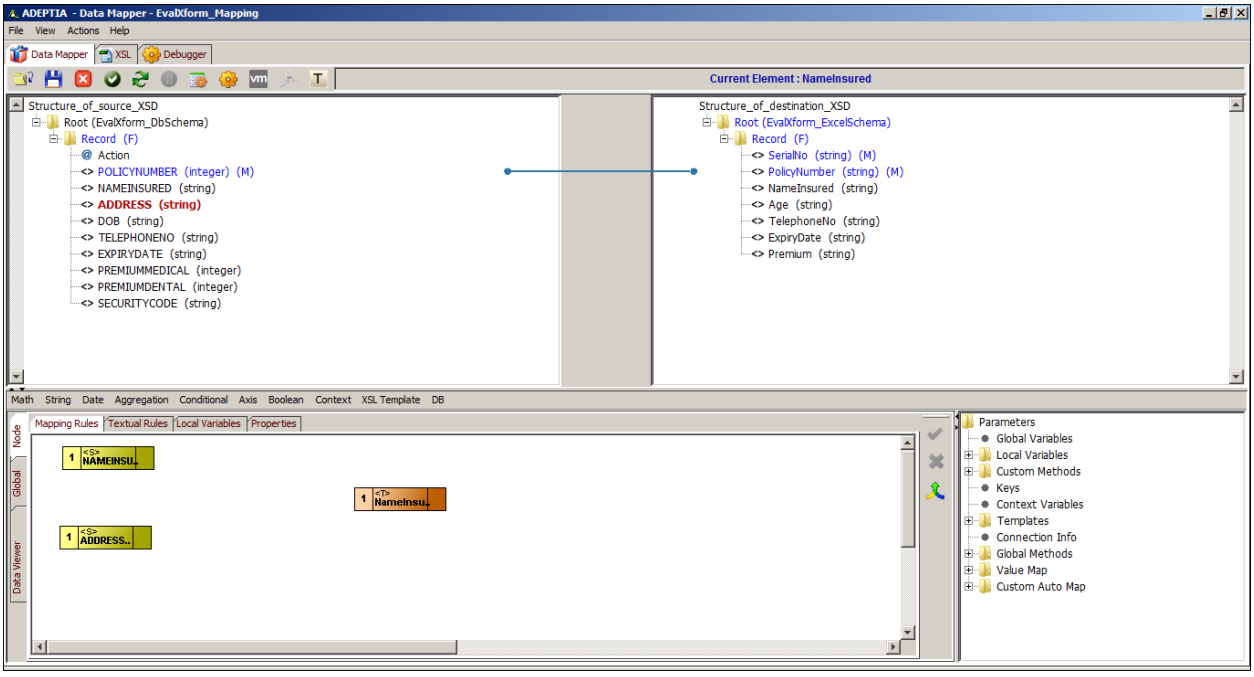

Figure 39: Select Source Nodes

22. Click the **String** mapping function and select **Concat** sub-function. The *Concat* node is shown in the Mapping Graph Area.

23. Create a link from the output of the *NameInsured* node to the first input of the *Concat* node.

24. Create a link from the output of the *Address* Node to the second input of the *Concat* node.

25. Create a link from the output of the *Concat* function node to input of the *NameInsured* node (see Figure 40).

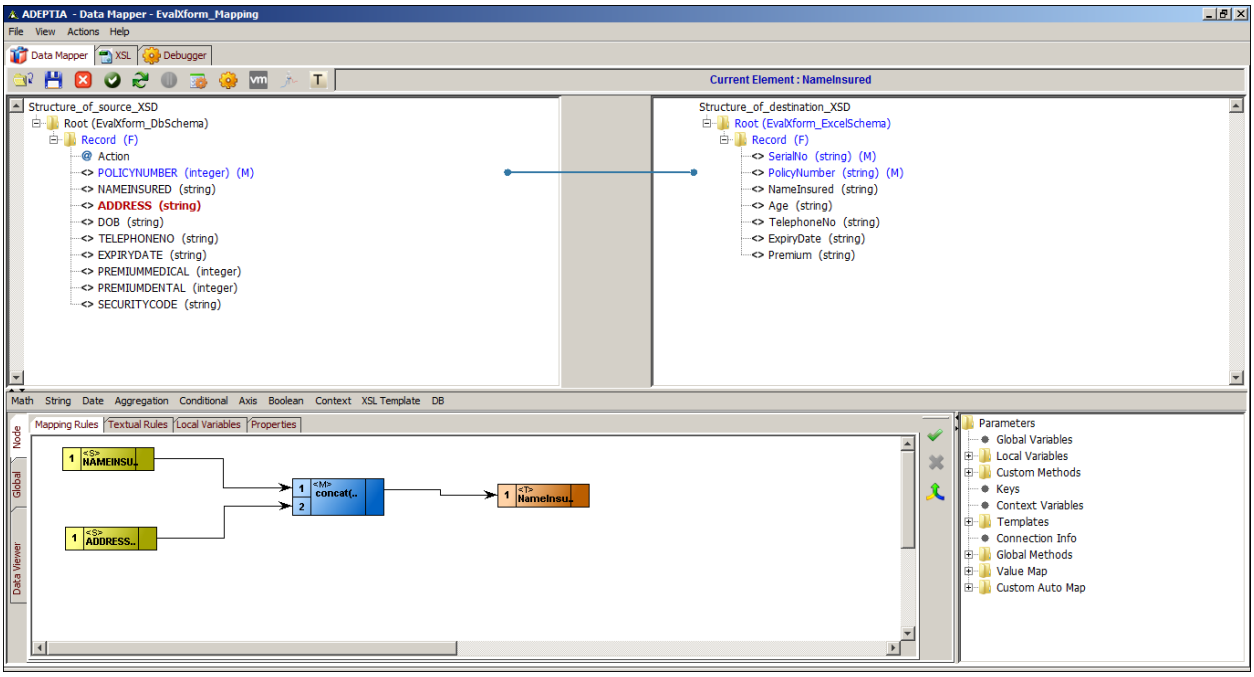

Figure 40: Create Links

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

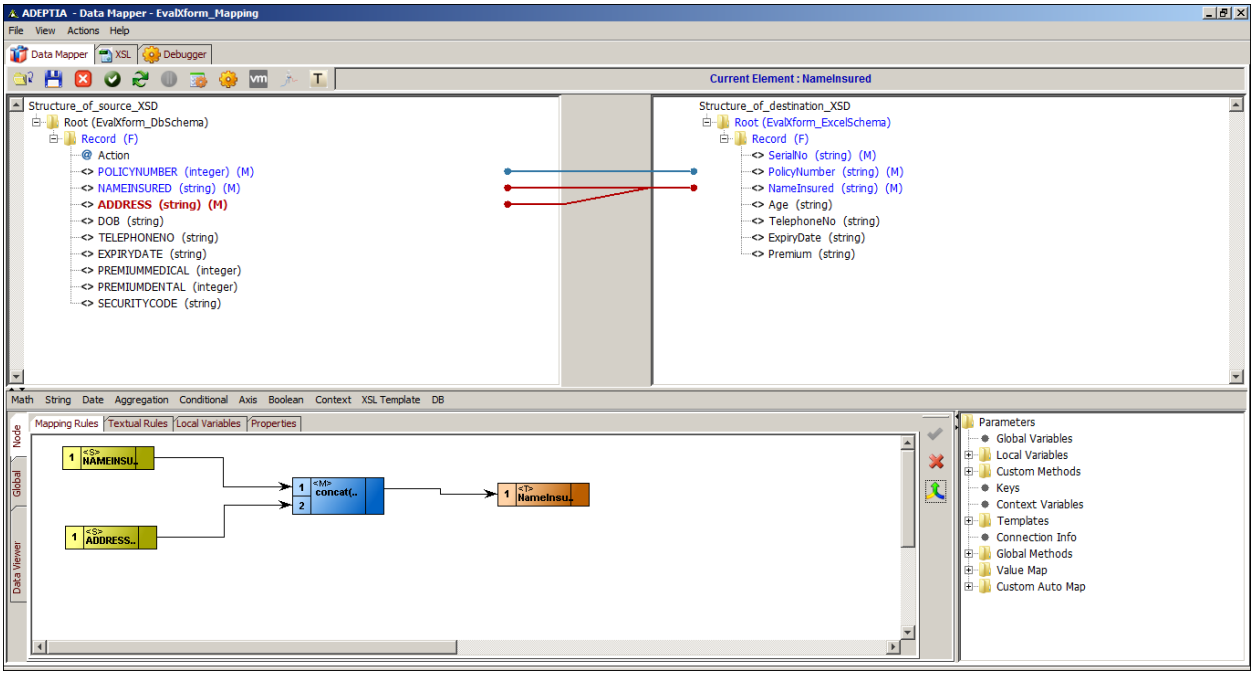

Figure 41: Apply Mapping

27. 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*.

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

29. Click the **Custom Methods** tab. The Custom Methods Panel is displayed (see Figure 42).

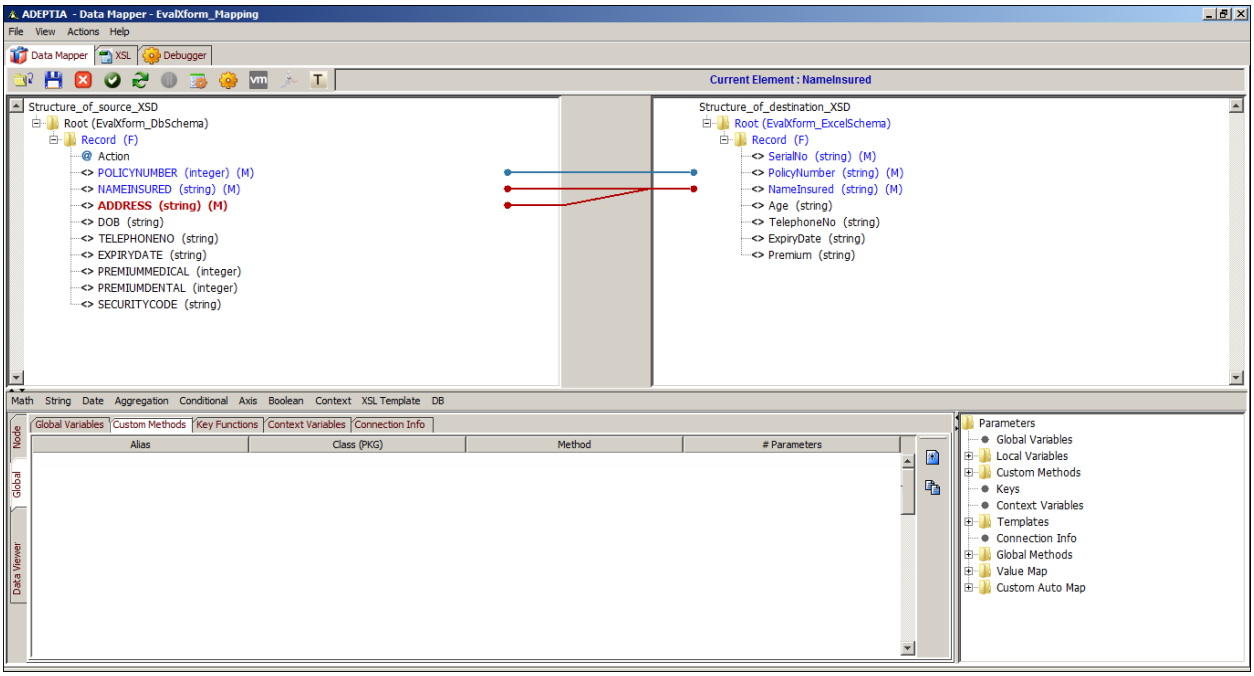

Figure 42: Custom Methods Pane

30. Click the **Add Method** ( ) button to add a Custom Method. A row is inserted with following columns (see Figure 43):

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

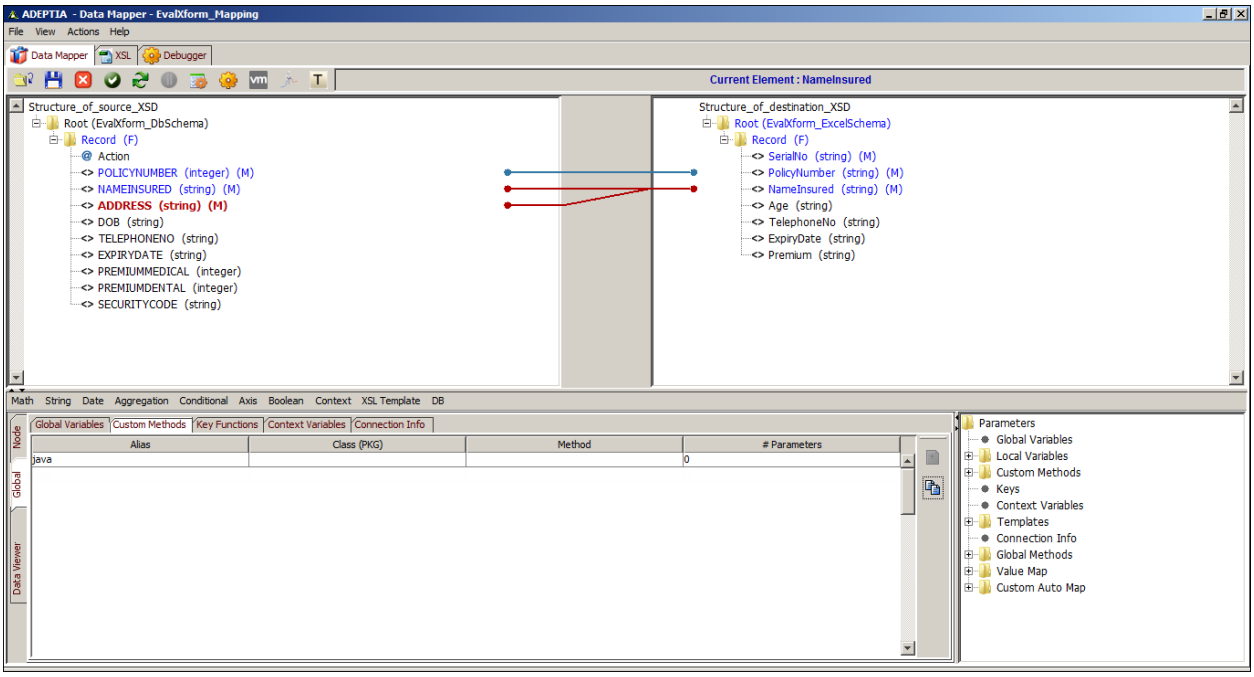

Figure 43: Add Custom Method

31. *Alias* column automatically gets populated with value 'java'.
32. Click the *Alias* field, press [**Tab**] or [**Enter**] key to go to the *Class (PKG)* field
33. Enter the fully qualified name of the Java class (*CustomMethodCall*) in the *Class (PKG)* field.
34. 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.

35. 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.

36. Press [**Tab**] or [**Enter**] key to return to the *Alias* field. This will save the added Custom Method (see Figure 44).

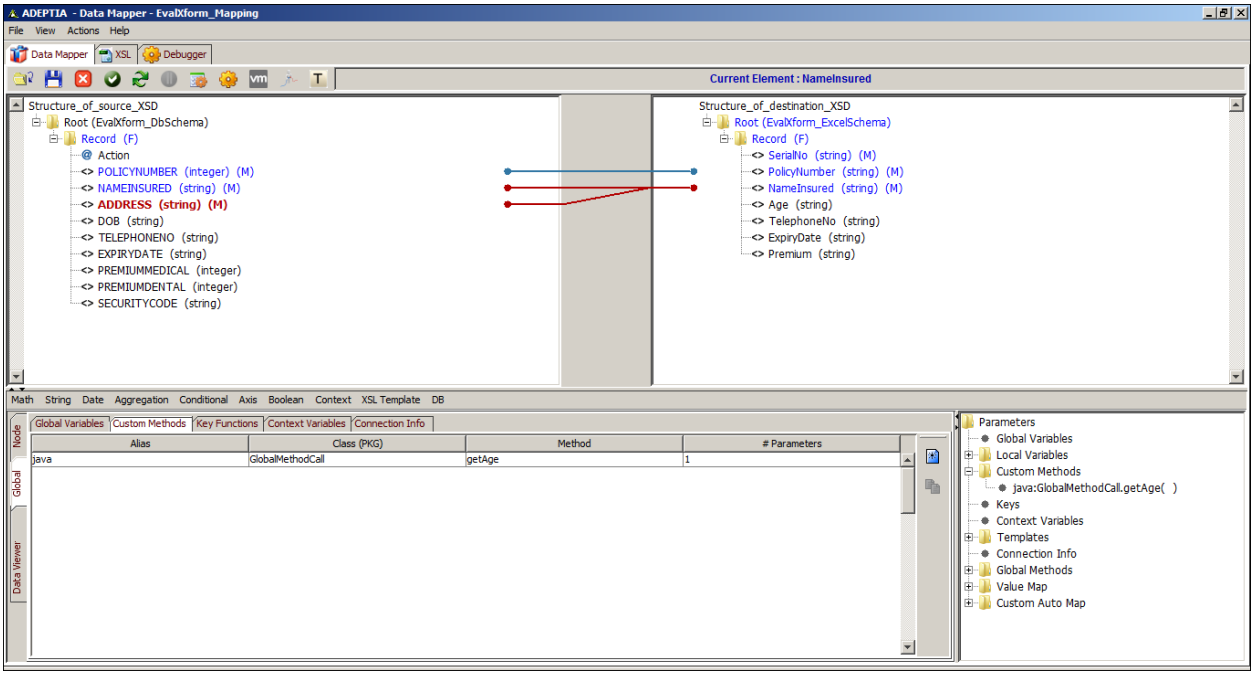

Figure 44: Save Custom Method

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

37. Click the **Node** tab to map the declared Custom Method to the *Age* node. The Mapping Graph Area is displayed.
38. Click the node **Age** in the Target Panel. The *Age* node is displayed in the Mapping Graph Area.
39. Double-click the **DOB** node in the Source Panel. The *DOB* node is displayed in the Mapping Graph Area.

40. 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 45).

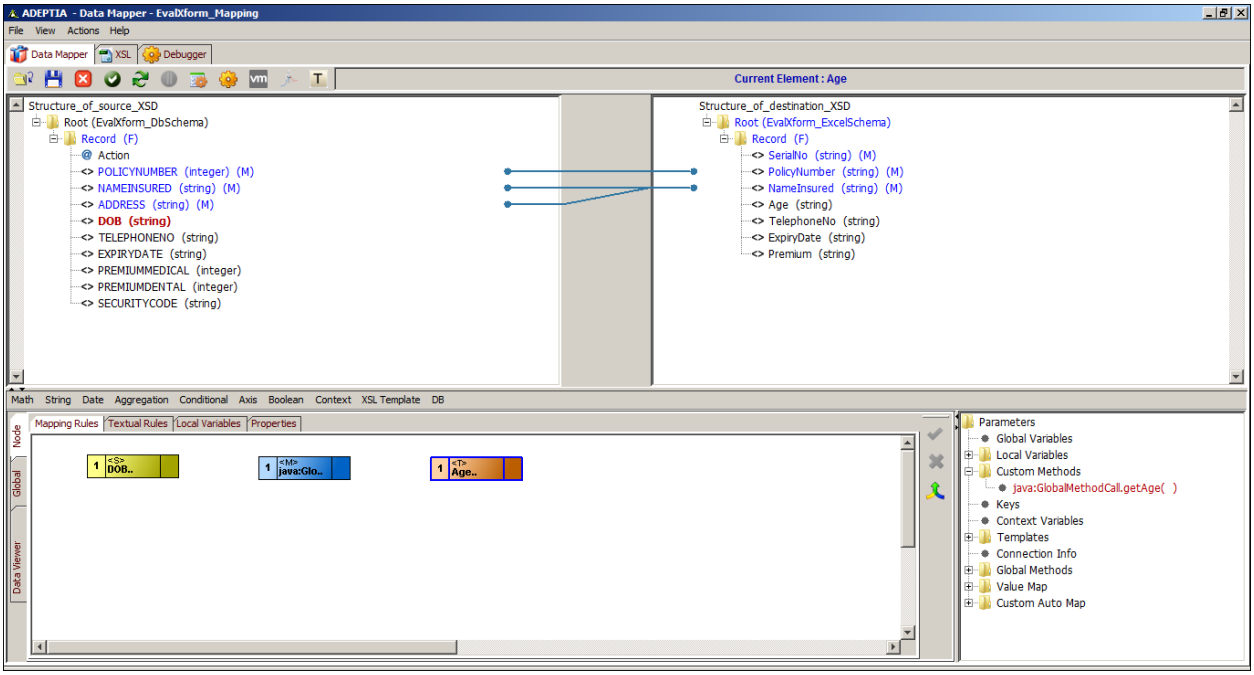

Figure 45: Select Custom Method

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

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

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

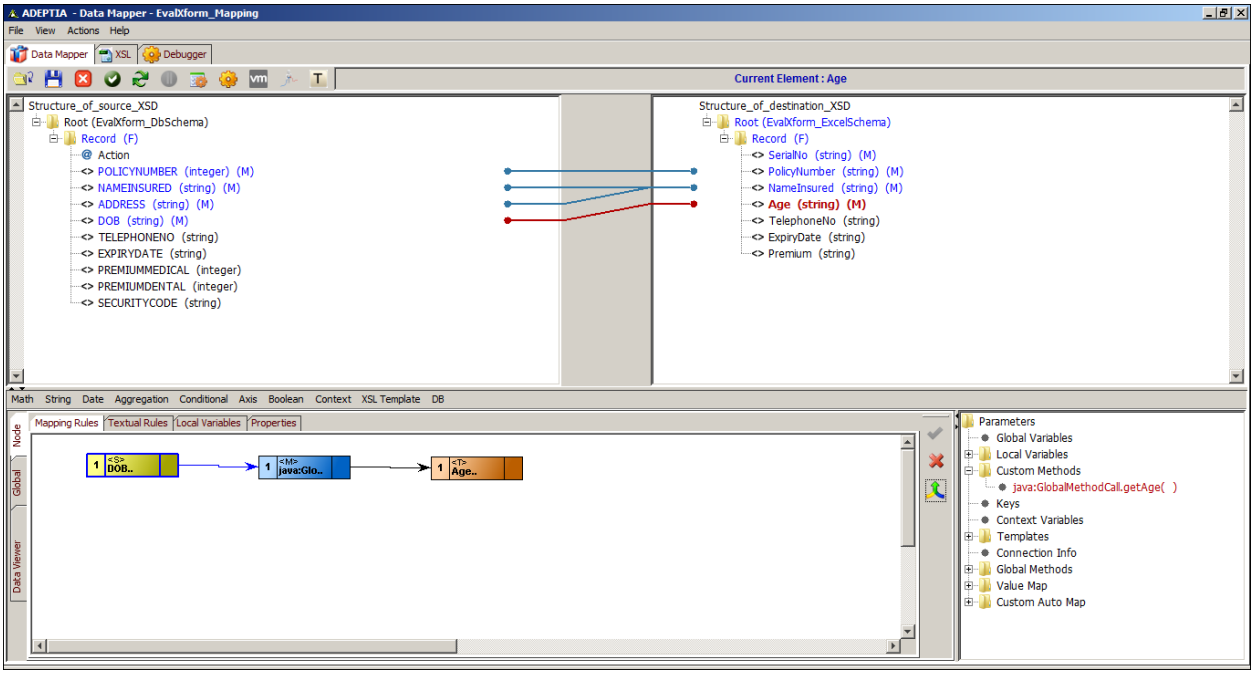

Figure 46: Map Custom Method

44. 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.

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

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

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

48. 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 47).

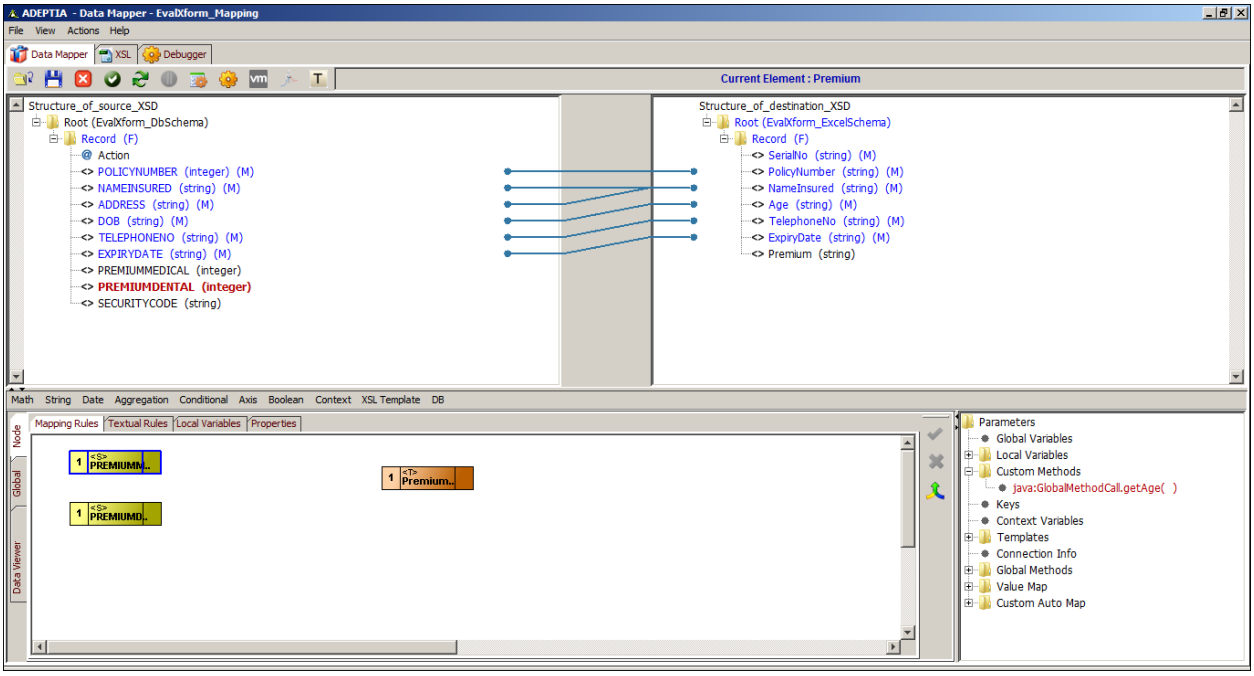

Figure 47: Select Source Nodes

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

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

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

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

53. Click the **Apply Mapping** button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 48).

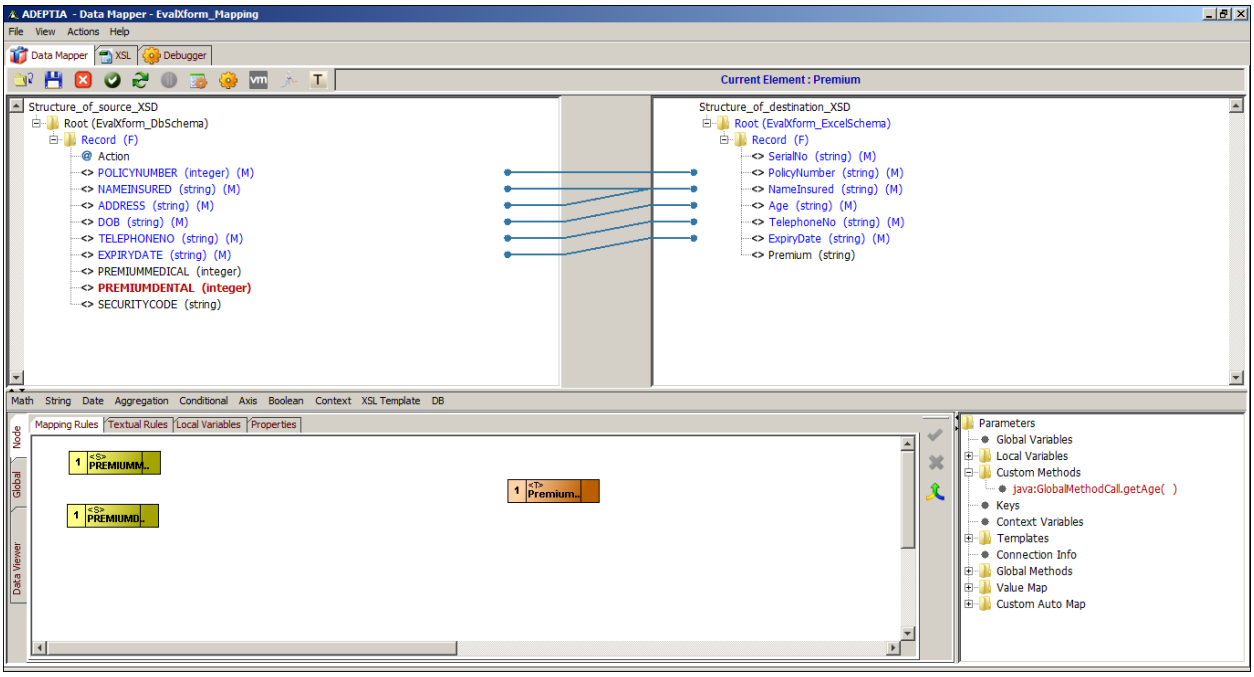

Figure 48: Map Premium Nodes

54. 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.

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

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

57. 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.

58. Double-click the *Constant* node. The Input dialog box is displayed (see Figure 49).

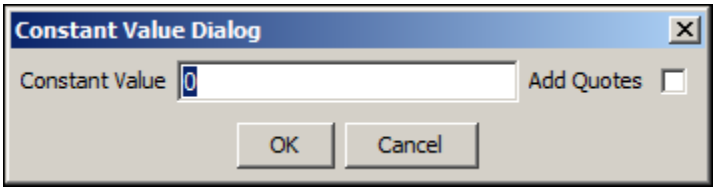

Figure 49: Input Dialog Box

59. 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 50).

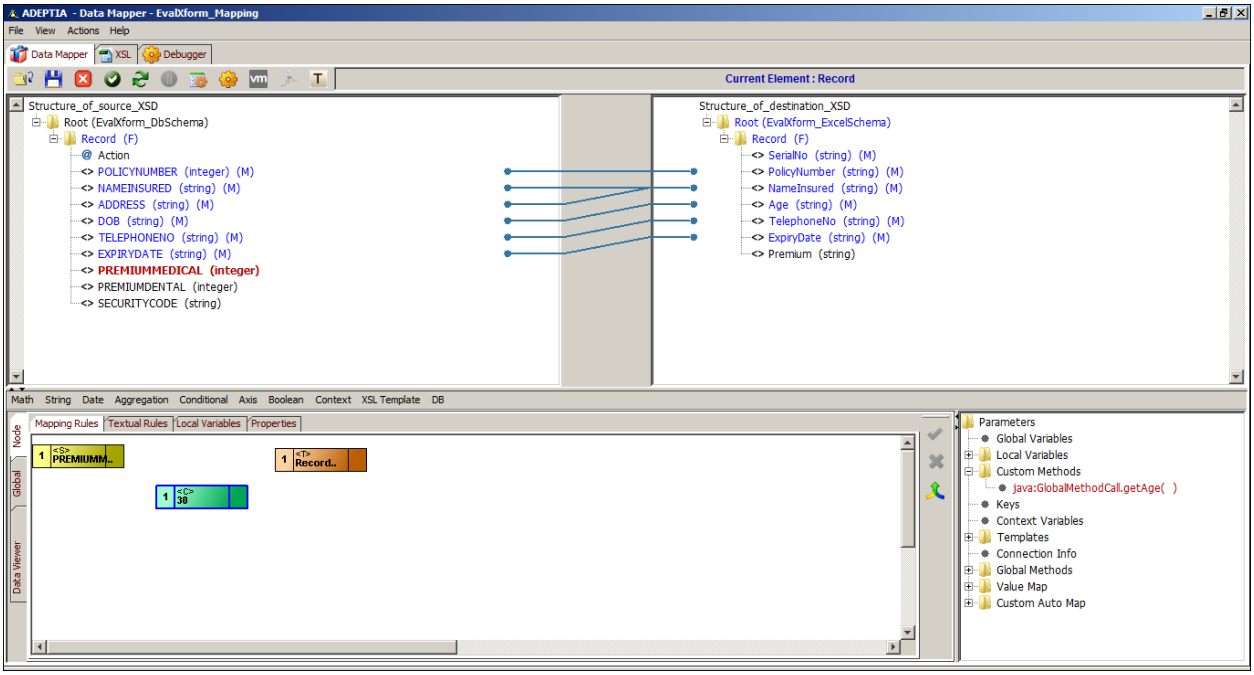

Figure 50: Constant Value Entered

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

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

62. Create another link from the *Constant* node to the second input of the *Greater Than* node (see Figure 51).

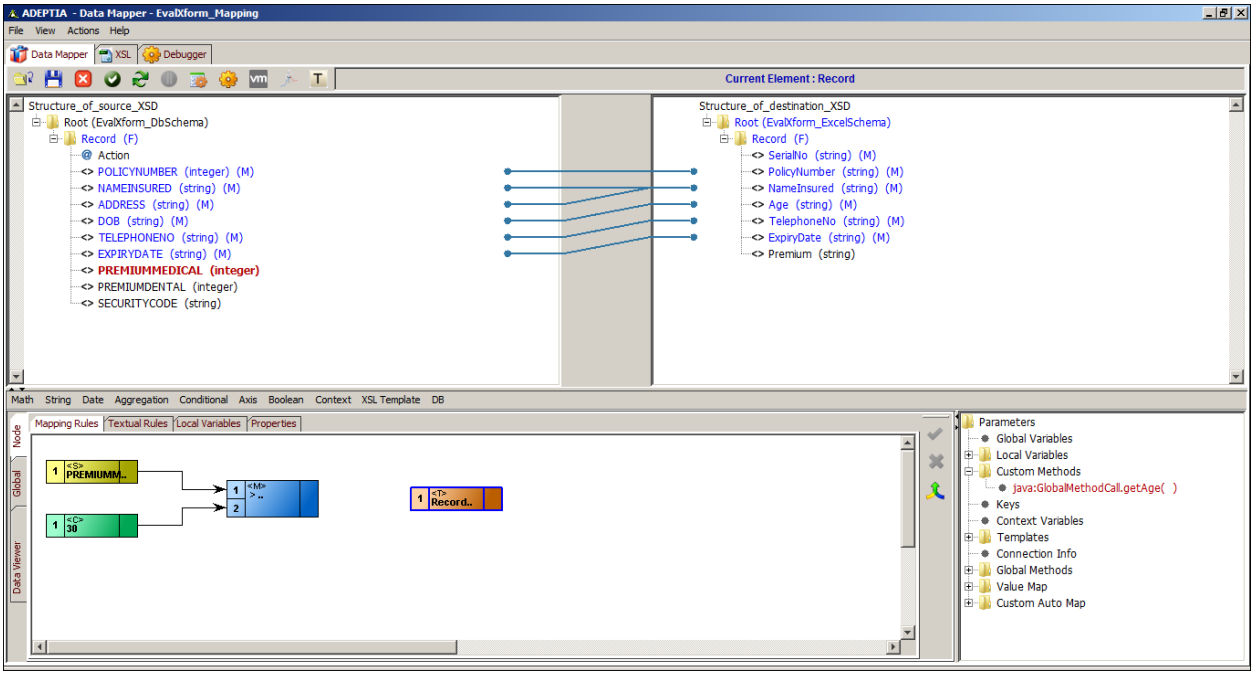

Figure 51: Create Links

63. Create another *Constant* node with the value (‘*OK*’).

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

64. 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 52).

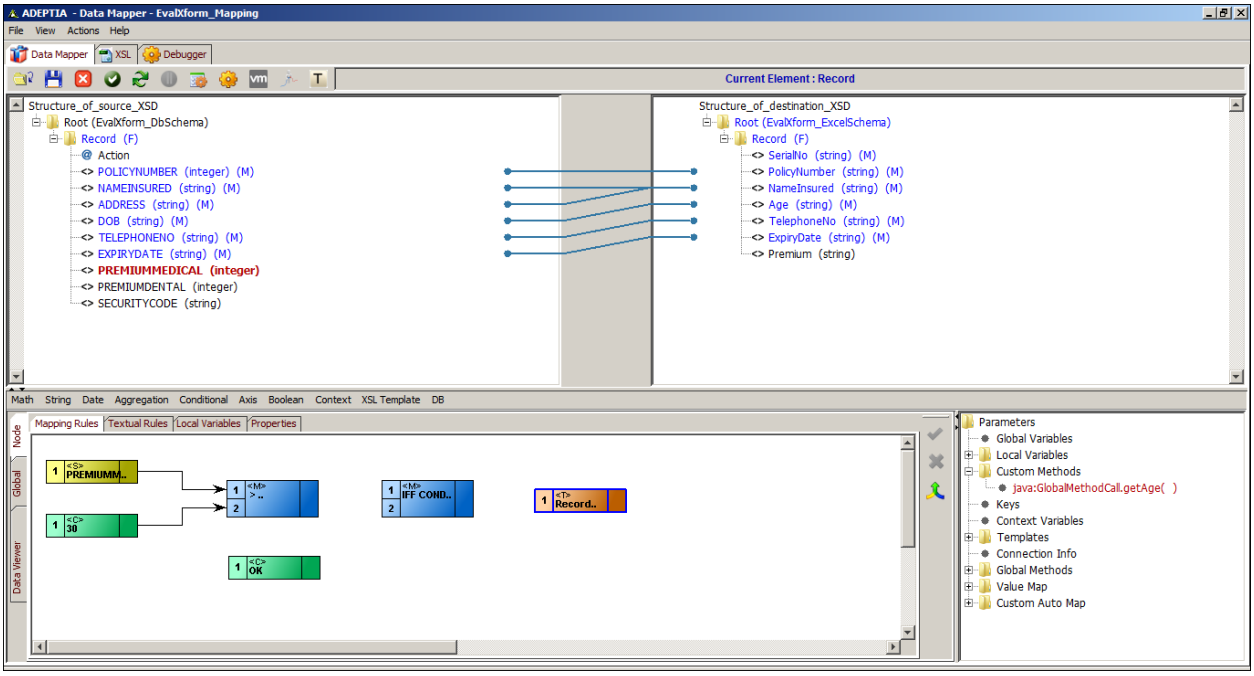

Figure 52: Select IFF Condition

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

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

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

68. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 53).

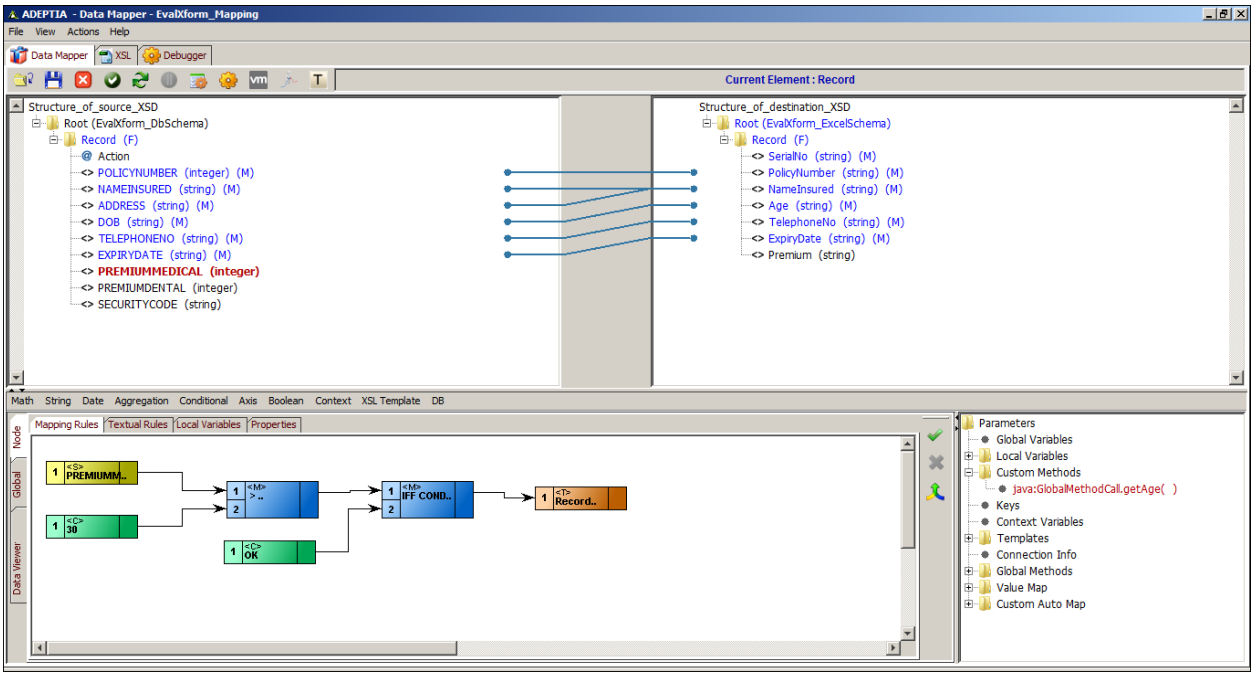

Figure 53: Map IFF Condition with Target Element

69. Lines indicating mapping between the source and target nodes are displayed (see Figure 54).

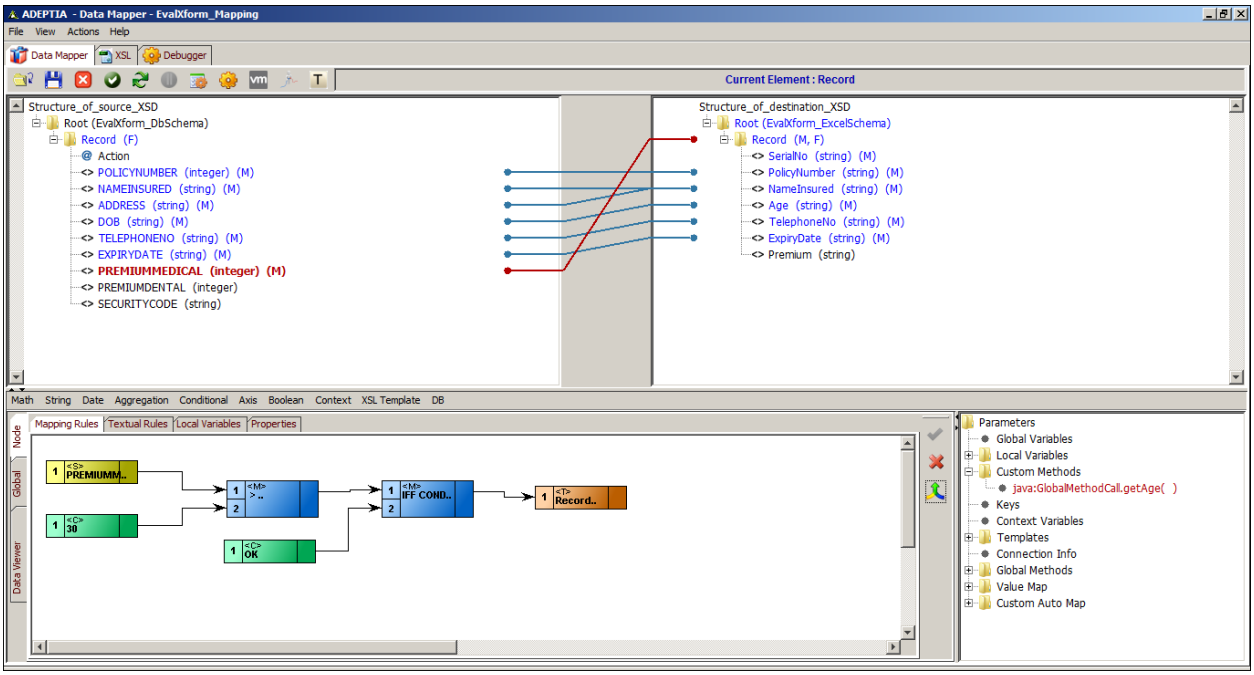

Figure 54: Map Custom Method

70. 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.

71. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping.
72. Enter the comments in the Specify comments for mapping object (object name) field.
73. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.
74. 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (Figure 55).

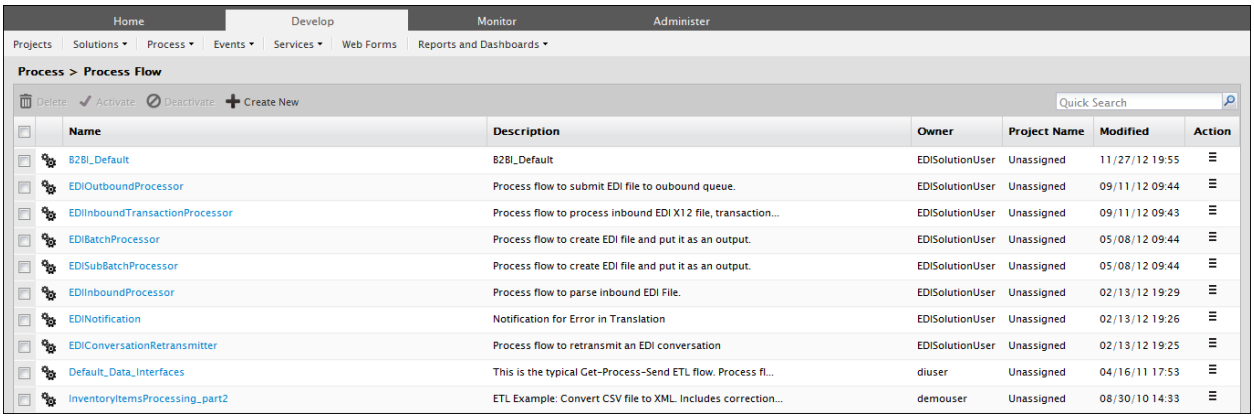

Figure 55: Manage Process Flow

3. Click the **Create New** link. The *Create Process Flow* screen is displayed (see Figure 56).

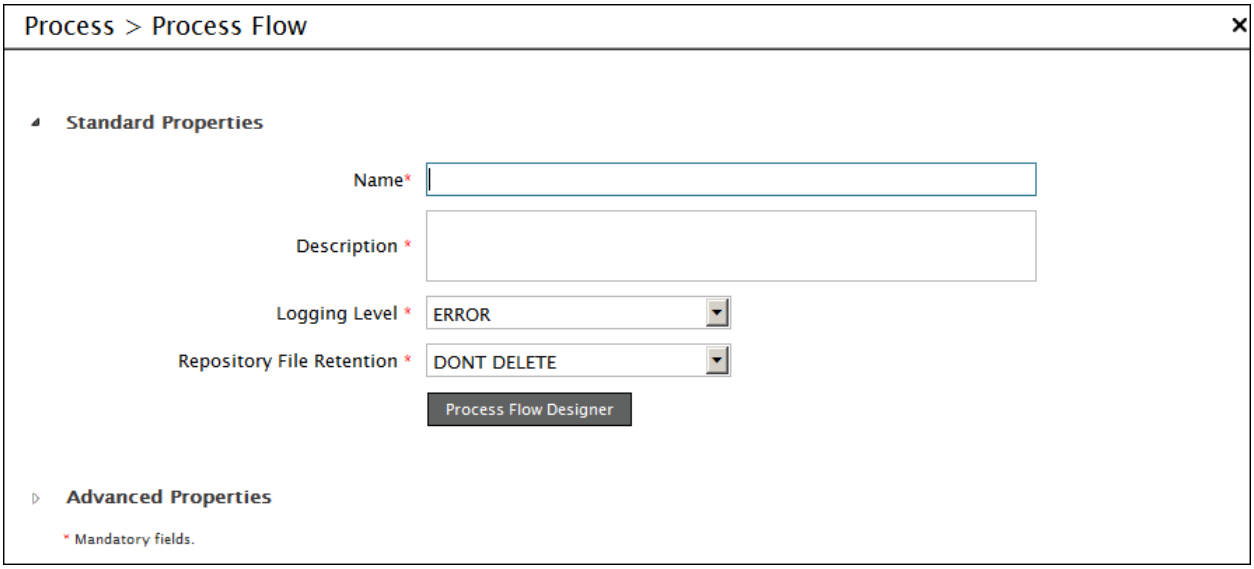

Figure 56: Create Process Flow

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* dropdown list. There are four levels of logging. The following table describes these logging levels Table 10.

Table 10: Logging Levels

Logging Levels	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.

The following Table 11 describes these options.

Table 11: 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 (refer to Figure 57).

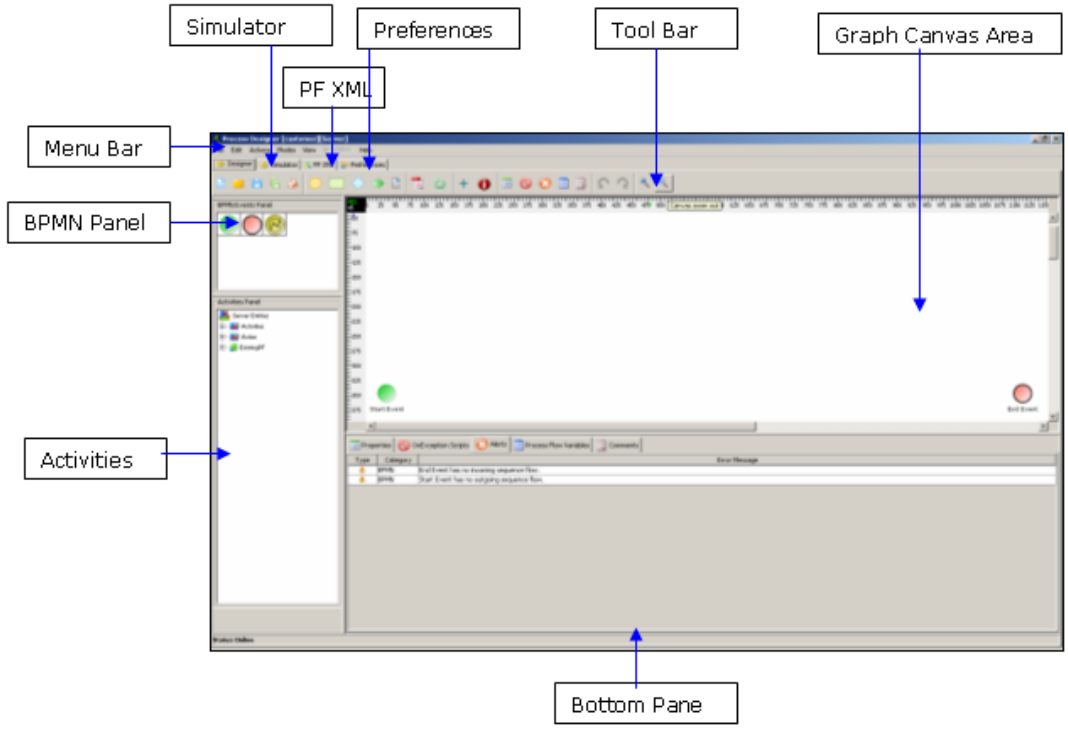

Figure 57: Process Designer Applet

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. In the Process Designer window, click **[+] Activities** in Activities Panel to expand the list of services and then click **[+] Source**. All 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 58).

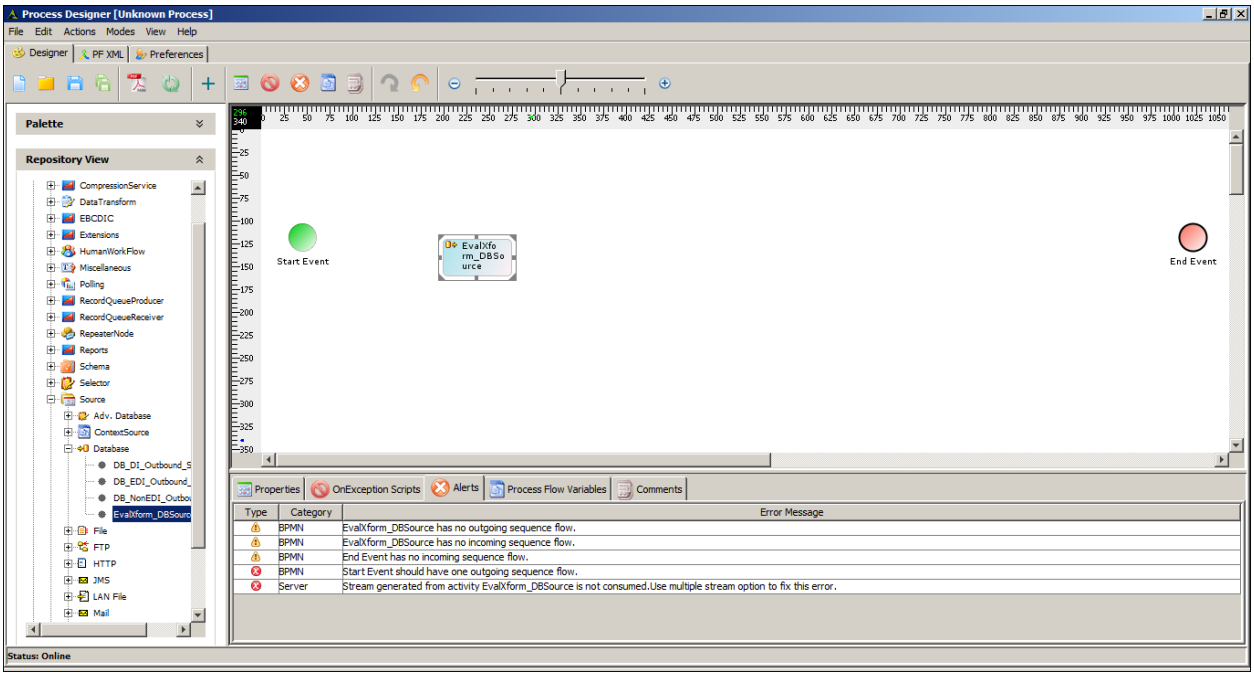

Figure 58: Drag Database Source to Graph Canvas Area

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 59).

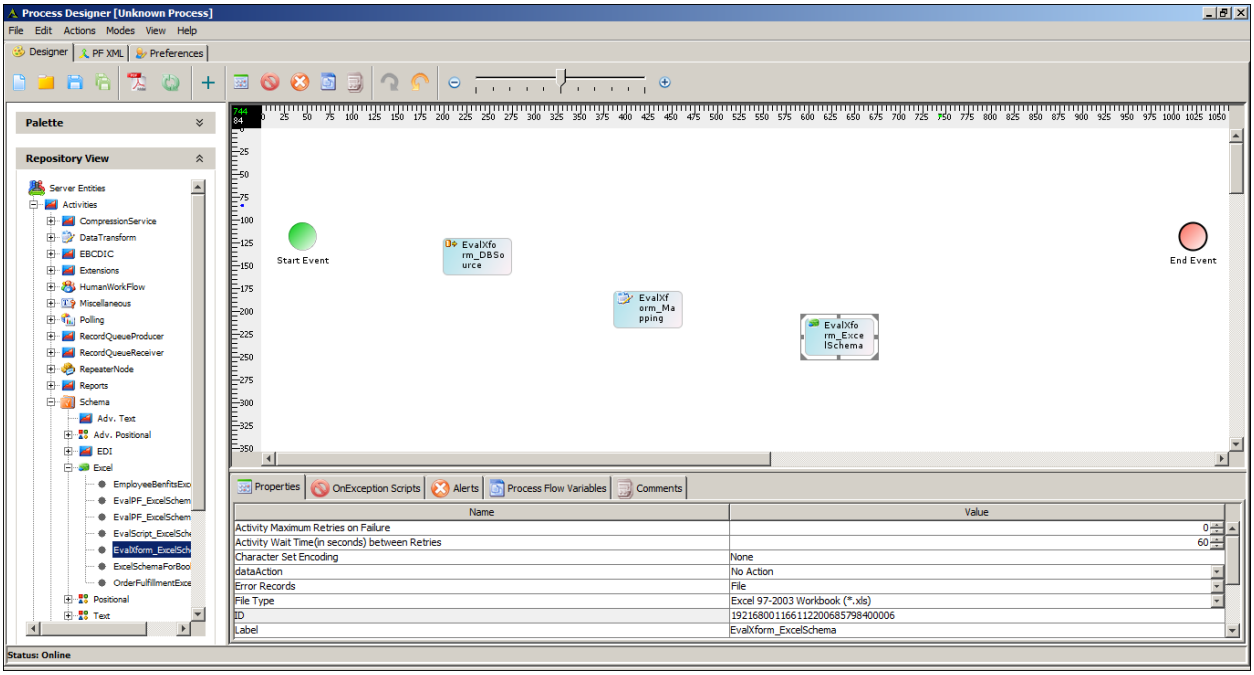

Figure 59: View Properties of *EvalXFormExcelSchema* Activity

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 60).

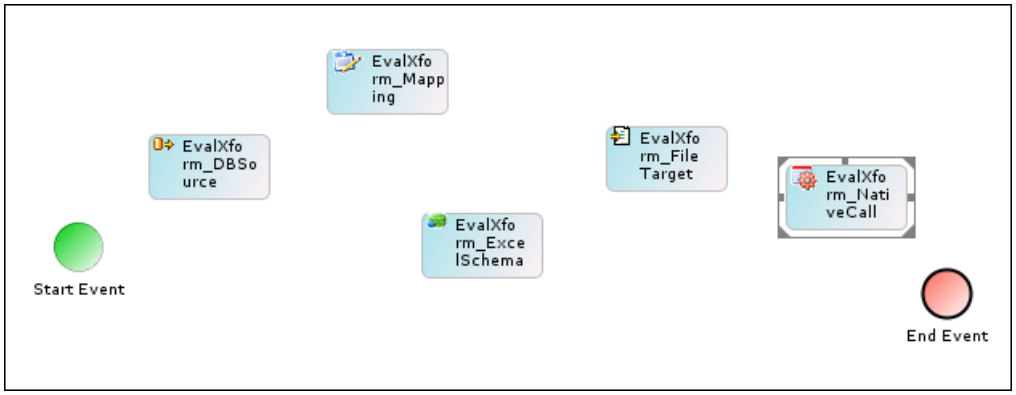

Figure 60: Connect Start Event to Database Source

21. Similarly, connect all other activities as shown in Figure 61

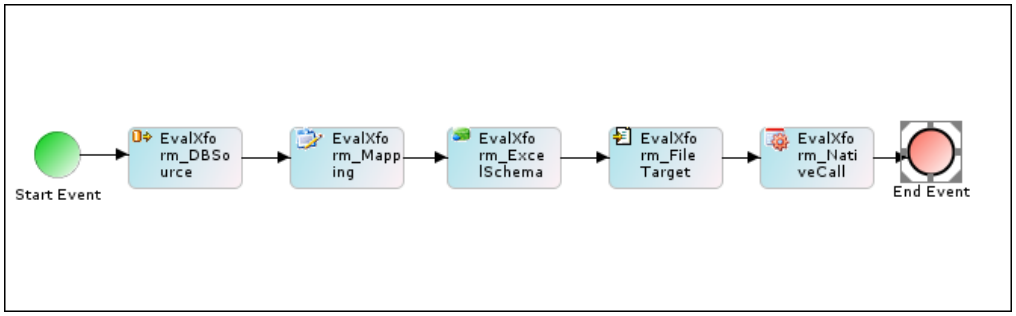

Figure 61: Connect all Activities

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 193).

23. Enter 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**.

3

SCRIPTED SERVICE PROCESS FLOW

This section describes the Scripted Service Process Flow.

INTRODUCTION

In the Adeptia Suite this process flow is available in:

EBIM Suite	BPM Suite	ESB Suite	ETL Suite
	√	√	√

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:

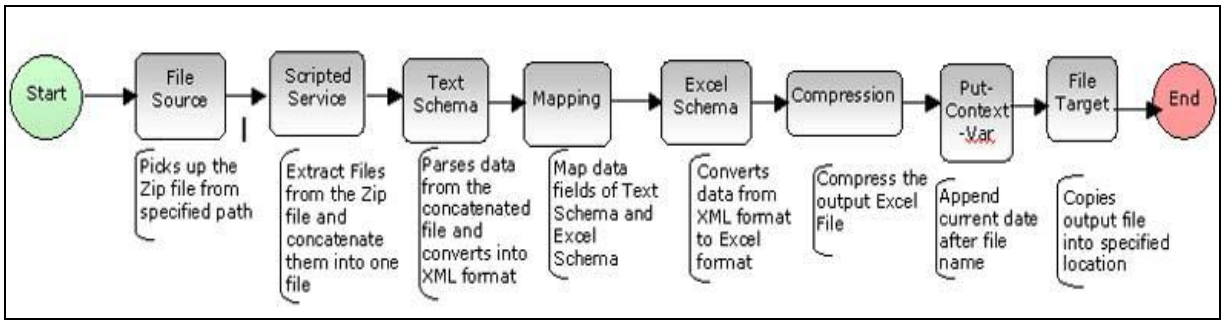

Figure 62: Flow Chart to Show Scripted Service Process Flow

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 12: 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed (refer Figure 2).
3. Click the *Execute* button, which is adjacent to *EvalScript_ProcessFlow.* A confirmation dialog box is displayed (refer to Figure 3).
4. Click the button **Go to Logs** to view the status of the Process Flow execution. The Process Flow Logs are displayed (see Figure 63).

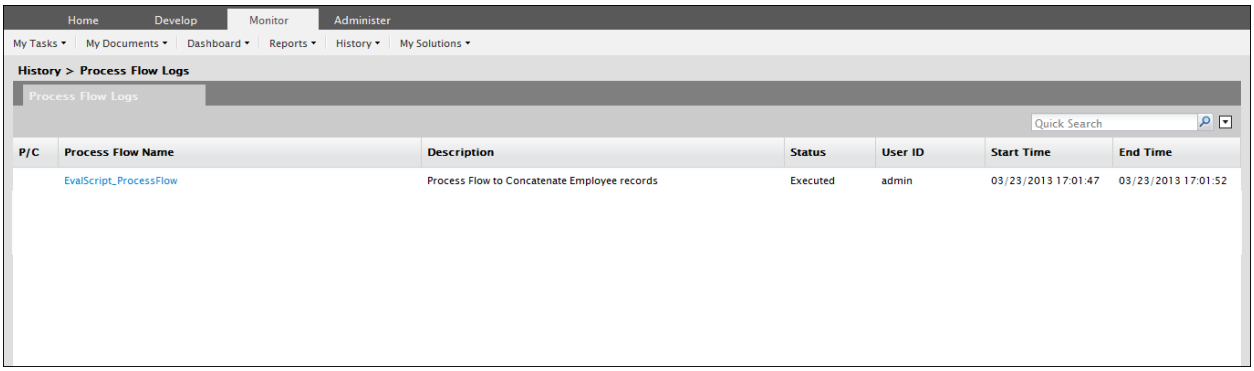

Figure 63: Searched Process Flows

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

5. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 64).

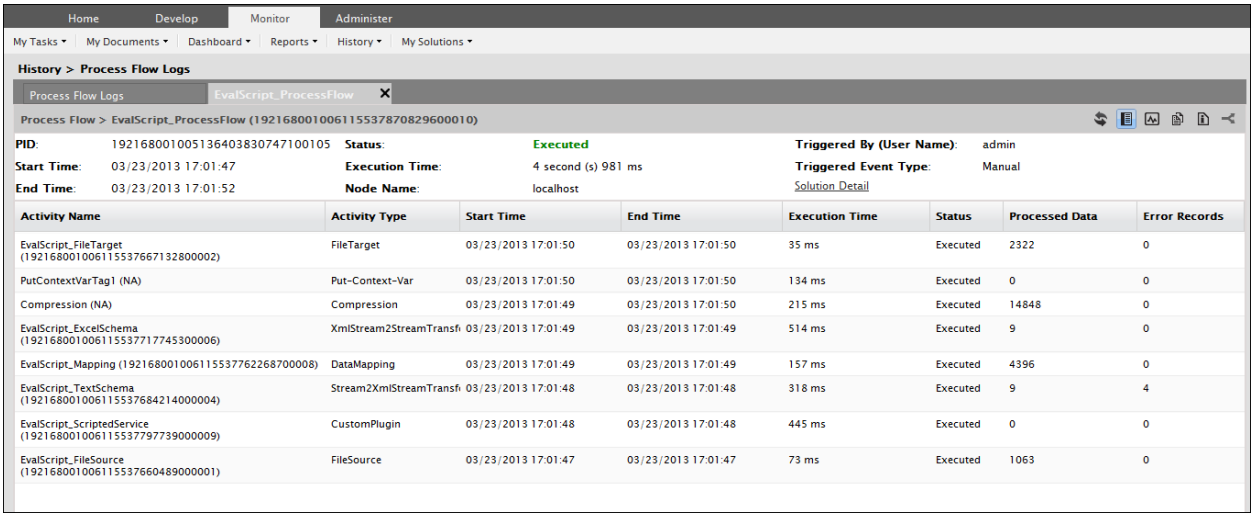

Figure 64: View Process Flow Log Details

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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Source** and then click **File**.

The *Manage File Source* screen is displayed with the list of existing File source activities (see Figure 65).

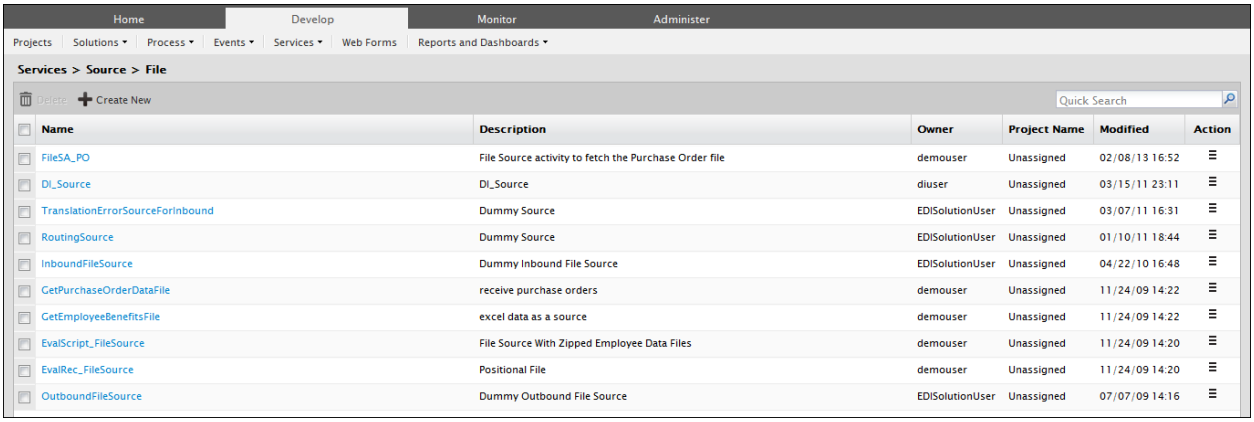

Figure 65: Manage File Source

3. Select the required File source activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective File source activity in edit mode (see Figure 66).

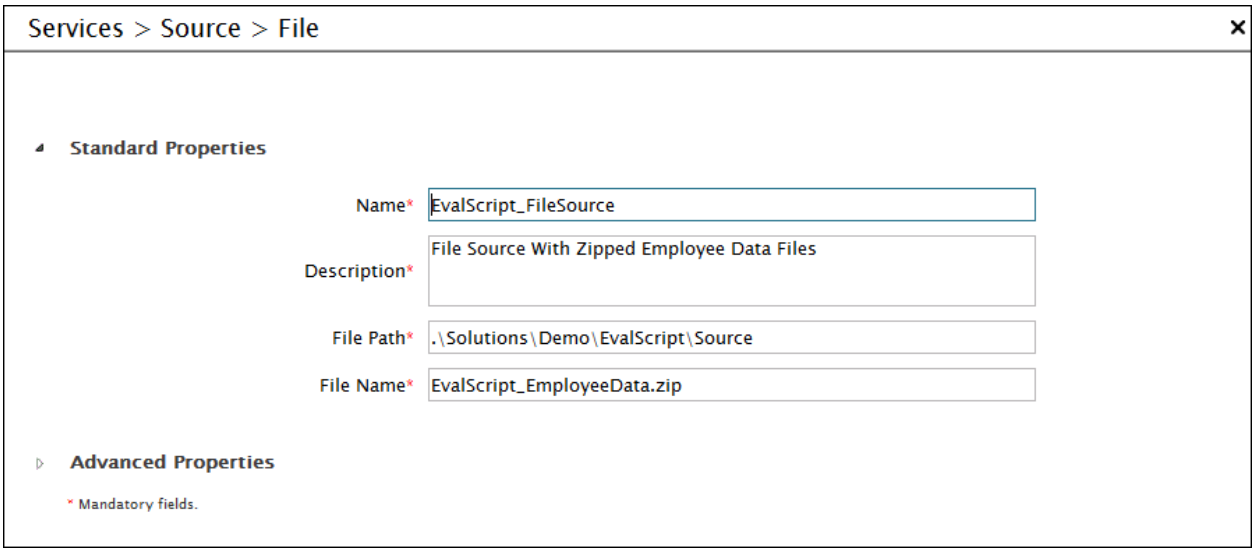

Figure 66: Edit *EvalScript_FileSource* Activity

The following table defines the fields on this screen.

Table 13: Details of Fields on Edit File Source Screen

Field Name	Description
Name	Name of the File Source
Description	Description of the File Source
File Path	Source Path of the source file. For example: ../../Solutions/Demo/EvalScript/

Field Name	Description
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 175).
7. Enter comments in the **Add Comments** field.
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 67:

```
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 67: Sample JAVA Code

Steps to edit the Scripted Service

1. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Extensions** and then click **Custom Plugin**. The *Manage Custom Plugin* screen is displayed with the list of existing custom plugin activities (see Figure 68).

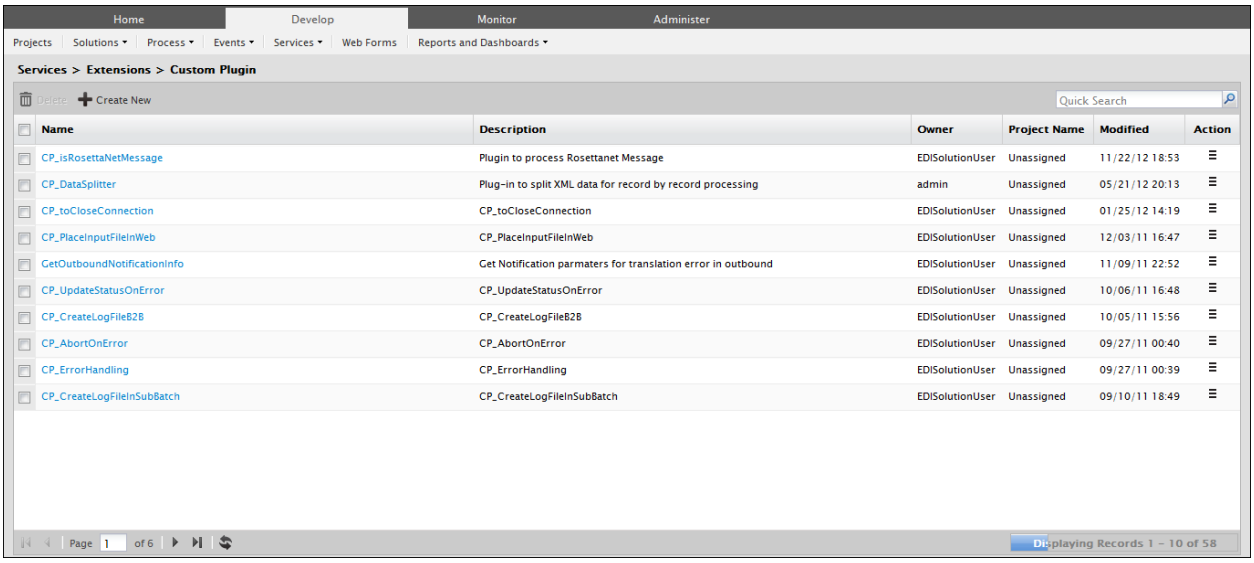

Figure 68: Manage Custom Plugins

3. Select the required Custom Plugin activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Custom Plugin activity in edit mode (see Figure 69).

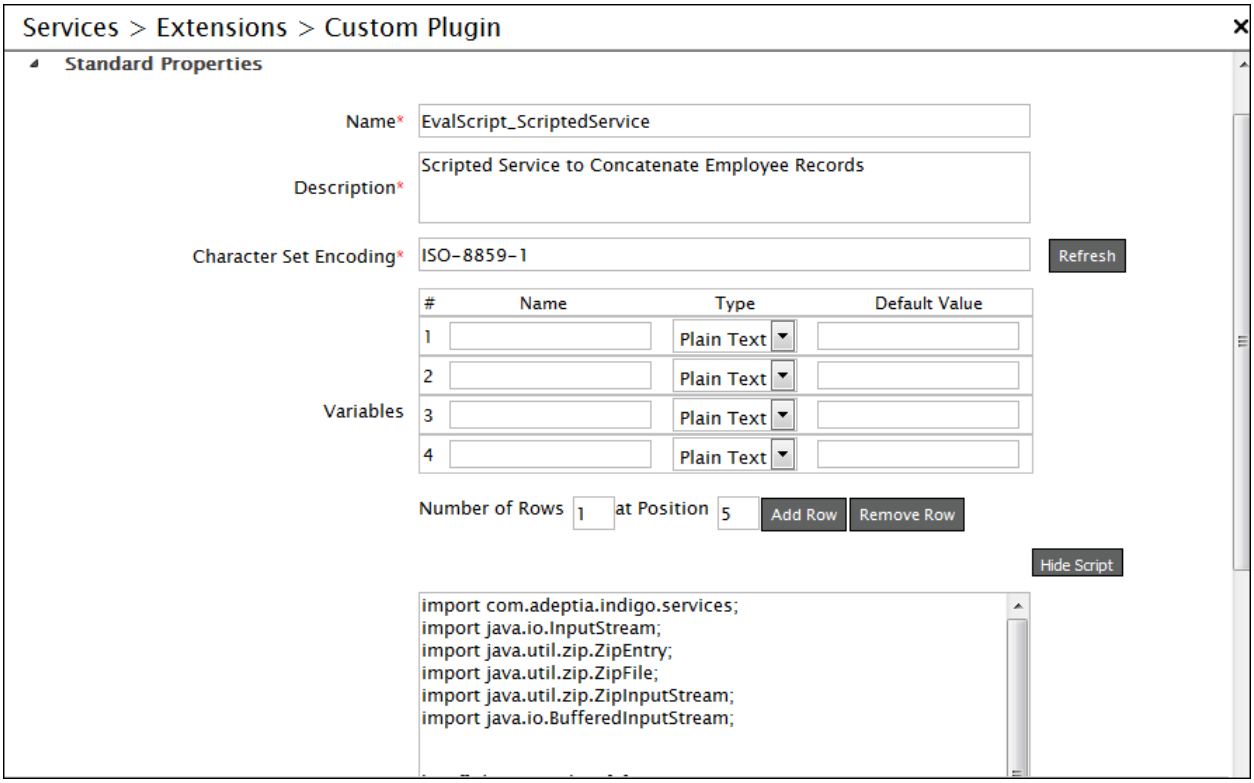

Figure 69: Edit *EvalScript_ScriptedService* Activity

The following tables defines the fields on this screen.

Table 14: Details of Fields on Edit Custom Plugin Screen

Field Name	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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Services** > **Schema**, and then click **Text**. The *Manage Text Schema* screen is displayed with the list of existing Text Schemas (refer to Error! Reference source not found.).
3. Select the required Text schema activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Text schema activity in edit mode (see Figure 70).

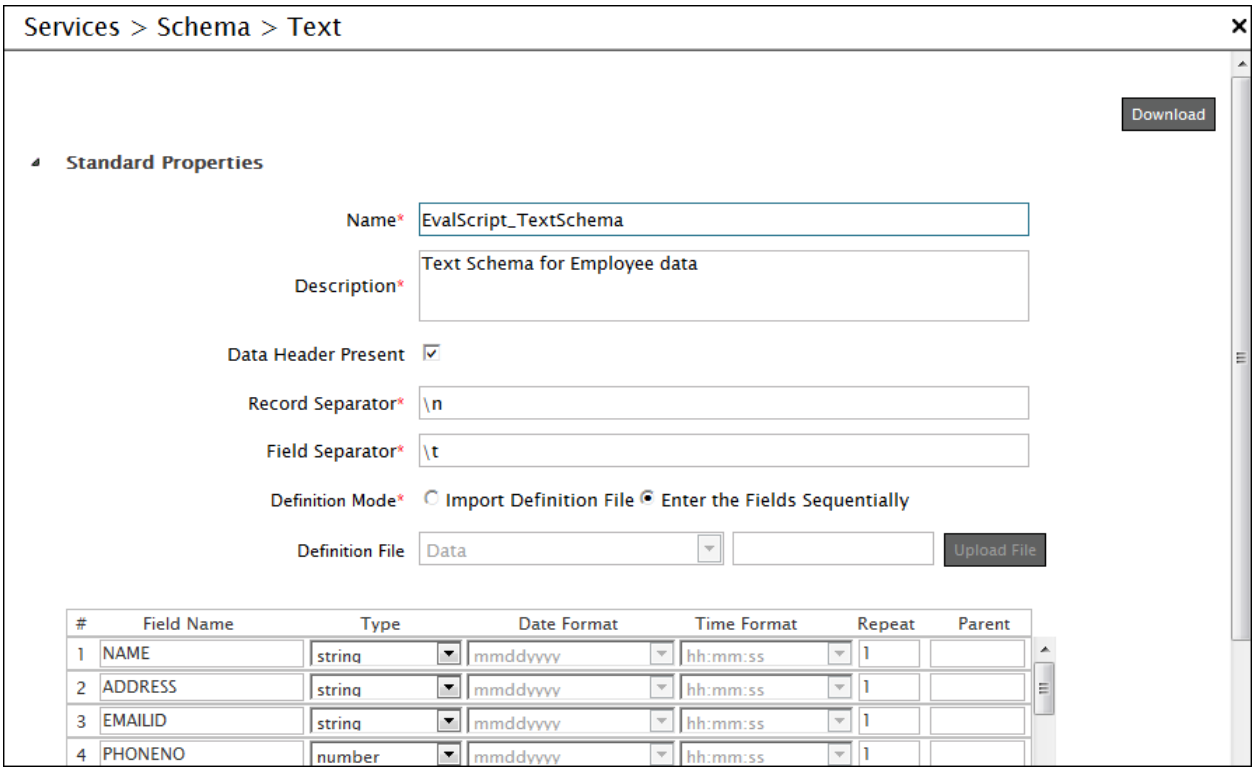

Figure 70: Edit *EvalScript_TextSchema* Activity

The following table defines the fields on this screen.

Table 15: Details of Fields on Edit Text Schema Screen

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

Field Name	Description
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: • Use Definition File • 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	String: String can be used for any type of data. Number: Contains numbers 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 (Chocolate20perpack) 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. 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.
7. Enter comments in the **Add Comments** field.

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 178).
2. Select the type of schema to test from the **Type** dropdown 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation 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 **Error! Reference source not found.**).
3. Select the required Data Mapping activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Data Mapping activity in edit mode (see Figure 71).

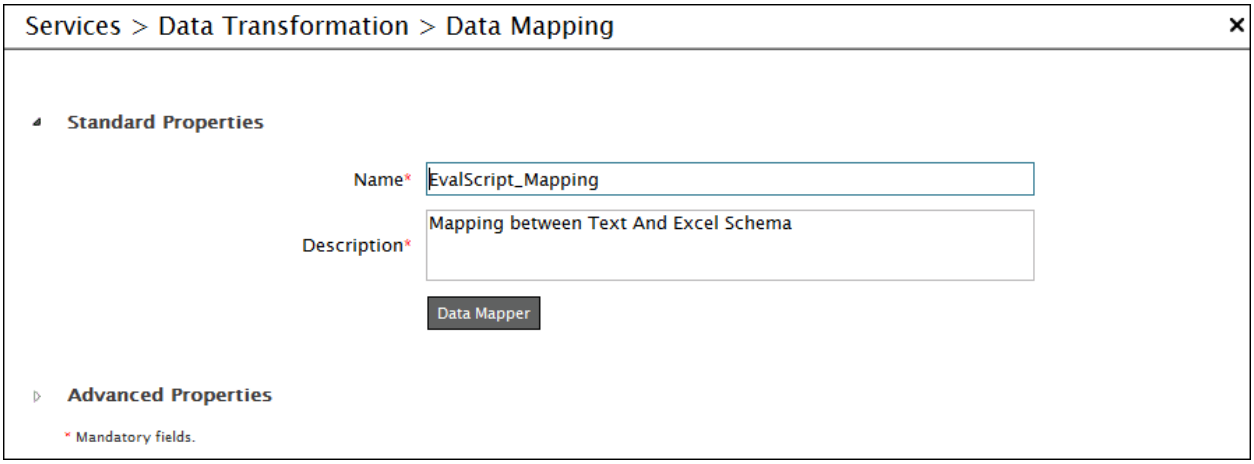

Figure 71: Edit *EvalScript_Mapping* Activity

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 72).

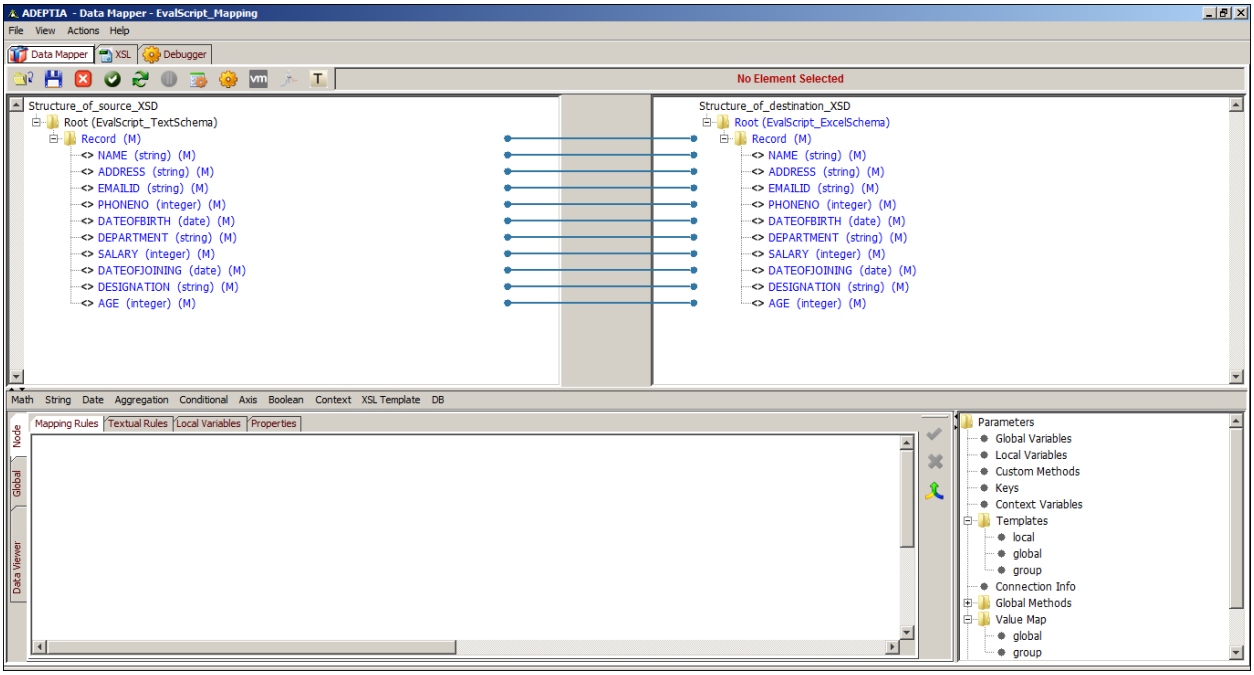

Figure 72: *EvalScript_Mapping* Activity in Data Mapper

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 181).
9. Enter the comments in the Specify comments for mapping object (object name) field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.

2. Go to > **Services** > **Schema**, and then click **Excel**. The *Manage Excel Schema* screen is displayed with the list of existing Excel Schemas (refer to **Error! Reference source not found.**).
3. Select the required Excel schema activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Excel schema activity in edit mode (see Figure 73).

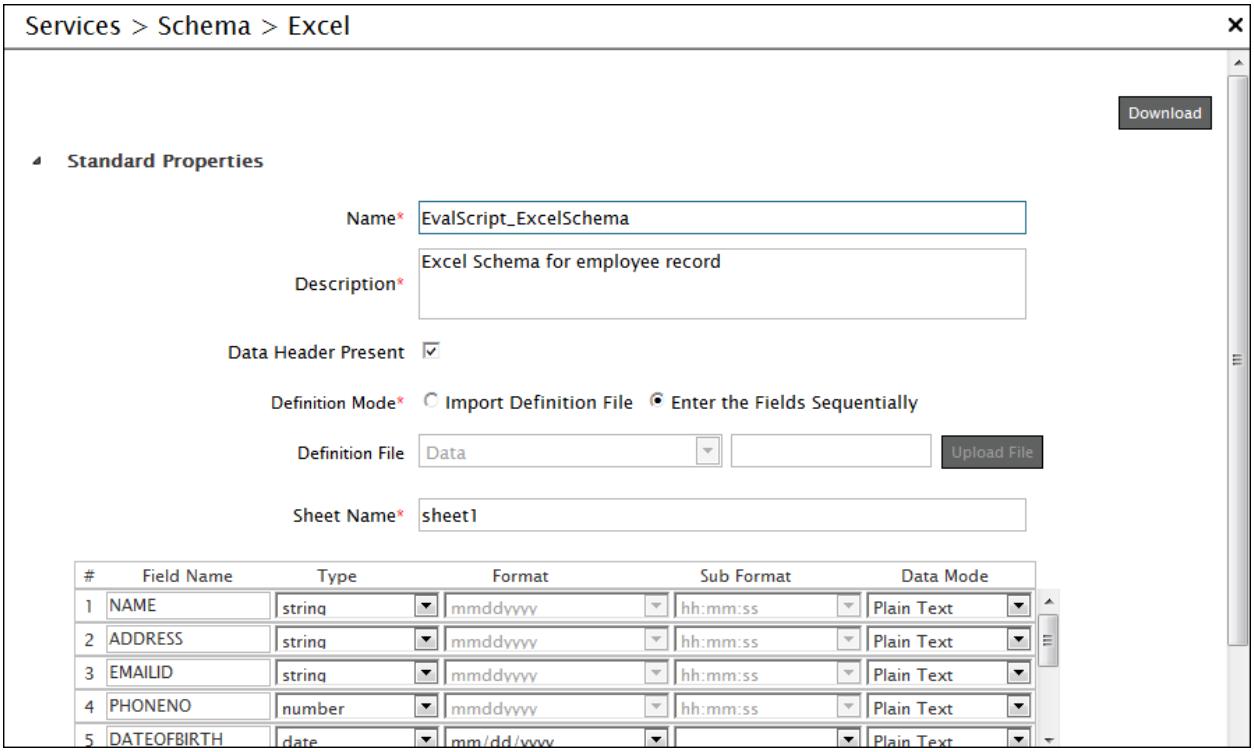

Figure 73: Edit *EvalScript_ExcelSchema* Activity

The following table defines the fields on this screen.

Table 16: Details of Fields on Edit Excel Schema Screen

Field Name	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.

Field Name	Description
Create Schema Definition	Schema can be defined using one of the following options: • Use Definition File • Enter the Field 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. Number: Contains numbers Date: Contains Date and Time Currency Contains Currency Value
Format	If data type is *Date*, select the data format from *Format* dropdown list.
SubFormat	If data type is *Date*, select time format from *SubFormat* dropdown 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.
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **File**.

   The *Manage File Target* screen is displayed with the list of existing File Target activities (refer to Figure 22).
3. Select the required File target activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective File target activity in edit mode (see Figure 74).

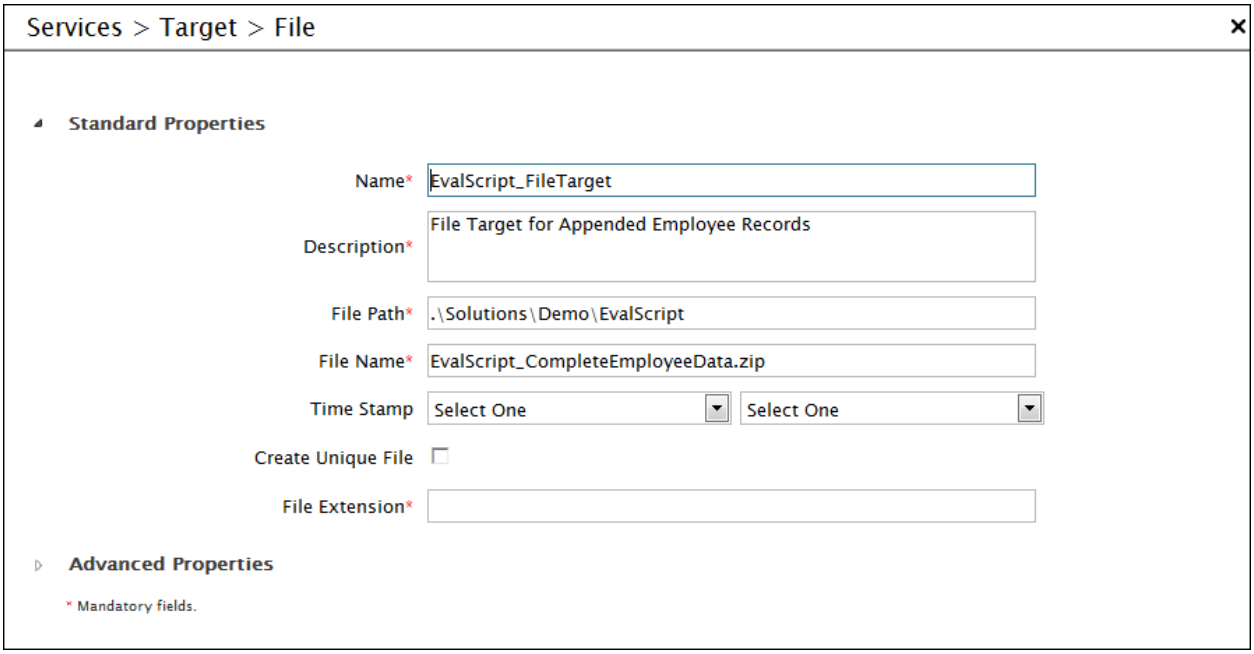

Figure 74: Edit *EvalScript_FileTarget* Activity

The following table defines the fields on this screen.

Table 6.6 Details of Fields on Edit File Target Screen

Field Name	Description
Name	Name of the File Target
Description	Description of the File Target
File Path	Target Path and the target file. For example: ../../Solutions/Demo/EvalScript/
File Name	Name of target file. For example: 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.
7. Enter 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.

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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (refer to Figure 55).
3. Click the **Create New** link. The *Create Process Flow* screen is displayed (refer to Figure 56).
4. Enter the name and description of the Process Flow.
5. Select the logging level from the *Logging Level* dropdown list.
6. Select repository file retention from the **Repository File Retention** option.
7. Click the **Process Designer** button to open Process Designer.

For more information on how to create process flows and on *Process Designer* screen, refer to the Creating Process Flow section.

8. Click **OK** to synchronize.
9. Click **[+] Activities** in Repository View to expand the list of services and then click **[+] Source**. All 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 75).

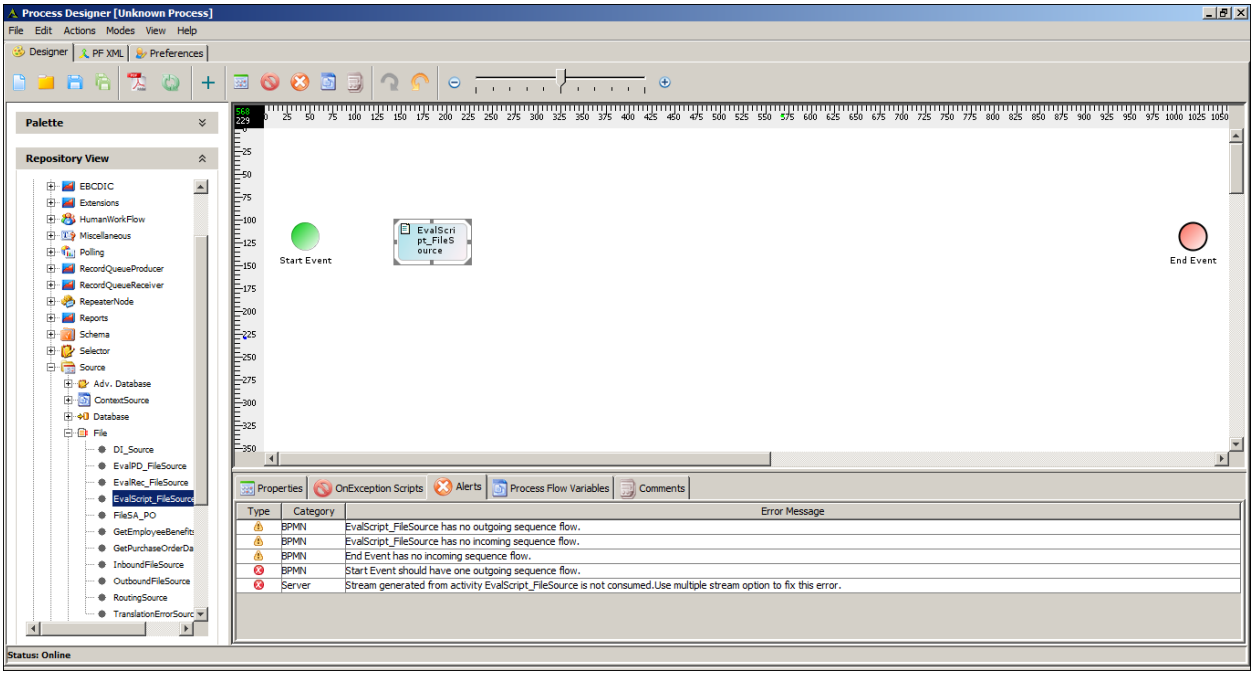

Figure 75: Drag File Source to Graph Canvas Area

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 View, and select **View Properties**. Properties of *EvalScript*_ExcelSchema are shown in the bottom pane (see Figure 76).

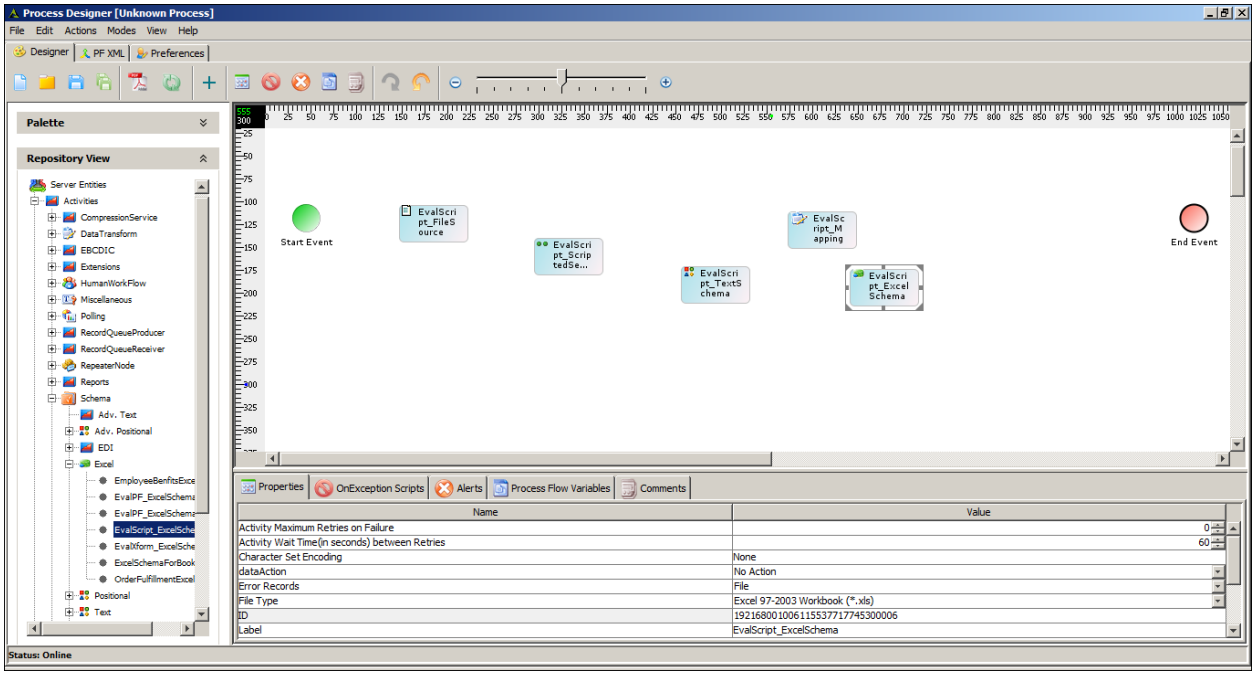

Figure 76: View Properties of *EvalScript_ExcelSchema* Activity

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 77).

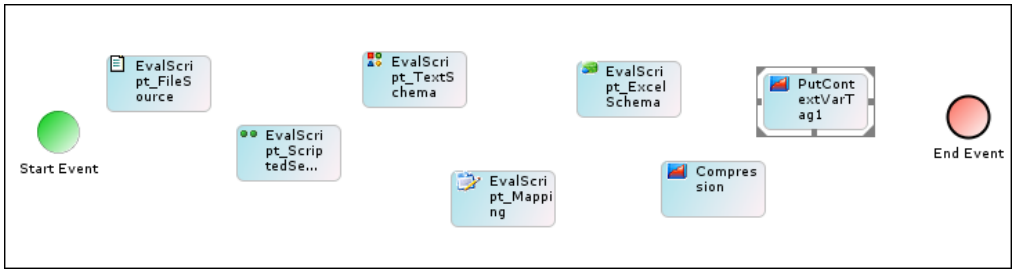

Figure 77: Drag Put-Context Var to Graph Canvas Area

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 78).

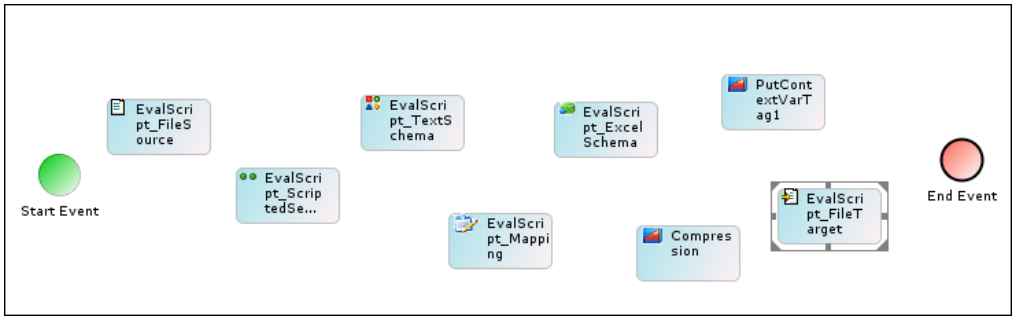

Figure 78: Connect Start Event to File Source

25. Similarly, connect all other activities as shown in Figure 79.

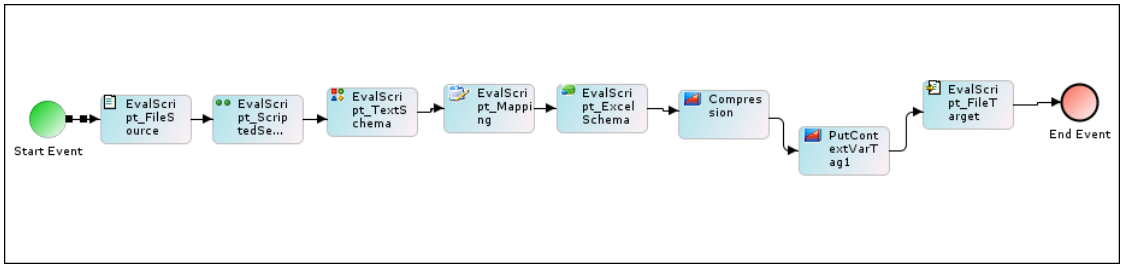

Figure 79: Connect all Activities

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 80).

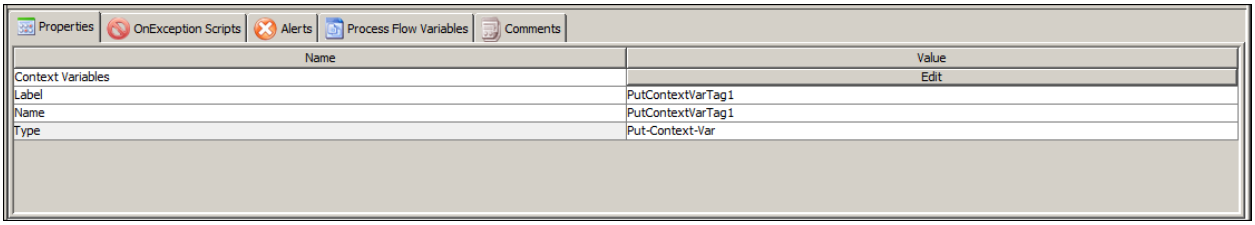

Figure 80: View Put-Context Var Properties

27. Click **Edit** from the value field of the *context Variable* properties. The *Edit Context Variable* screen is displayed (see Figure 81).

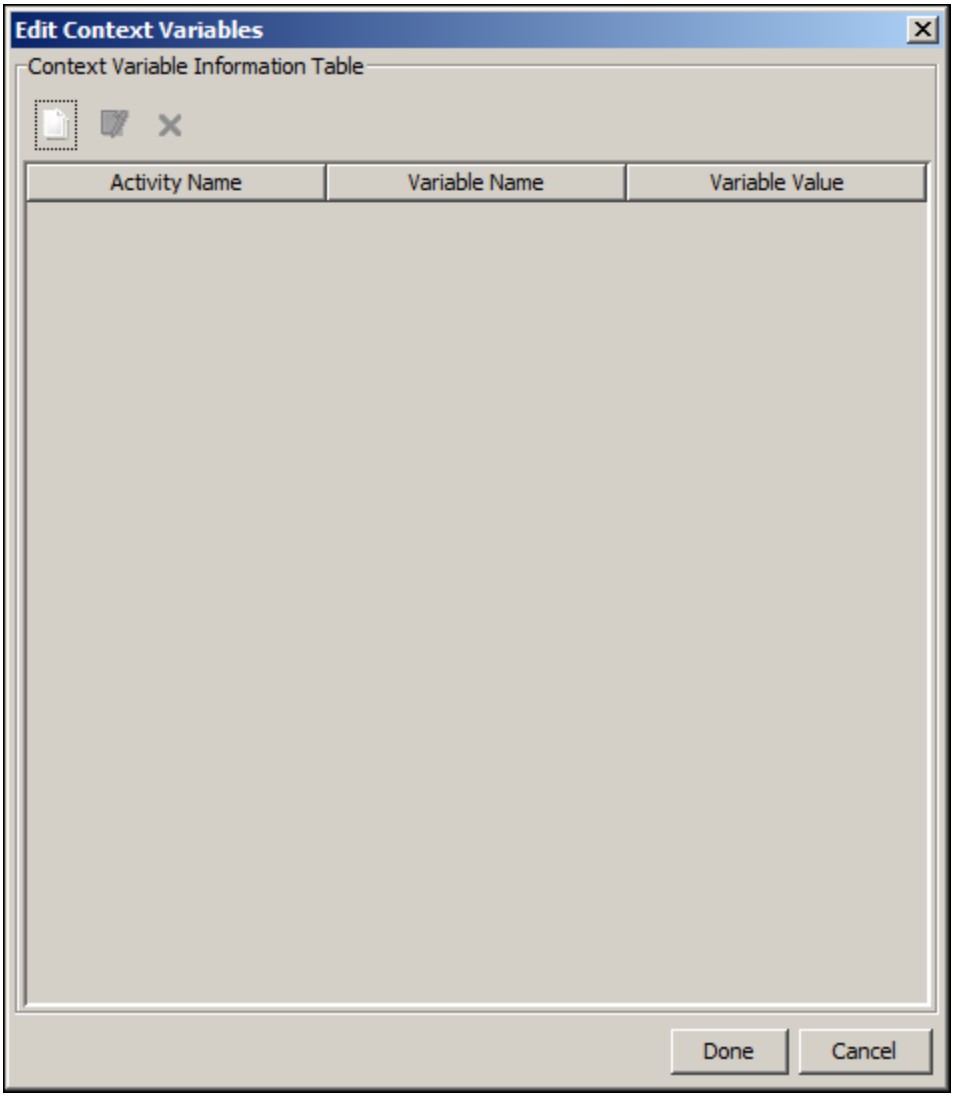

Figure 81: Edit Context Variable

28. Click **New variable** () button. The **Context Variable Information** dialog box is displayed (see Figure 82).

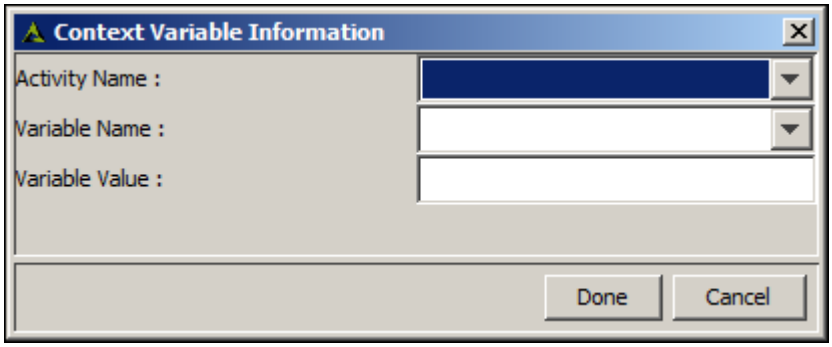

Figure 82: Add Context Variable

29. In the **Activity** dropdown list select the *EvalScript_FileTarget* activity.

30. Select *filename* from the Variable Name drop-down list.
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 83).

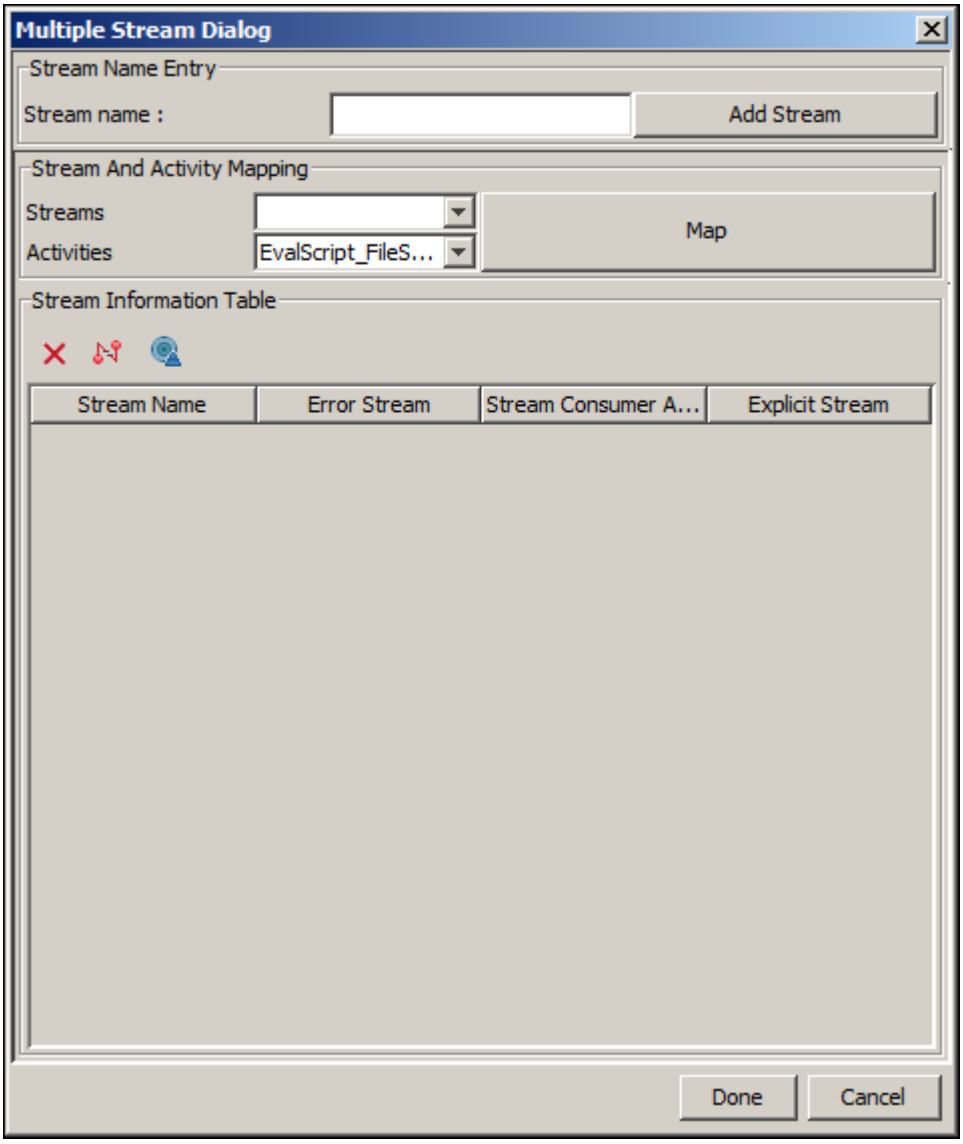

Figure 83: Multiple Stream Dialog Box

34. Select **EvalScript_FileTarget** from the **Activities** dropdown list and then click the **Default Stream** ( ) button.
35. Click the **Map** button. A stream between *Compression* and *EvalScript_FileTarget* is created (see Figure 84).

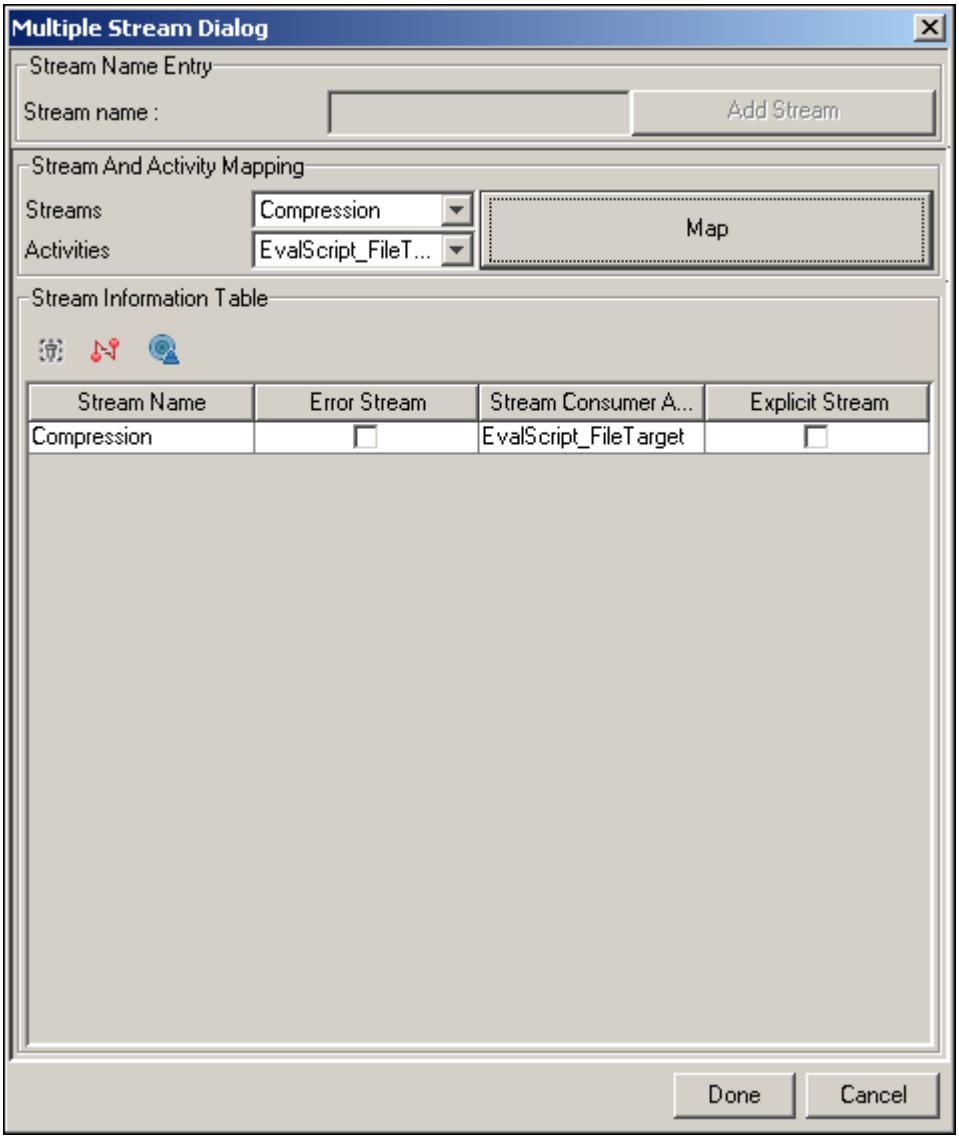

Figure 84: Define Data Stream

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 85).

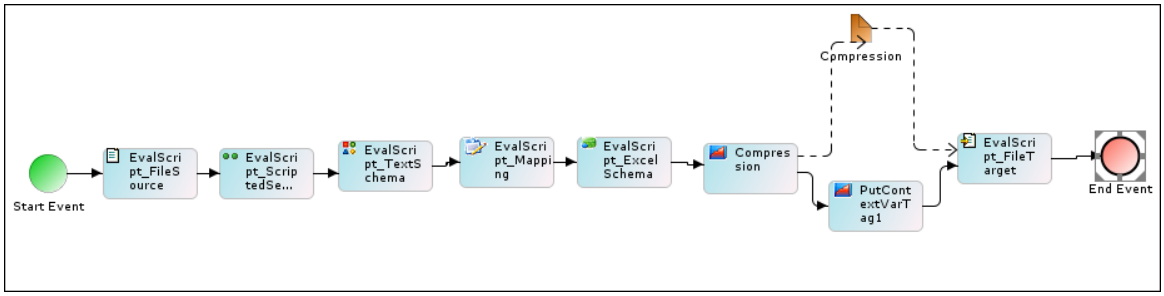

Figure 85: Data Stream Created

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 193).

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

4

PROCESS DESIGNER PROCESS FLOW

This section describes the Process Designer Process Flow.

INTRODUCTION

In the Adeptia Suite this process flow is available in:

EBIM Suite	BPM Suite	ESB Suite	ETL Suite
	√	√	√

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:

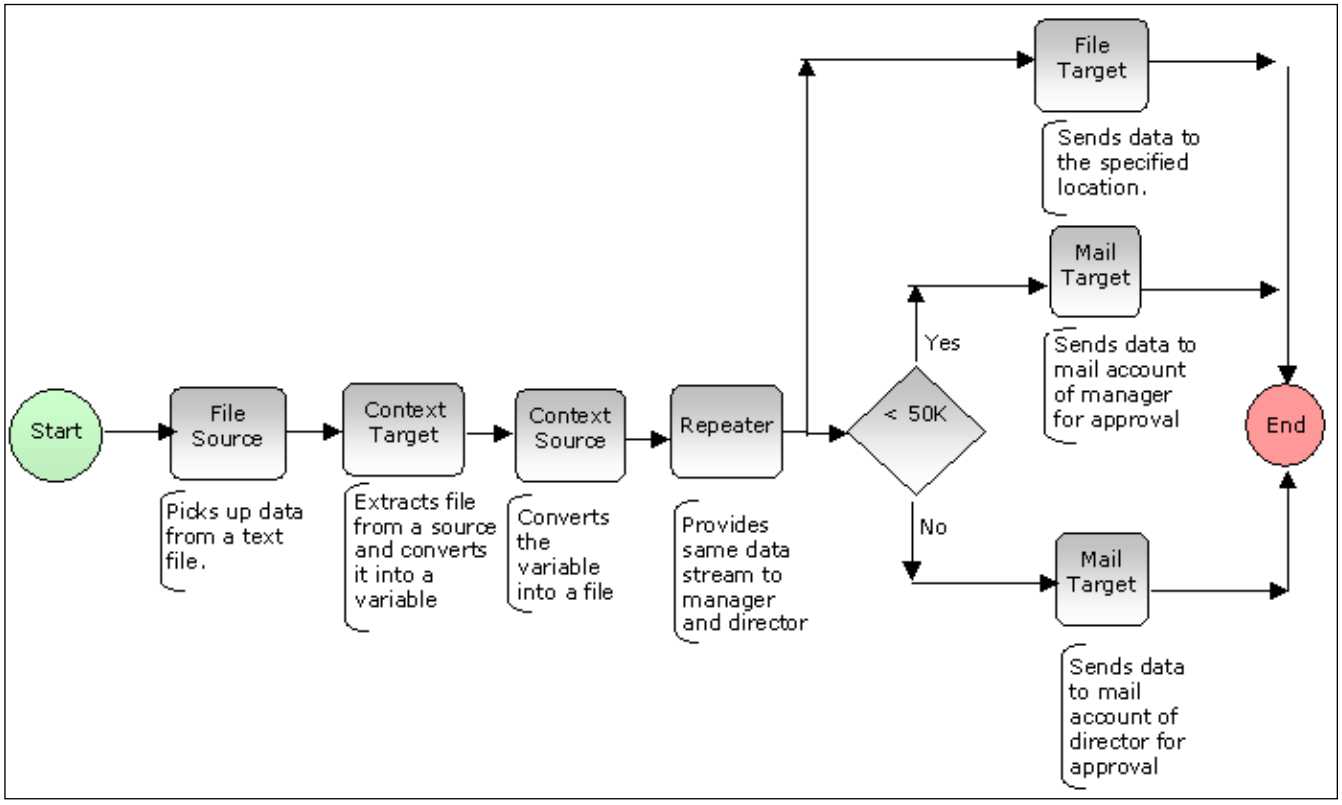

Figure 86: Flow Chart to show Process Flow

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 not 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 17: 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed (see Figure 2).
3. Click the *Execute* button, which is adjacent to *EvalPD_ProcessFlow*. A confirmation dialog box is displayed (see Figure 3).
4. Click the button **Go to Logs** to view the status of the Process Flow execution. The Process Flow Logs are displayed (see figure).

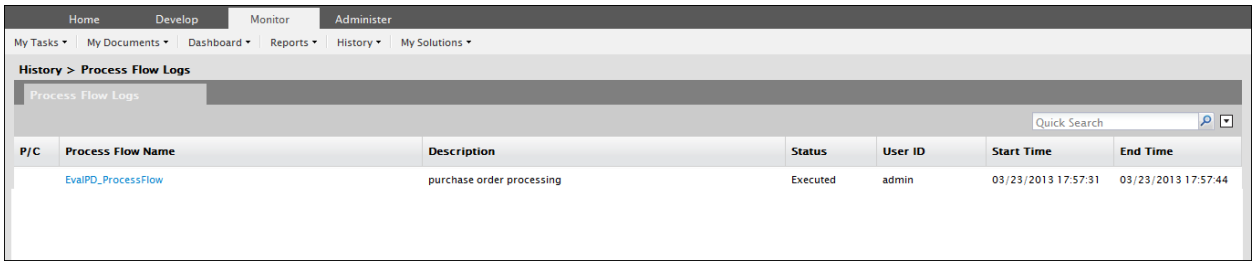

Figure 87: View Process Flow Log Details

5. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 88).

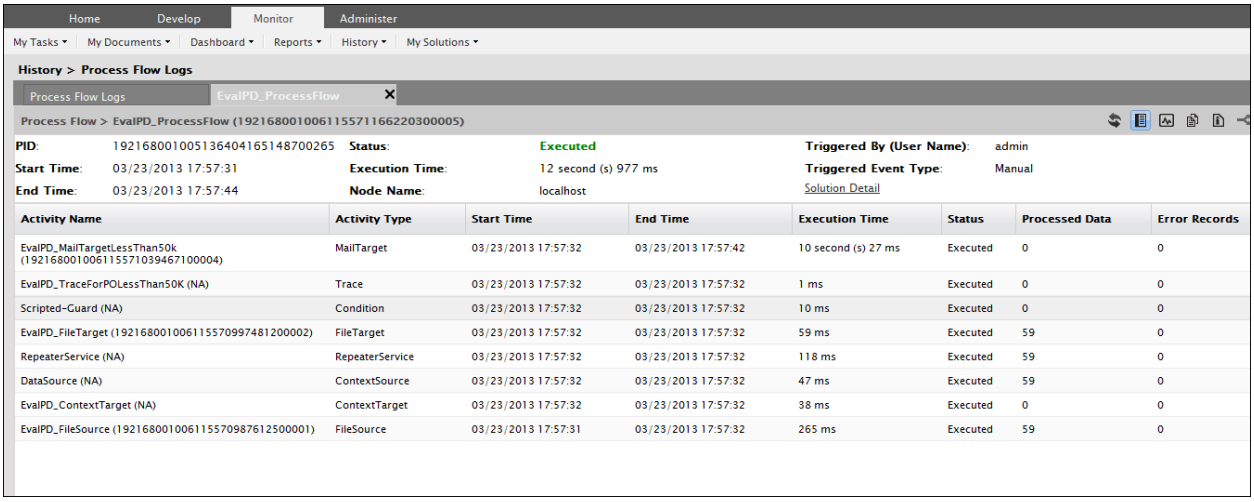

Figure 88: View Process Flow Log Details

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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Source** and then click **File**.

   The *Manage File Source* screen is displayed with the list of existing File Source activities (refer to Figure 65).
3. Select the required File source activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective File source activity in edit mode (see Figure 89).

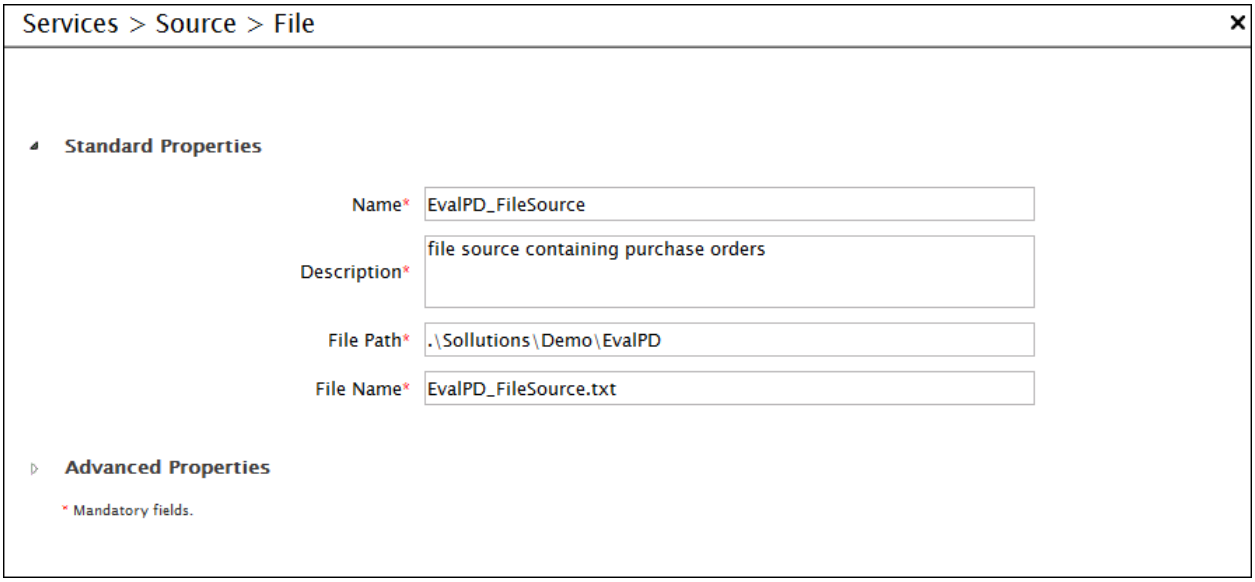

Figure 89: Edit *EvalPD_FileSource* Activity

The following table defines the fields on this screen.

Table 18: Details of Fields on Edit File Source Screen

Field Name	Description
Name	Name of the File Source
Description	Description of the File Source
File Path	Path of the source file. For example: ../../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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **File**.

   The *Manage File Target* screen is displayed with the list of existing File Target activities (refer to Figure 22).
3. Select the required File target activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective File target activity in edit mode (see Figure 90).

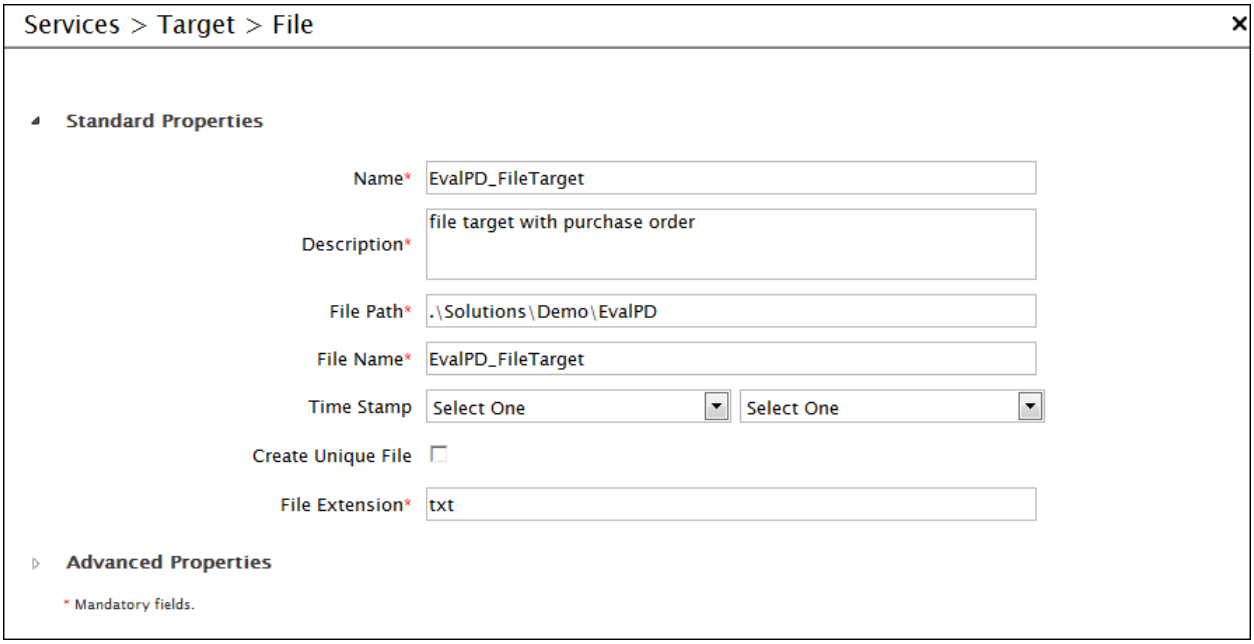

Figure 90: Edit *EvalPD_FileTarget* Activity

The following table defines the fields on this screen.

Table 19: Details of Fields on Edit File Target Screen

Field Name	Description
Name	Name of the File Target
Description	Description of the File Target

Field Name	Description
File Path	Path of the target file. For example: `../../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 175).
7. Enter 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.

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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **Mail**.

The *Manage Mail Target* screen is displayed with a list of existing Mail Target activities (see Figure 91).

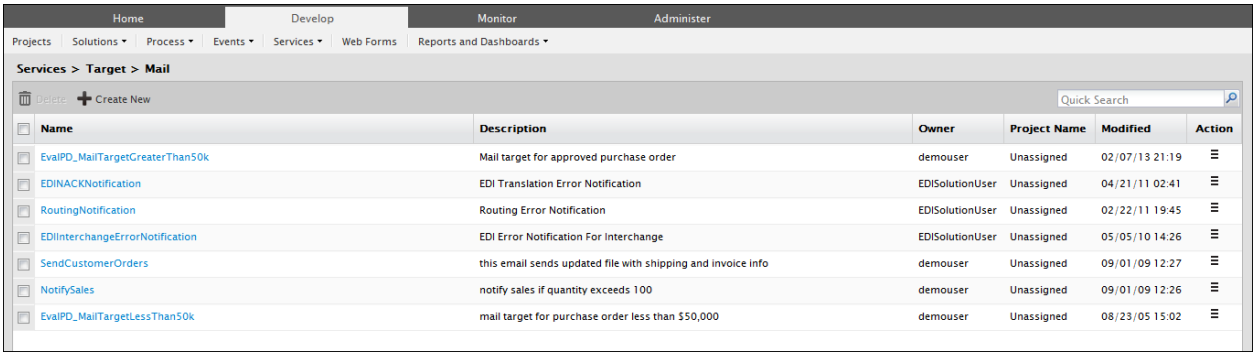

Figure 91: Manage Mail Target

3. Select the required Mail target activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Mail target activity in edit mode (see Figure 92).

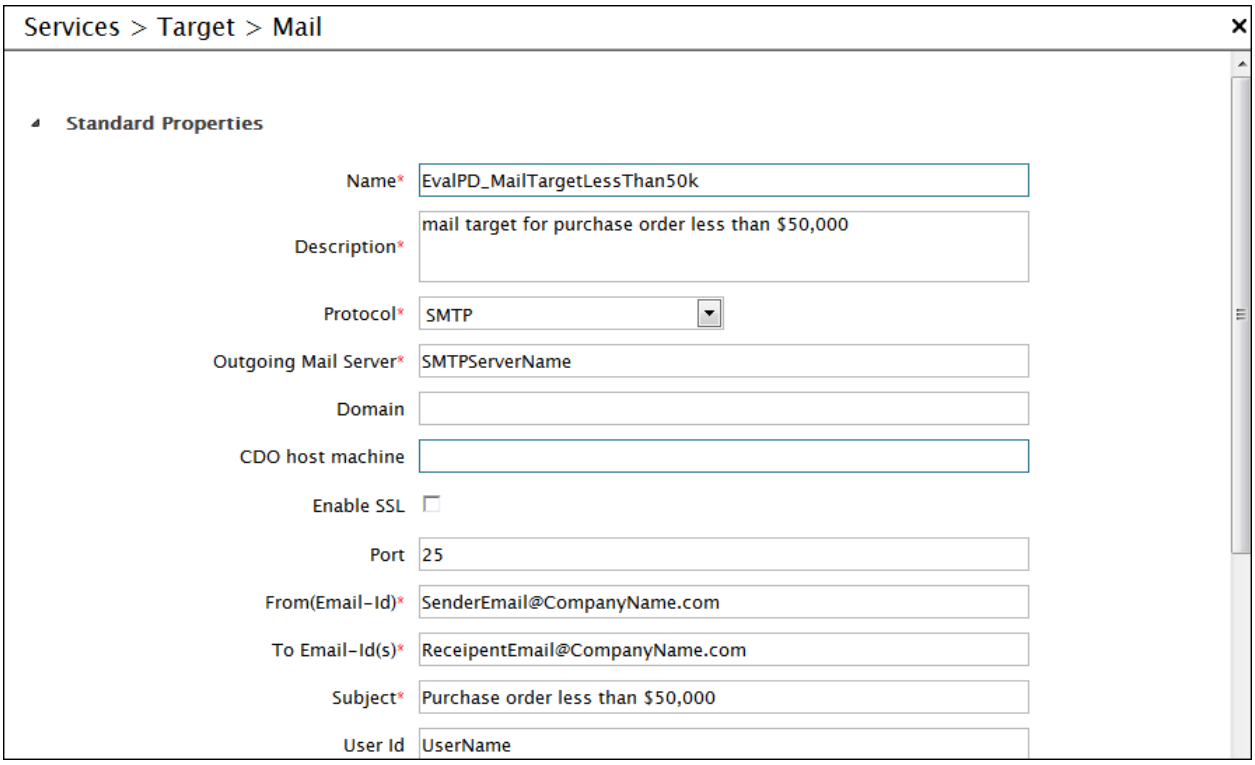

Figure 92: Edit *EvalPD_MailTargetLessThan50K* Activity

The following table defines the fields on this screen.

Table 20: Details of Fields on Edit Mail Target Screen

Field Name	Description
Name	Name of the Mail Target.
Description	Description of the Mail Target.

Field Name	Description
Outgoing Mail(SMTP) Server	Name or IP address of the Outgoing Mail or SMTP 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.
7. Enter 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (refer to Figure 55).
3. Click the **Create New** link. The *Create Process Flow* screen is displayed (refer to Figure 56).
4. Enter the name and description of the Process Flow.
5. Click the **Process Designer** button to open Process Designer. The *Process Designer* screen is displayed (refer to Figure 57).

   For more information on how to create process flows and on *Process Designer* screen, refer to the Creating Process Flow section.

6. Click **OK** to synchronize.
7. Click **[+] Activities** in Repository Panel, to expand the list of services and then click **[+] Source**. All items in the **Source** category are displayed.
8. Click the **[+] File Source**. A list of existing File Source activities that is displayed.

9. Select **EvalPD_FileSource** and drag it to the Graph Canvas Area (see Figure 93).

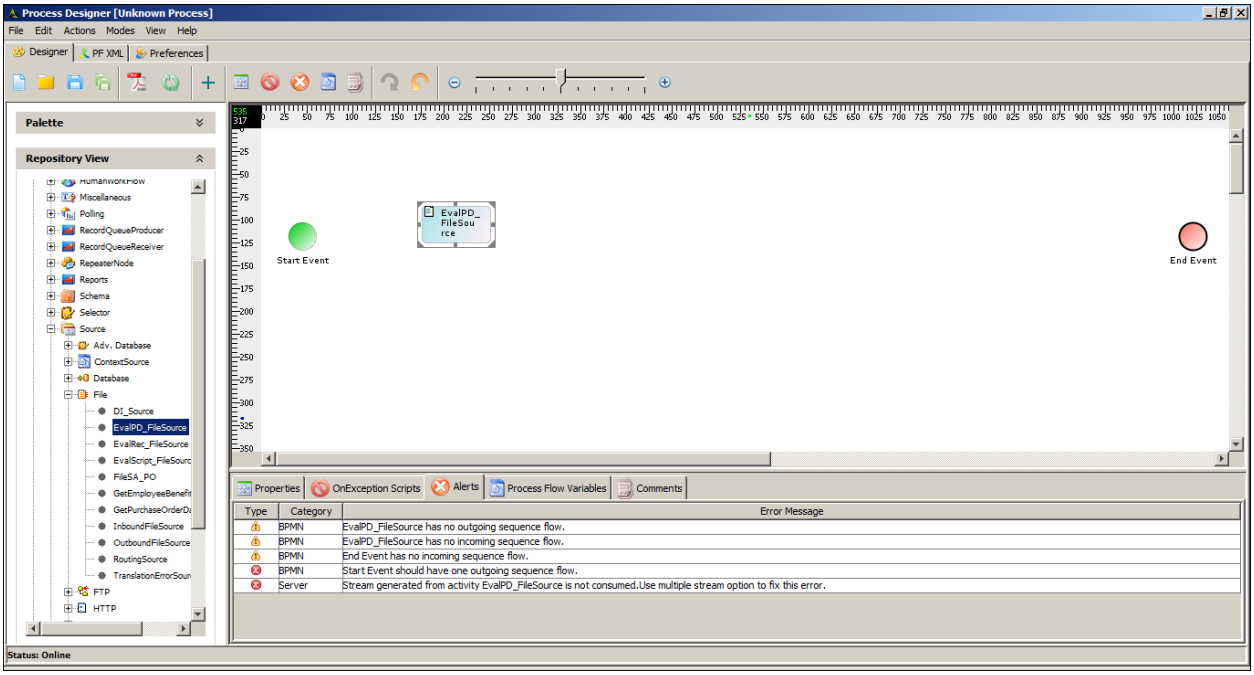

Figure 93: Drag File Source Activity to Graph Canvas Area

10. 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 94).

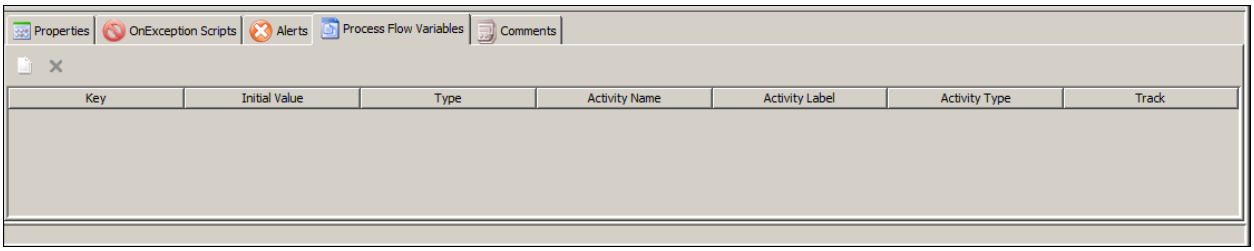

Figure 94: Process Flow Variables Panel

11. Click the **New Process Flow Variable** ( ) button. The Process flow variable entry dialog box is displayed (see Figure 95).

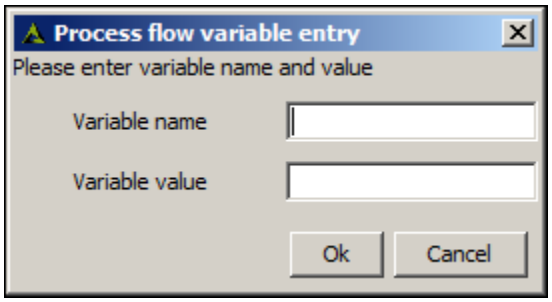

Figure 95: Create Process Flow Variable

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

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

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

15. Drag the Context Target node to the Graph Canvas Area (see Figure 96).

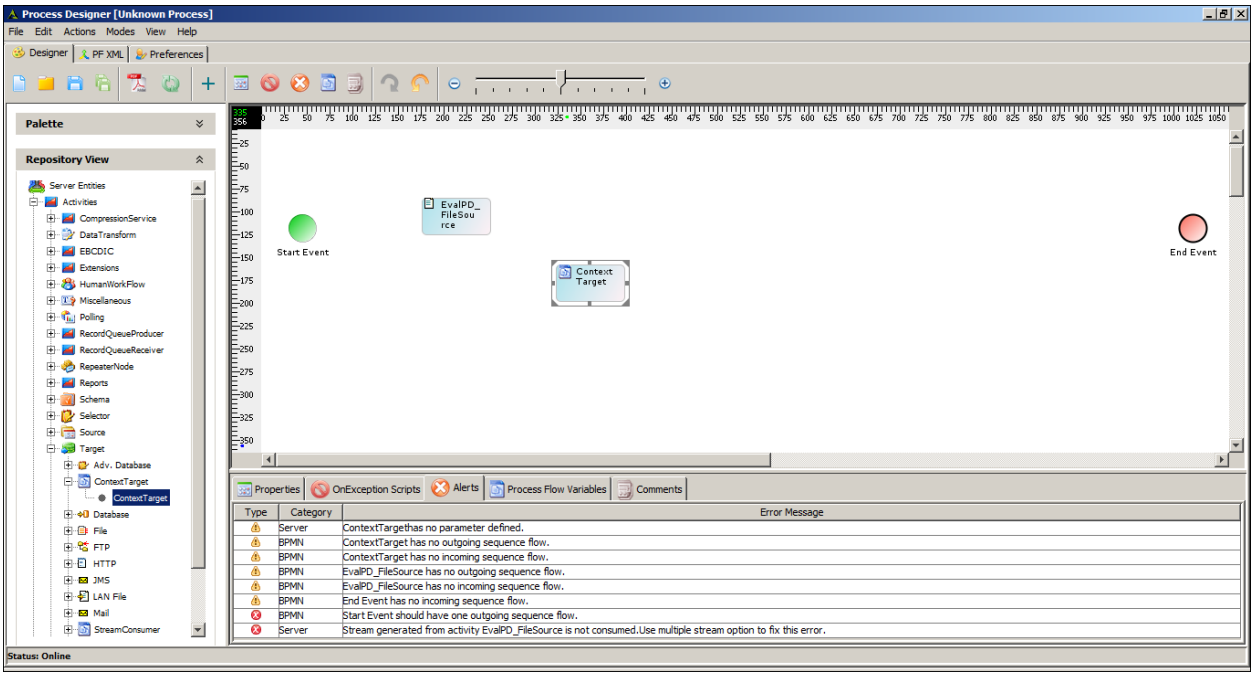

Figure 96: Drag Context Source Activity to Graph Canvas Area

16. Right-click the Context Target and select **View Properties**. Properties of the Context Target are displayed in the Bottom Pane (see Figure 97).

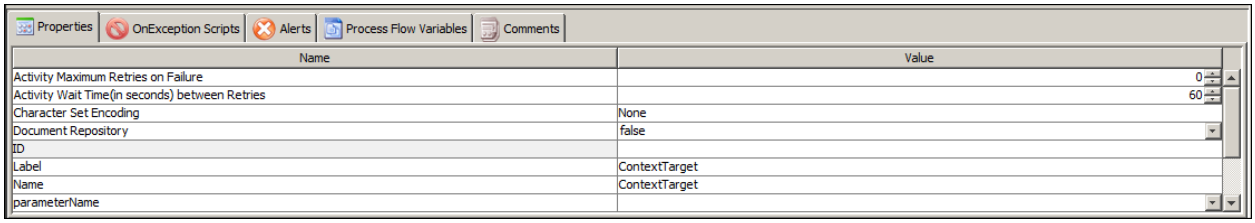

Figure 97: View Properties of Context Target Activity

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

18. Enter name of the Context Target activity i.e. Data (as given in the step 12) for the property **Name**.

19. Select *Data* from the *parameterName* dropdown list (see Figure 98).

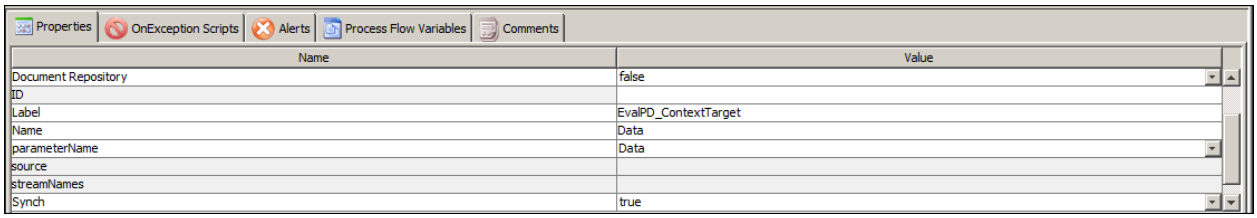

Figure 98: Enter Context Target Name

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

21. Right-click the Context Source and select **View Properties**. Properties of the Context Source are displayed in the Bottom Pane (see Figure 99).

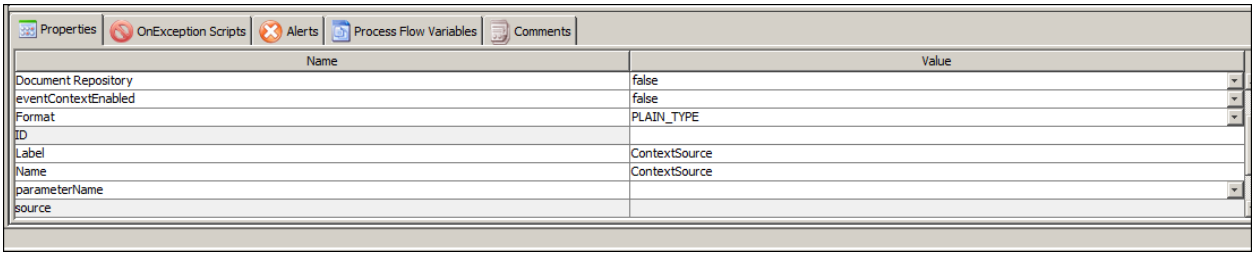

Figure 99: View Properties of Context Source activity

22. Enter the name of the Context Source as *DataSource* in the *Value* field of the *Name* property.

23. Change the value of Label from Context Source to EvalPD_ContextSource.

24. Select *Data* from the *parameterName* dropdown list (see Figure 100).

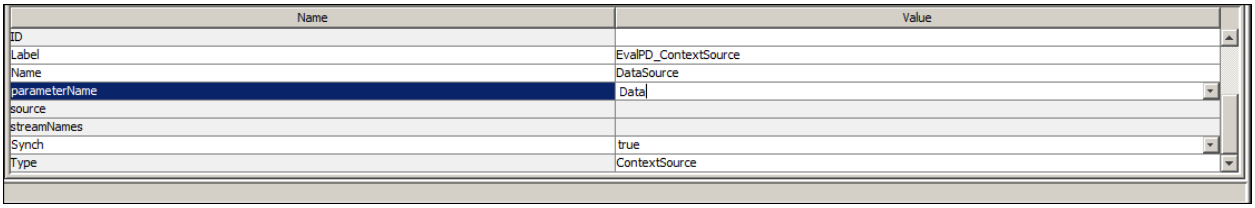

Figure 100: Enter Context Source Name

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

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

27. To select a BPMN Gateway, click the **Gateway** () icon in the Palette and drag it to the Graph Canvas Area (see Figure 101). In this Process Flow, **Gateway** is used to check the purchase amount and to decide which mail target is to be used.

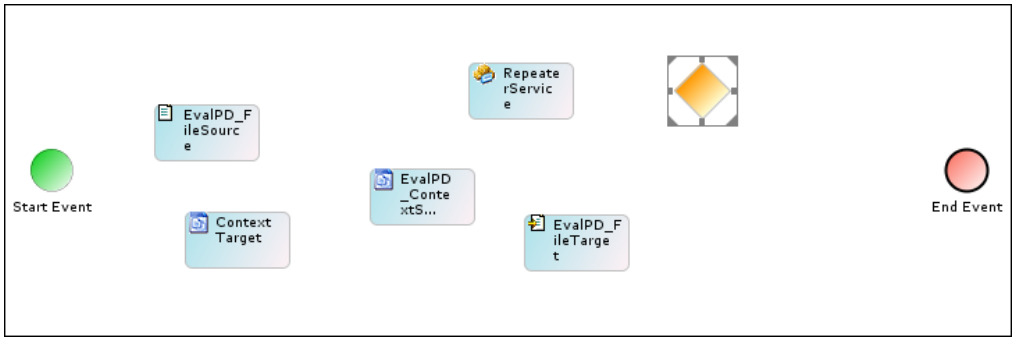

Figure 101: Drag BPMN Gateway Element to Graph Canvas Area

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

> An Action name is always unique.

29. 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 102).

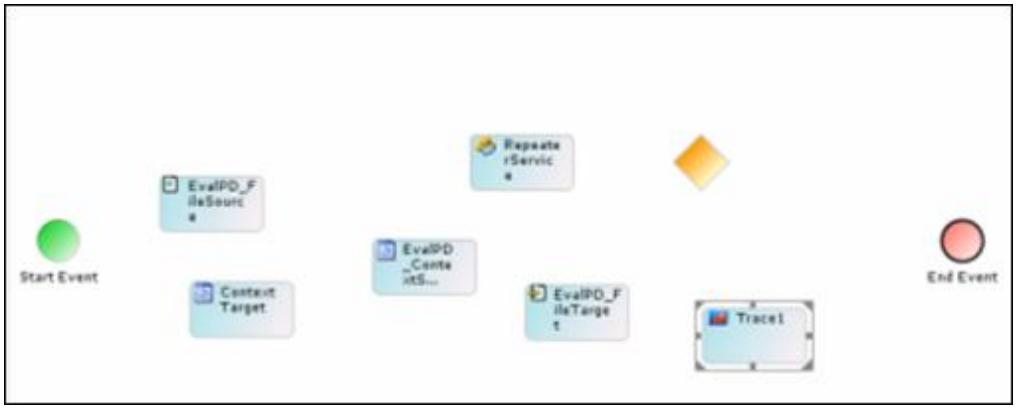

Figure 102: Drag Trace Action

30. 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 103).

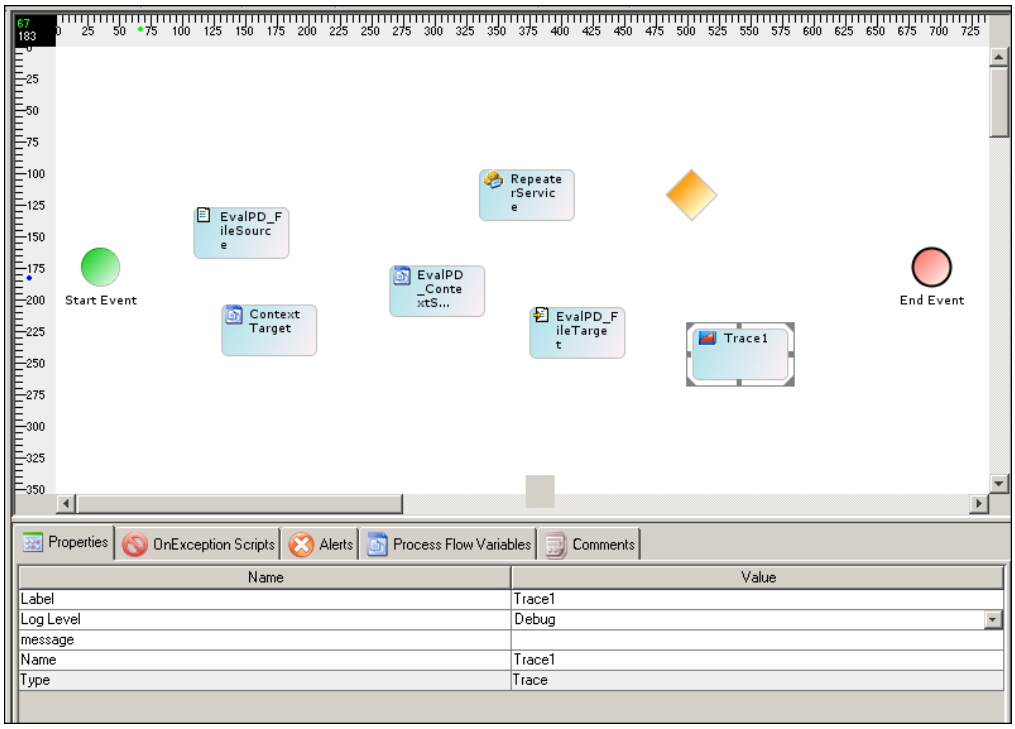

Figure 103: View Properties of Trace Action

31. 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

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

33. 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 104).

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

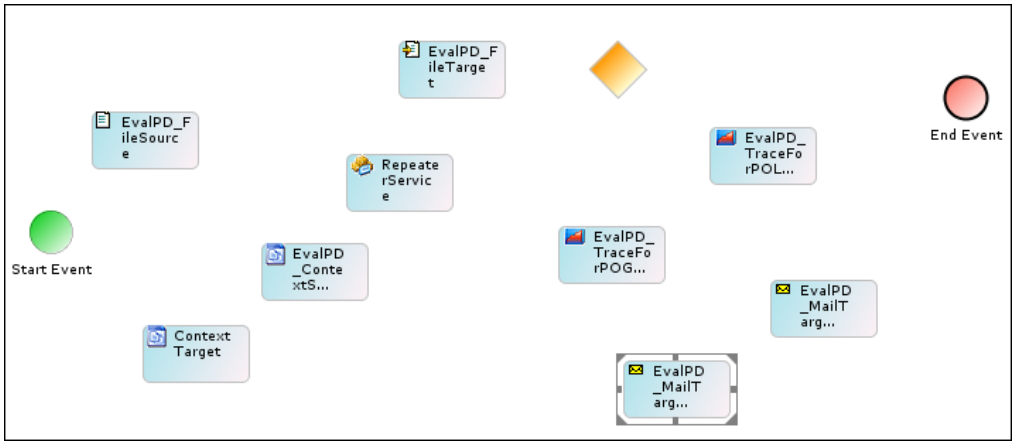

Figure 104: Drag another Trace Action

34. Click **[+] Target** and then **[+] Mail Target.** Select **EvalPD_MailTargetLessThan50K** activity and drag it to the Graph Canvas Area.
35. Select **EvalPD_MailTargetGreaterThan50K** activity and drag it to the Graph Canvas Area.
36. Once all the activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
37. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
38. To connect *Start Event* with *EvalPD_FileSource,* drag mouse pointer from *Start Event* to *EvalPD_FileSource* (see Figure 105).

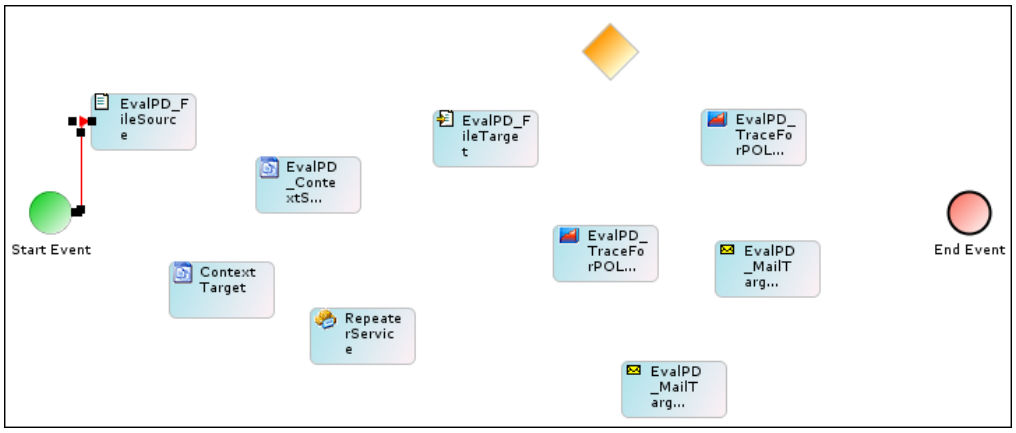

Figure 105: Connect Start Event to File Source Activity

39. Similarly, connect all other activities as shown in Figure 106.

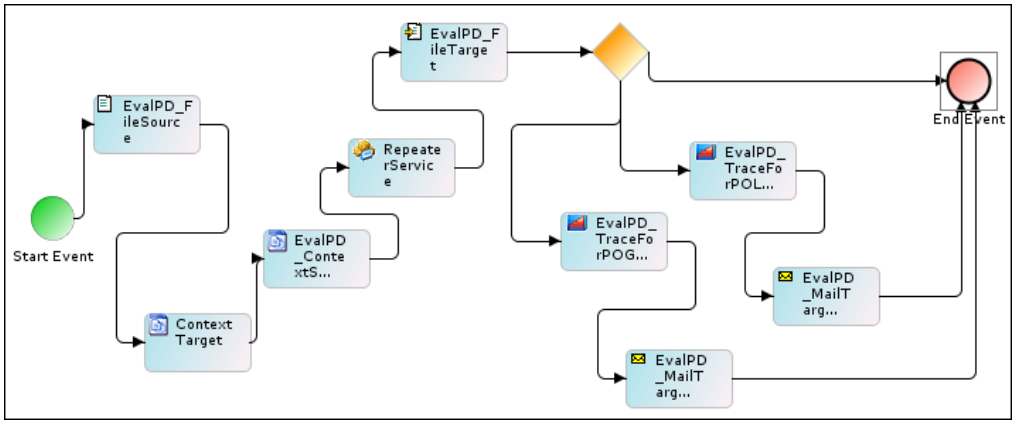

Figure 106: Connect all Activities

Figure 106 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 106, 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.

40. To create data stream, right-click **Repeater Service** and select **Multiple Stream**. The Multiple Stream dialog box is displayed (refer to Figure 83).
41. Enter number of streams (2) in the *Enter stream count* field, and then click the **Add Stream** button.
42. Select *1* from the *Streams* dropdown list, and select **EvalPD_FileTarget**. from the **Activities** dropdown list.

43. Click the **Map** button. A stream between *Repeater Service* and *EvalPD_FileTarget* is created (see Figure 107).

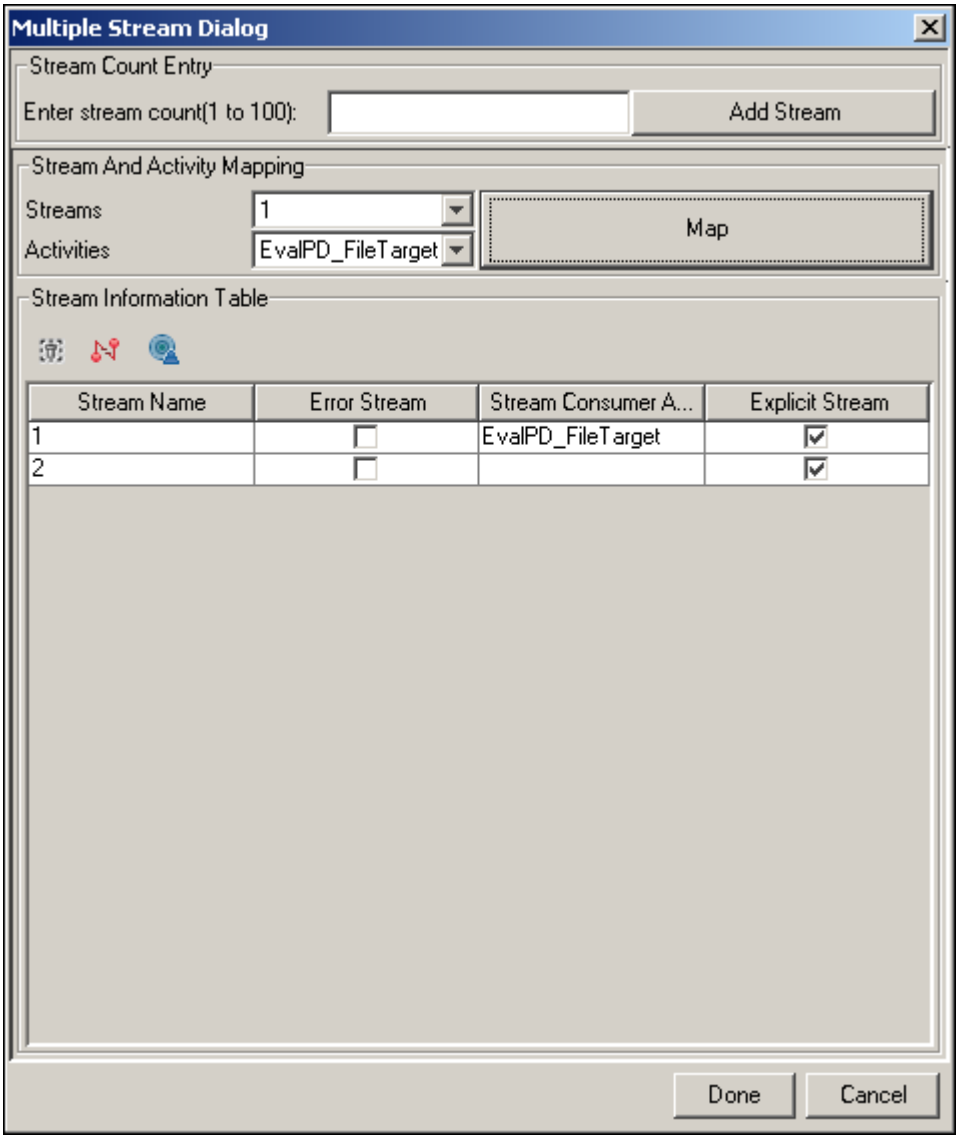

Figure 107: Create Stream

44. Ensure that the **Explicit Stream** checkbox is selected.

45. Select *2* from the *Streams* dropdown list, and select **EvalPD_MailTargetLessThan50K** from the **Activities** dropdown list, and click the **Map** button.

46. Select **EvalPD_MailTargetGreaterThan50K** from *Activities* dropdown list, and click the **Map** button. A dialog box is displayed (see Figure 108).

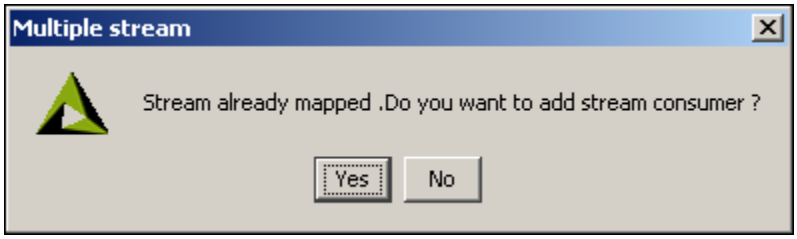

Figure 108: Add another Stream

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

48. Click the **Done** button to close the Multiple Stream Dialog box. Data stream created is shown in the Graph Canvas area (see Figure 109).

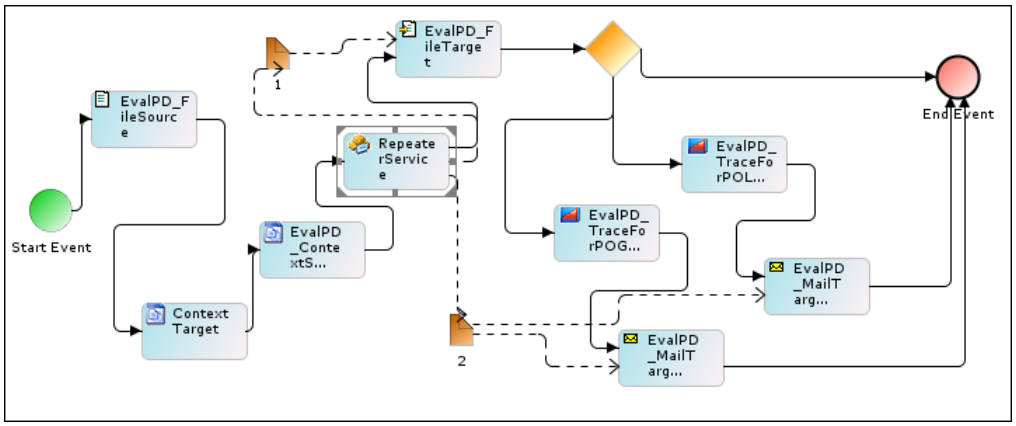

Figure 109: Multiple Streams Created

49. 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.

50. Click the **Edit Condition** in the Properties Panel. The Condition Wizard is displayed (refer to Figure 224).

51. Select **Java Condition** and click the **Next** button. The **Condition Type** screen is displayed (refer to Figure 225).

52. Enter the sample Java Code to specify the decision criteria and click the **Done** button (see Figure 110).

```
// 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 110: Sample JAVA Code

53. Similarly, define the following decision criteria on the Control Flow connecting decision node and *EvalPD_TraceForPOGreaterThan50K*. (see Figure 111).

```
// 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 111: Define Decision Criteria

54. 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 227).

55. 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.

56. 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 193).

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

> The comment should be at least 1 character in length.

58. Click **OK** to save the comments. This displays a screen confirming that the process flow has been created successfully.

59. Exit the Process Designer by clicking the **File** menu and selecting **Exit**.

7

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:

EBIM Suite	BPM Suite	ESB 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:

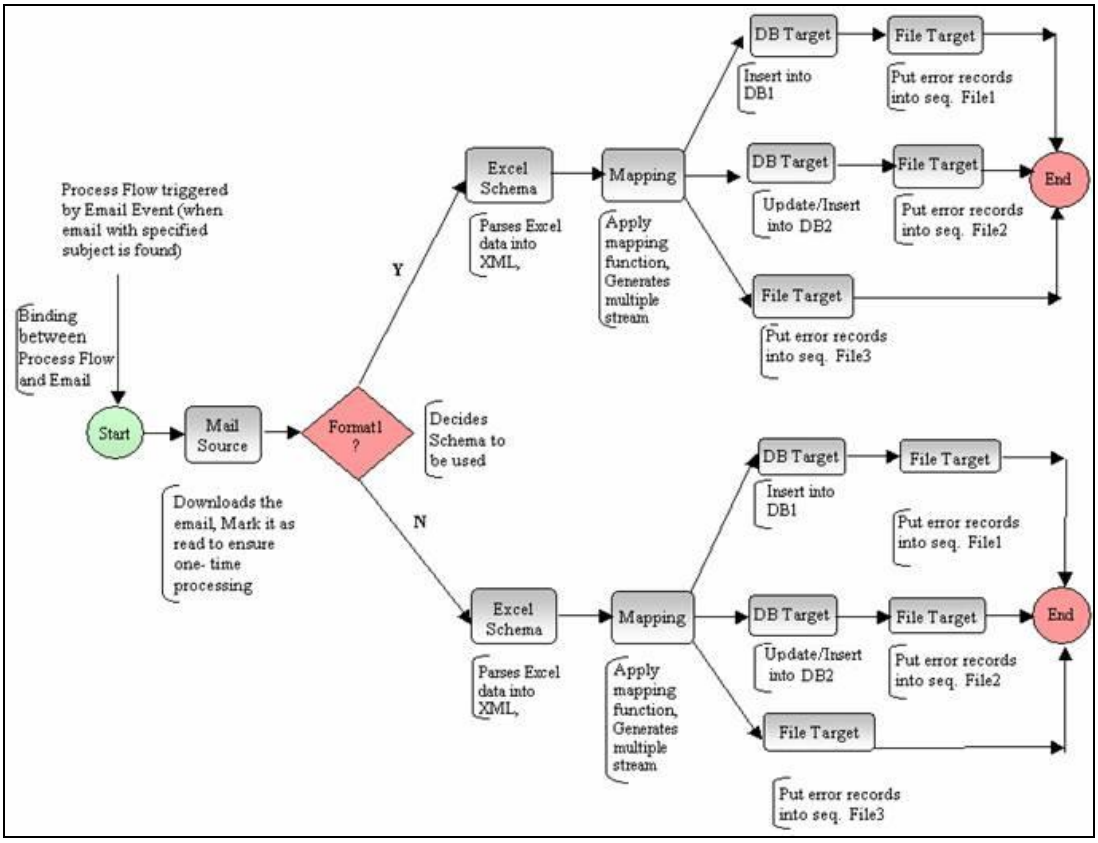

Figure 112: Flow Chart of the Process Flow

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 21 and Table 22 respectively.

Table 21: 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 22: 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 23: 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

Field Name	Description	Data Type
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Events > Mail**. The *Manage Mail Event* screen is displayed with the list of existing events (see Figure 113).

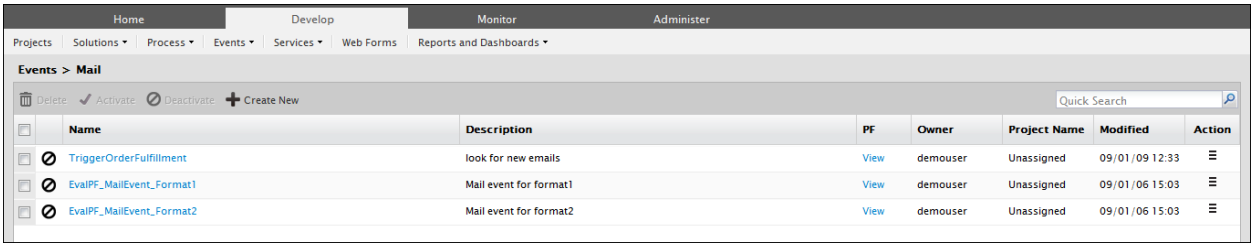

Figure 113: Manage Mail Event

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. On the Adeptia Suite homepage menu, click the **Monitor** tab.
2. Go to History > Process Flows Logs.

   The *Process Flow Logs* screen is displayed (see Figure 114).

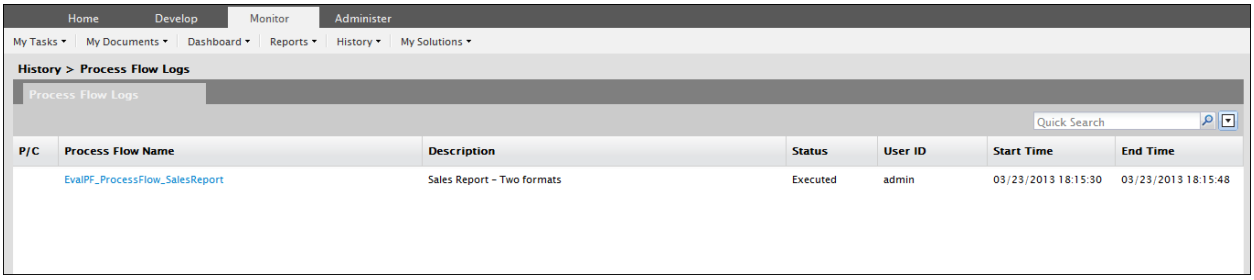

Figure 114: Process flow log

3. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 115).

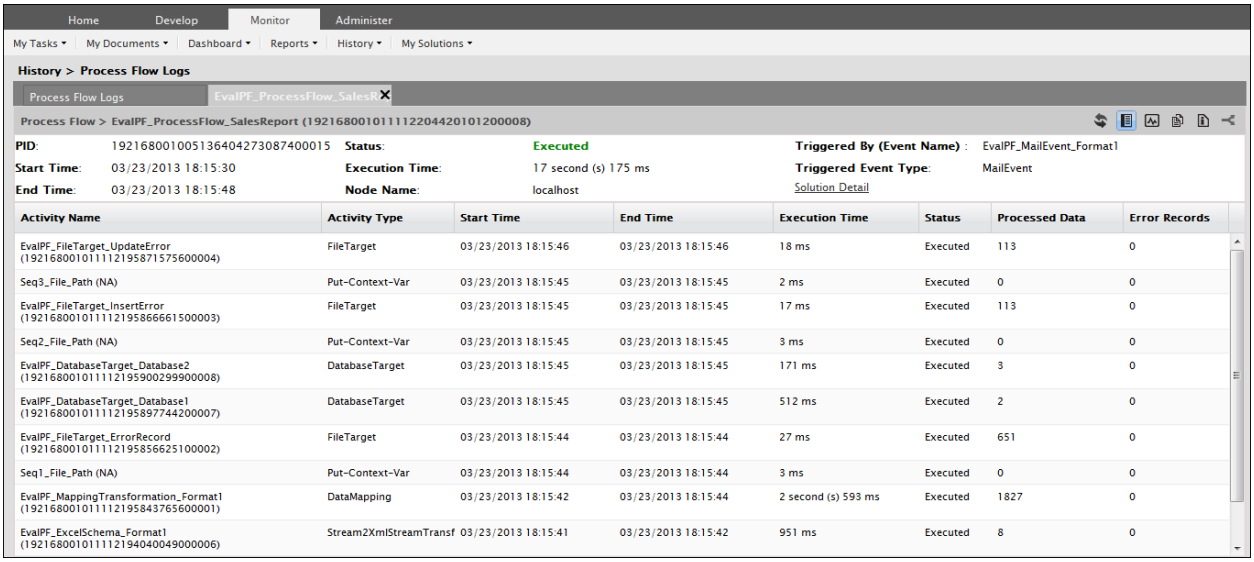

Figure 115: Searched Process Flows

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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Events > Mail**. The *Manage Mail Event* screen is displayed with the list of existing Mail Events (refer to Figure 113).
3. Select the required Mail Event activity and right-click the same to open the respective Mail event activity in edit mode (see Figure 116).

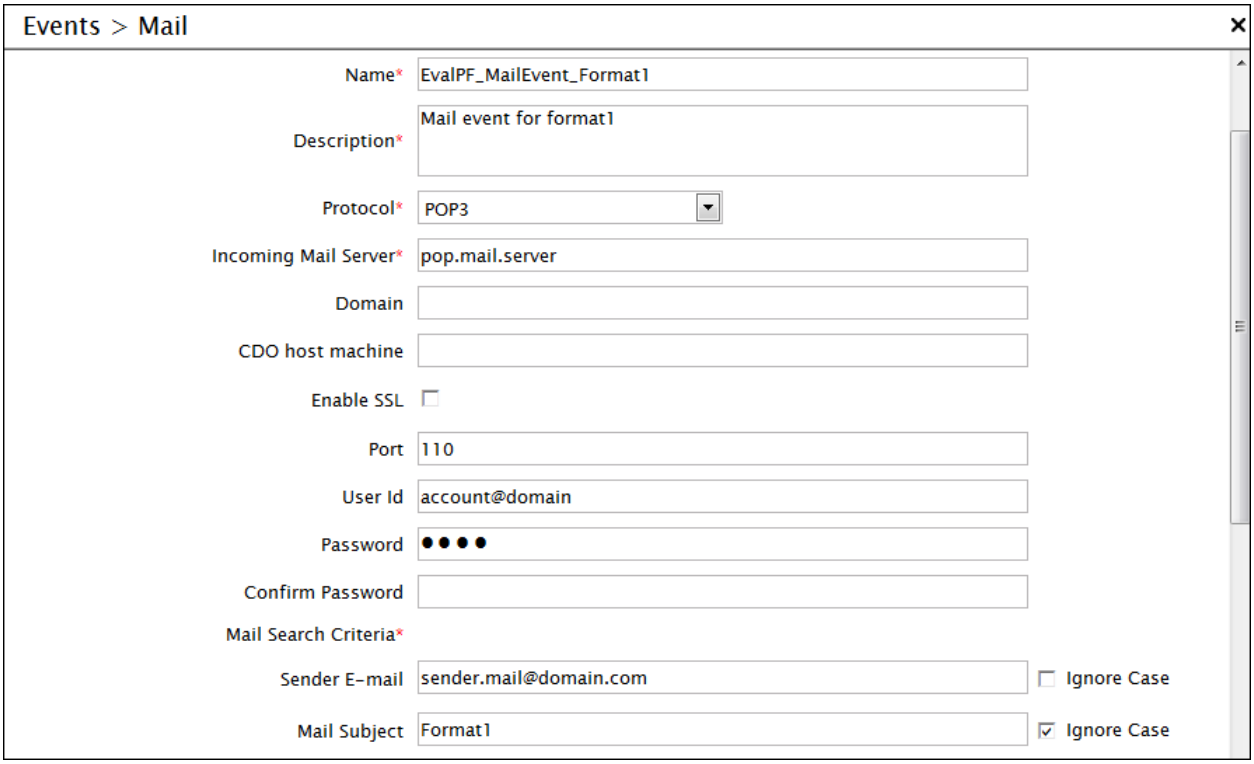

Figure 116: Edit *EvalPF_MailEvent_Format1* Activity

The following table defines the fields on this screen.

Table 24: Details of Fields on Edit Mail Event Screen

Field Name	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* dropdown 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 dropdown list.

Field Name	Description
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: • Sender' Email Address: Email address of the sender • Subject: Subject of mail • File Attachment: Name of the attached file 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.* 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 175).
6. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Source** and then click **Mail**.

The *Manage Mail Source* screen is displayed with the list of existing Mail source activities (see Figure 117).

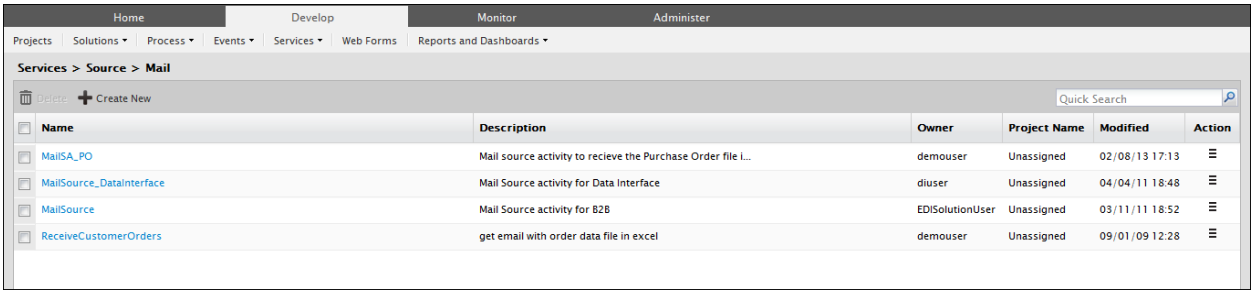

Figure 117: Manage File Source

3. Select the required Mail source activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective File source activity in edit mode (see Figure 118).

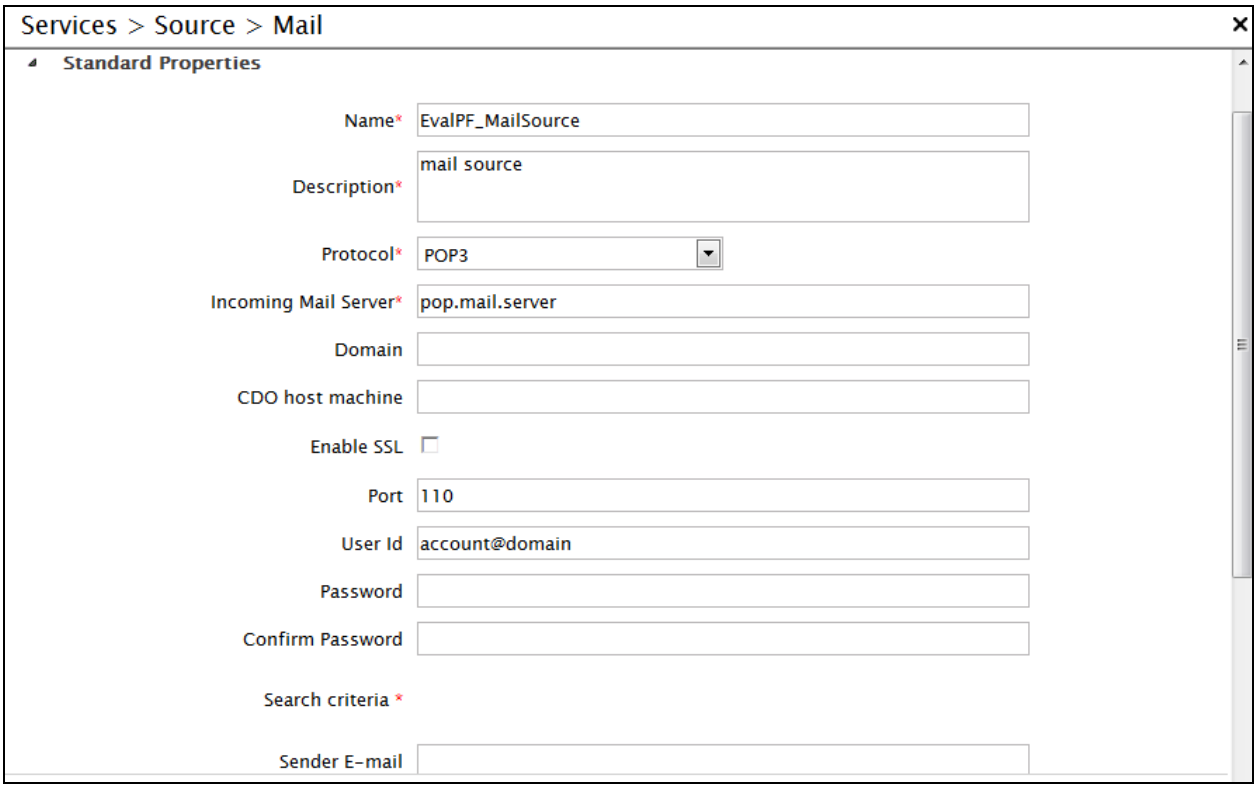

Figure 118: Edit *EvalPF_MailSource* Activity

The following table defines the fields on this screen.

Table 25: Details of Fields on Edit Mail Source Screen

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

Field Name	Description
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* dropdown 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 dropdown 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	Select 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 175).
7. Enter 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.

> You can verify the mail source activity at design time by clicking **Test Connection** button.

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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to > **Services** > **Schema**, and then click **Excel**. The *Manage Excel Schema* screen is displayed with the list of existing Excel Schemas (refer to **Error! Reference source not found.**).
3. Select the required Excel schema activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Excel schema activity in edit mode (see Figure 119).

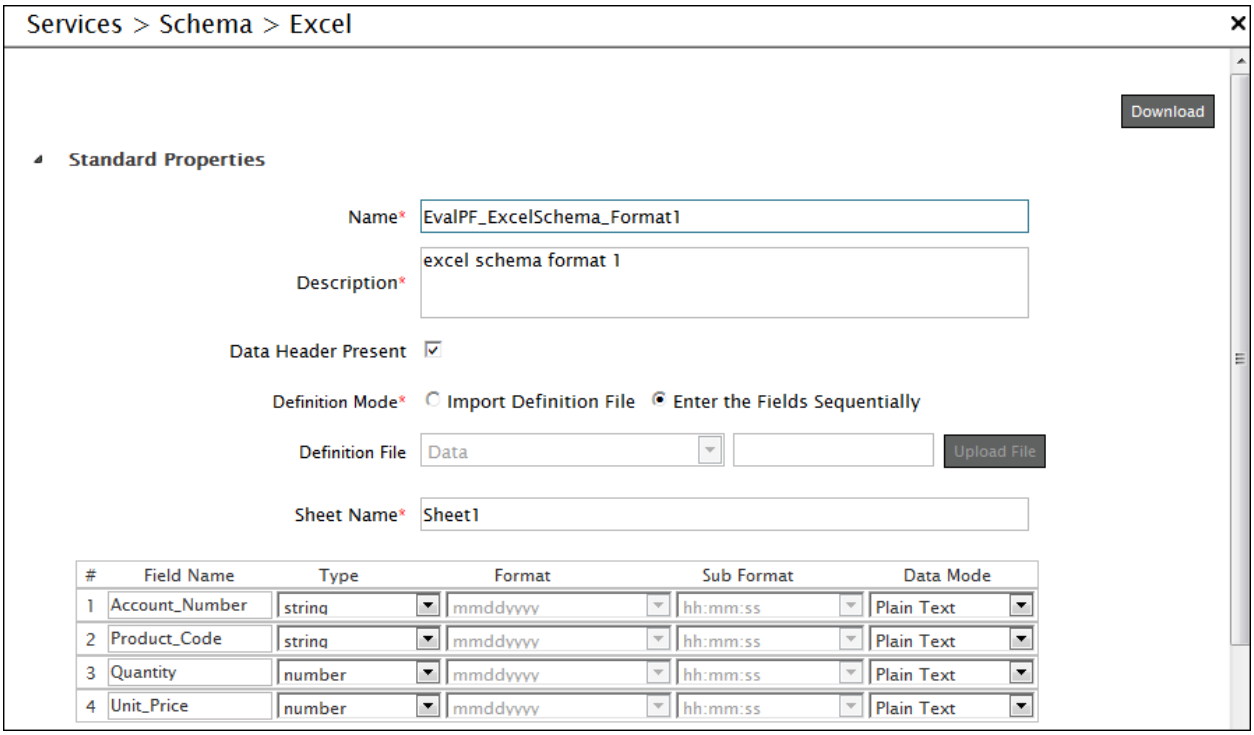

Figure 119: Edit *EvalPF_ExcelSchema_Format1* Activity

The following table defines the fields on this screen.

Table 26: Details of Fields on Edit Excel Schema Screen

Field Name	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: Use Definition File Enter the Field 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 following 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* dropdown list.
SubFormat	If data type is *Date*, select time format from *SubFormat* dropdown 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 175).

7. Enter comments in the **Add Comments** field.
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 178).
2. Select the type of schema to test, from the **Type** dropdown 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation and then click Data Mapping.

   The *Manage Data Mapping* screen is displayed with the list of existing activities (see Figure 26).
3. Select the required Data Mapping activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Data Mapping activity in edit mode (see Figure 120).

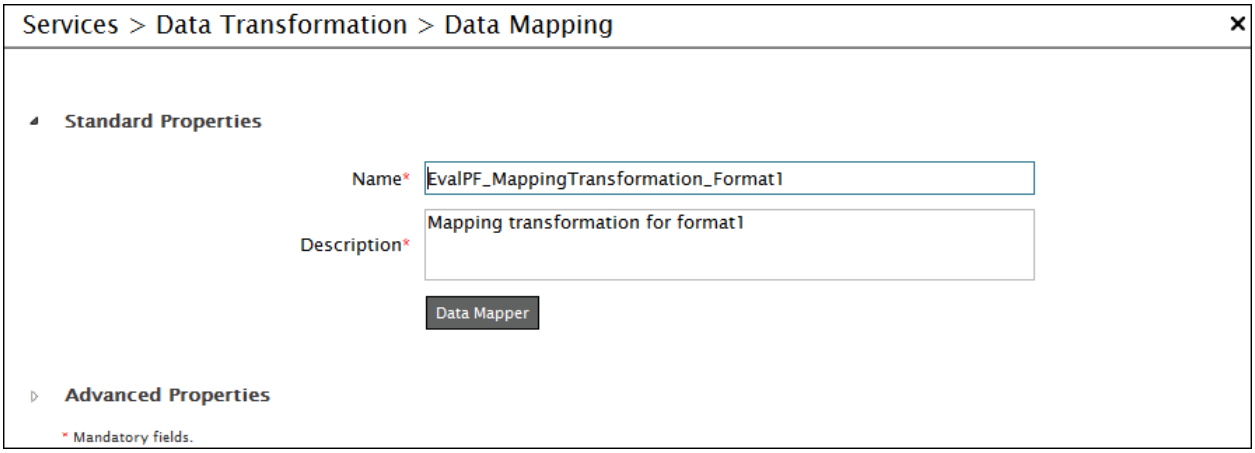

Figure 120: Edit *EvalPF_MappingTransformation_Format1* Activity

5. Click the **Data Mapper** button**.** The Data Mapper applet is displayed showing mapping between data fields of source and target schema (see Figure 121).

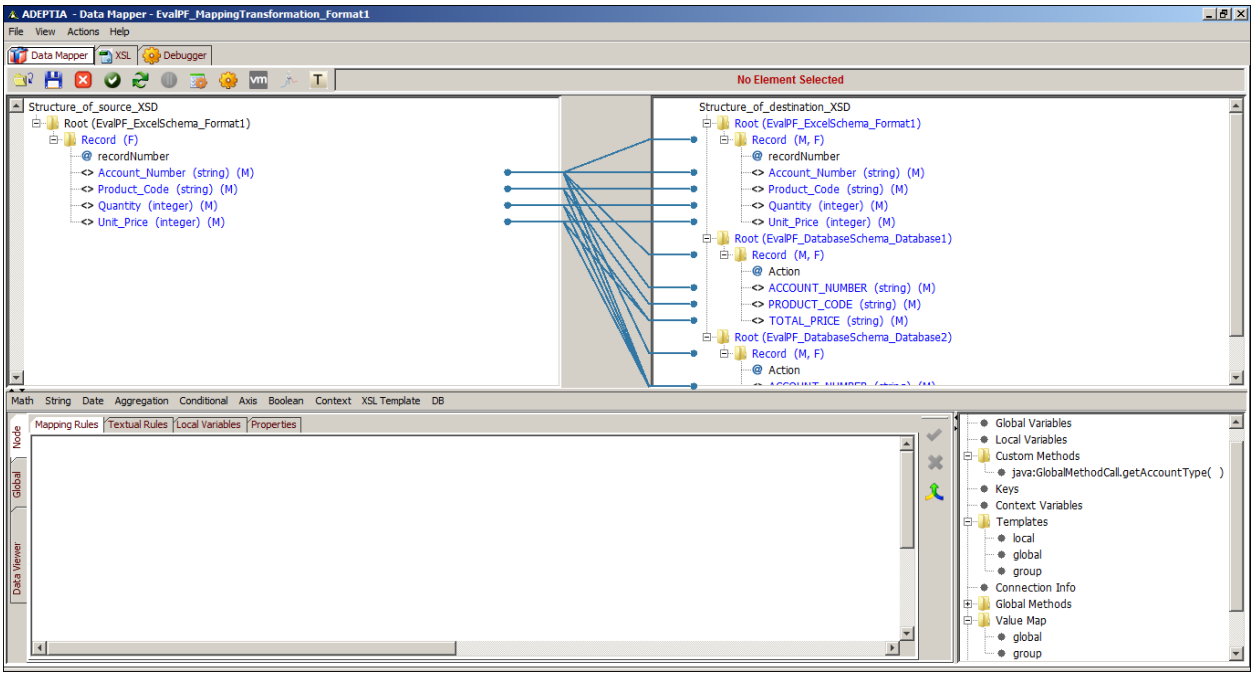

Figure 121: Edit *EvalPF_MappingTransformation_Format1* Activity in Data Mapper Applet

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 122).

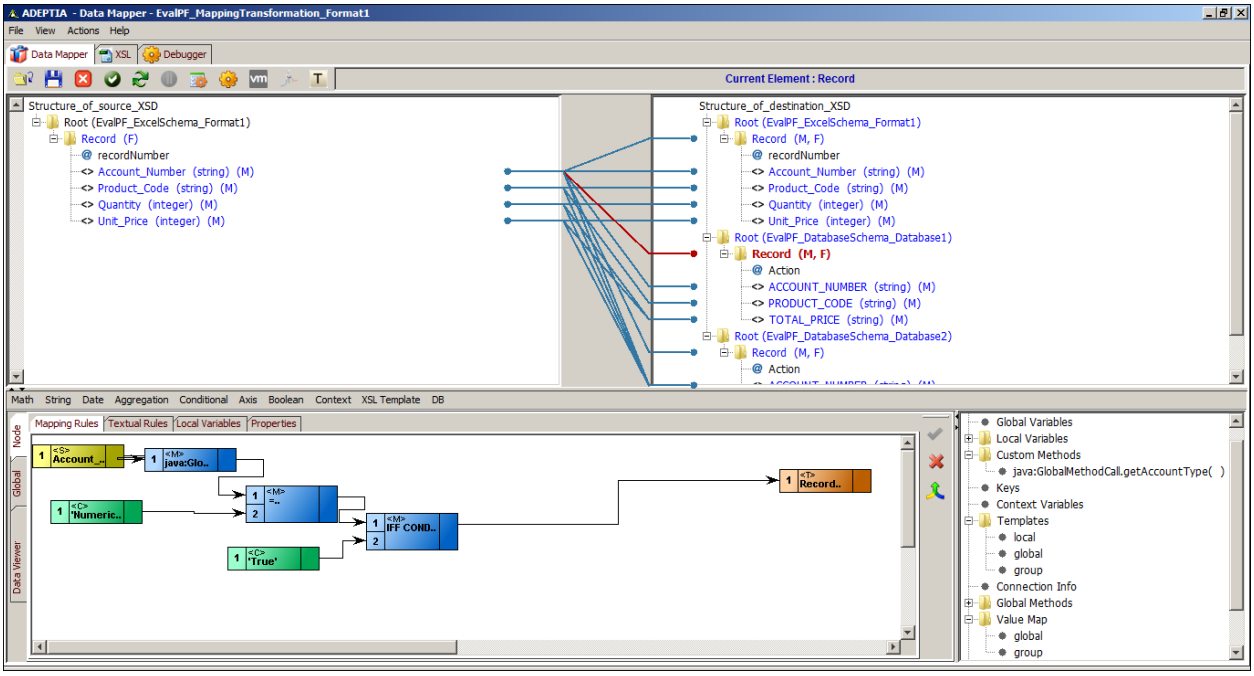

Figure 122: View Mapping Functions Used in *EvalPF_MappingTransformation_Format1* Activity

7. To analyze the impact of mapping function on output records, right-click any of function node (**IFF Condition**) and select **Information** (see Figure 123).

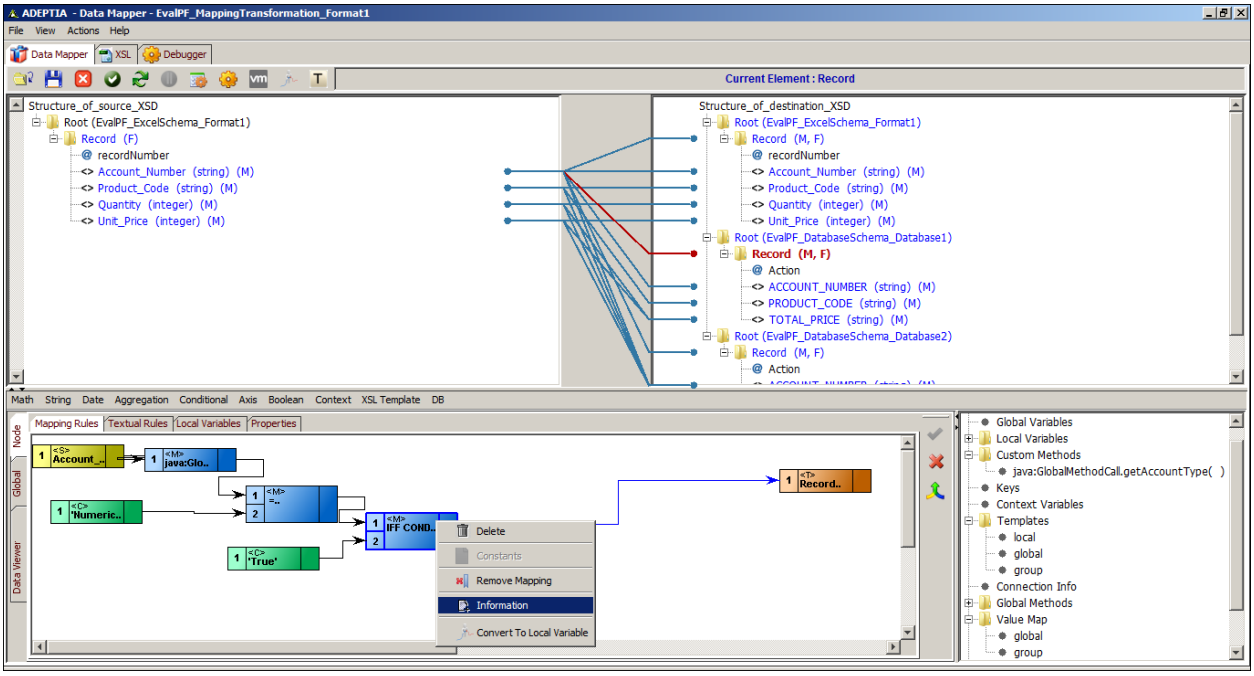

Figure 123: Select Mapping Function Information

8. A dialog box is displayed that shows information about that mapping function (see Figure 124).

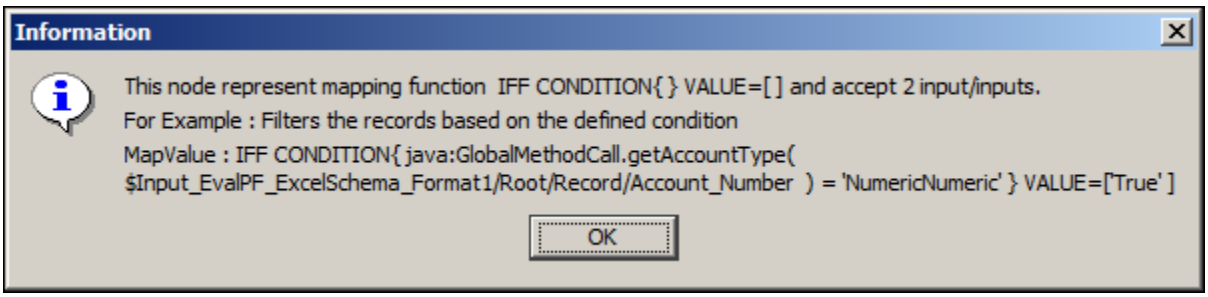

Figure 124: View Mapping Function Information

9. Click **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 181).
13. Enter the comments in the Specify comments for mapping object (object name) field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
8. Go to **Services > Connector**. All items in the **Connector** category are displayed.
9. Click **Database Driver**. The *Manage Database Driver* screen is displayed with the list of existing database drivers (refer to Figure 6).
10. Select the required Database Driver activity and right-click it to view the **More Actions** menu.

11. Click **Edit** link. This opens the respective Database Driver activity in the edit mode (see Figure 125).

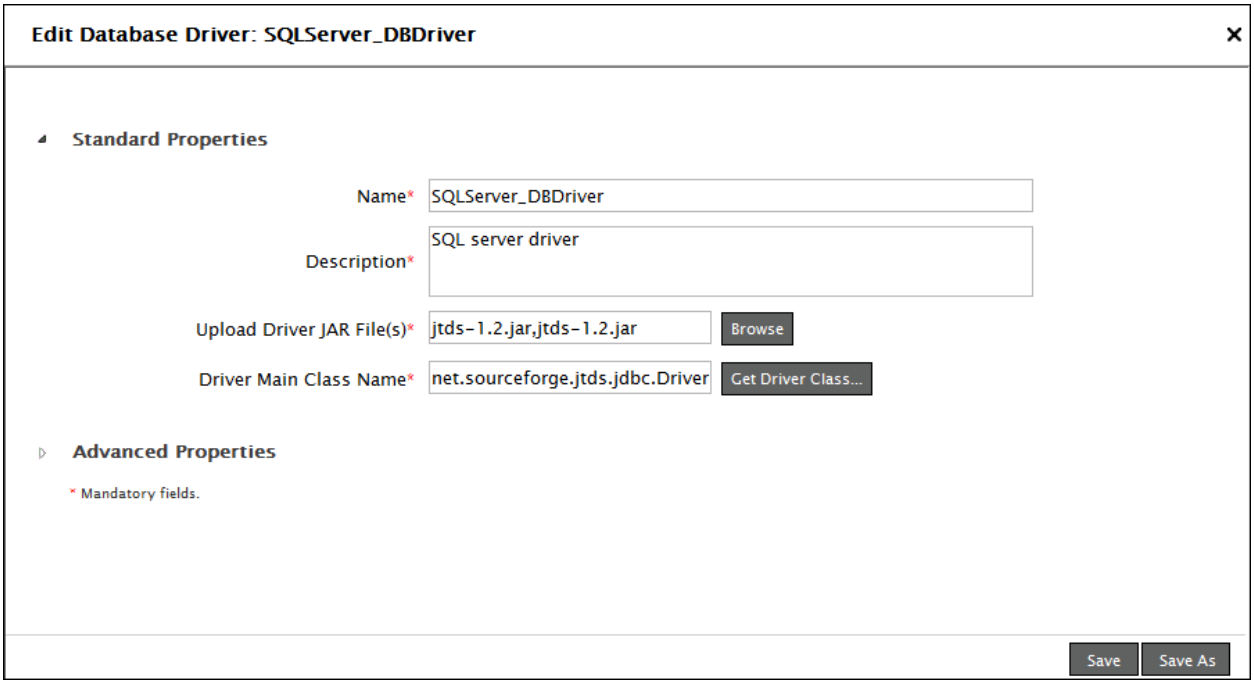

Figure 125: Edit *SQLServerDBDriver*

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

Table 27: Details of Fields on Edit Database Driver Screen

Field Name	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: Oracle: Ojdbc5.jar IBM DB2 (Ver 7.1): db2java.zip (7.1 version) IBM DB2 (Ver 8.1): db2jcc.jar MS SQL: msbase.jar, mssqlserver.jar and msutil.jar JTDS- SQL Server: Jtds.jar HSQL DB: hsqldb-1.7.2.jar

Field Name	Description
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 dropdown 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 sun.jdbc.odbc.JdbcOdbcDriver AS400: com.ibm.as400.access.AS400JDBCDriver MYSQL : com.mysql.jdbc.Driver

12. Make the necessary changes.
13. 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 175).
14. Enter comments in the **Add Comments** field.
15. 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **Database Info**. The *Manage Database Info* screen is displayed with the list of existing Database Info activities (refer to Figure 8).
4. Select the required Database Info activity and right-click the same to view the **More Actions** menu.

5. Click **Edit** to open the respective Database Info activity in edit mode (see Figure 126).

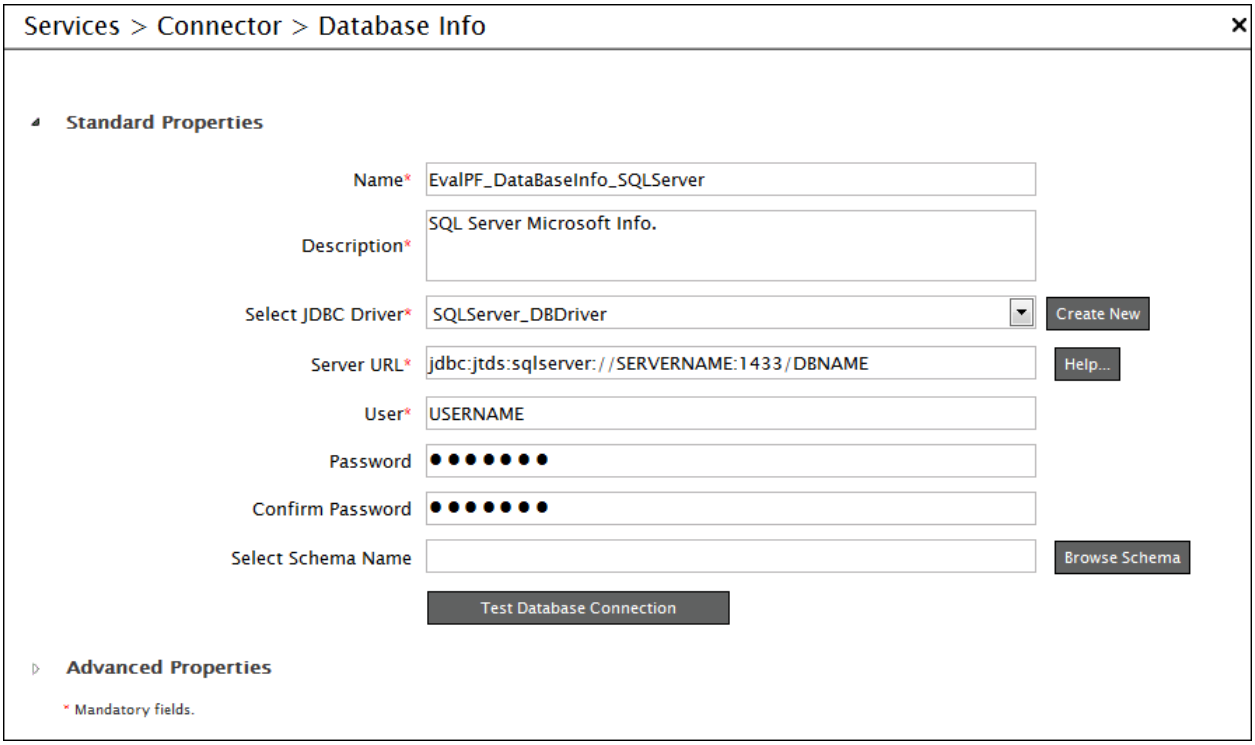

Figure 126: Edit *EvalPF_Databaseinfo_SQLServer*

The following tables define the fields on this screen.

Table 28: Details of Fields on Edit Database Info Screen

Field Name	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.

Field Name	Description
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) / IBM DB2 (Ver 8.1): jdbc:db2://databaseserver:6789/TOOLSDB MS SQL: jdbc:db2://databaseserver:50000/TOOLSDB SQL JTDS: jdbc:microsoft:sqlserver://databaseserver:1433;DatabaseName=master 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* 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.

6. Make the necessary changes.
7. 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 175).
8. Enter comments in the **Add Comments** field.
9. 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Schema** and then click **Database**. The *Manage Database Schema* screen is displayed (refer to Figure 10).
3. Select the required database activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Database schema activity in edit mode (see Figure 127).

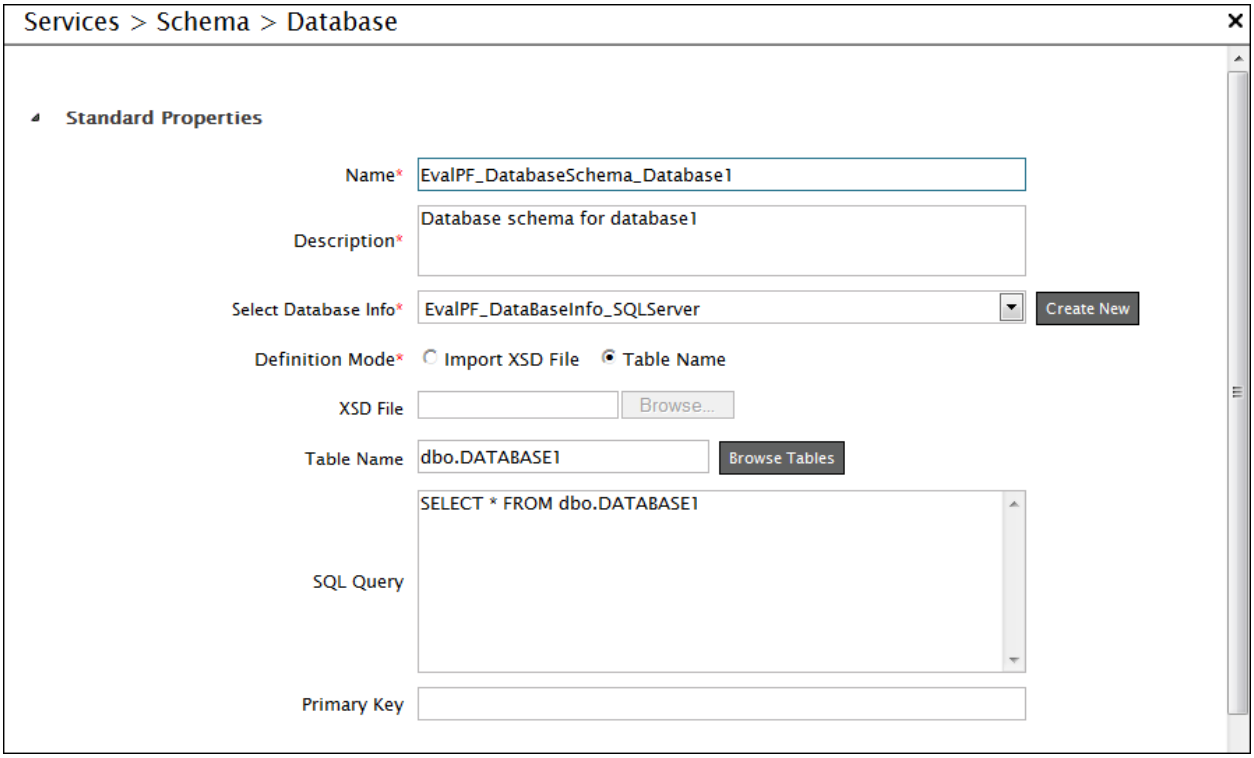

Figure 127: Edit *EvalPF_DatabaseSchema_Database1* Activity

The following table defines the fields on this screen.

Table 29: Details of Fields on Edit Database Schema Screen

Field Name	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: • Use XSD File • Table Name 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. 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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **Database**.

   The *Manage Mail Database* screen is displayed with a list of existing Database Target activities (see Figure 128).

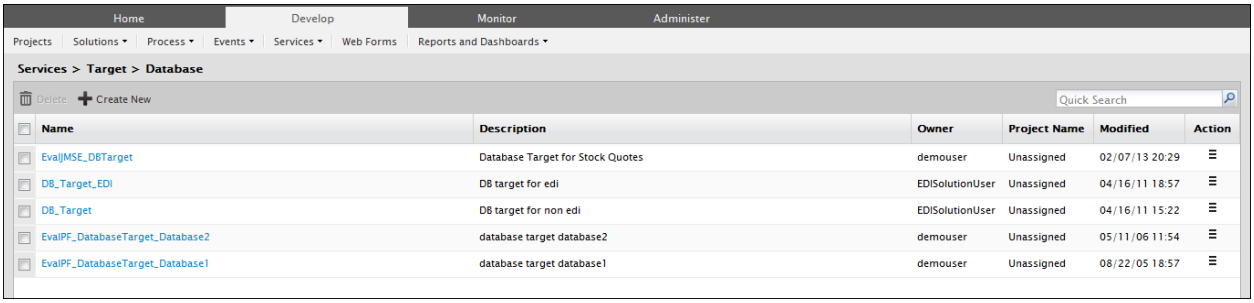

Figure 128: Manage Database Target

3. Select the required Database target activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Mail target activity in edit mode (see Figure 129).

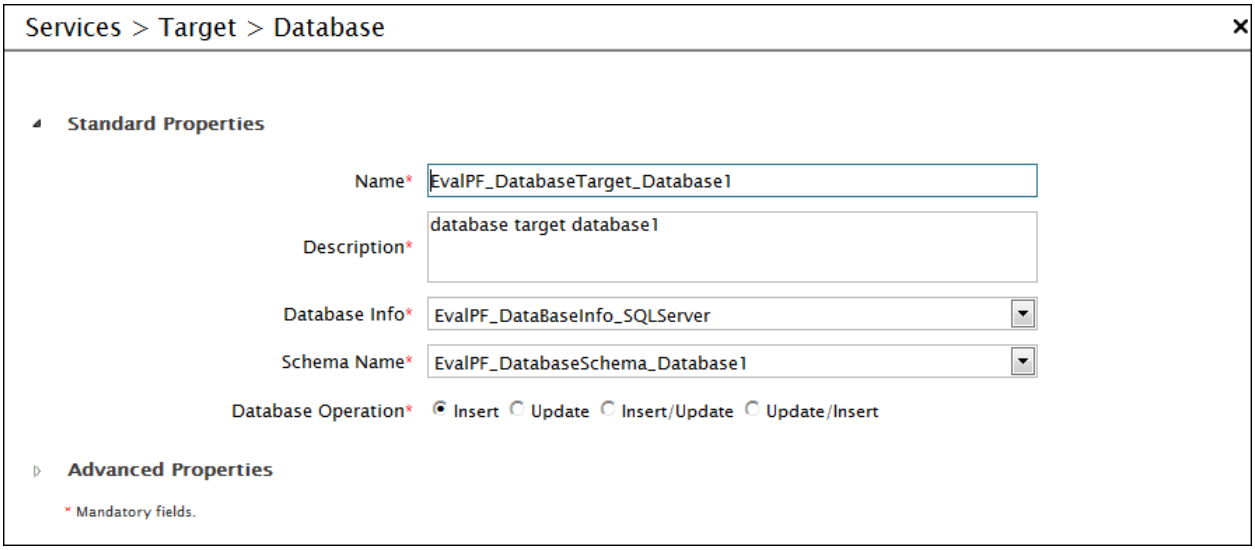

Figure 129: Edit *EvalPF_DatabaseTarget_Database1* Activity

The following table defines the fields on this screen.

Table 30: Details of Fields on Edit Database Target Screen

Field Name	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.

Field Name	Description
Database Operation	Database operation specifies how data records are inserted into database tables. Select one of the following database operations: • Insert • Update • Insert/Update • Update/Insert 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. **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. 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. 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. **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. 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 175).
7. Enter comments in the **Add Comments** field.

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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **File**.

   The *Manage File Target* screen is displayed with the list of existing File Target activities (refer to Figure 22).
3. Select the required File target activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective File target activity in edit mode (see Figure 130).

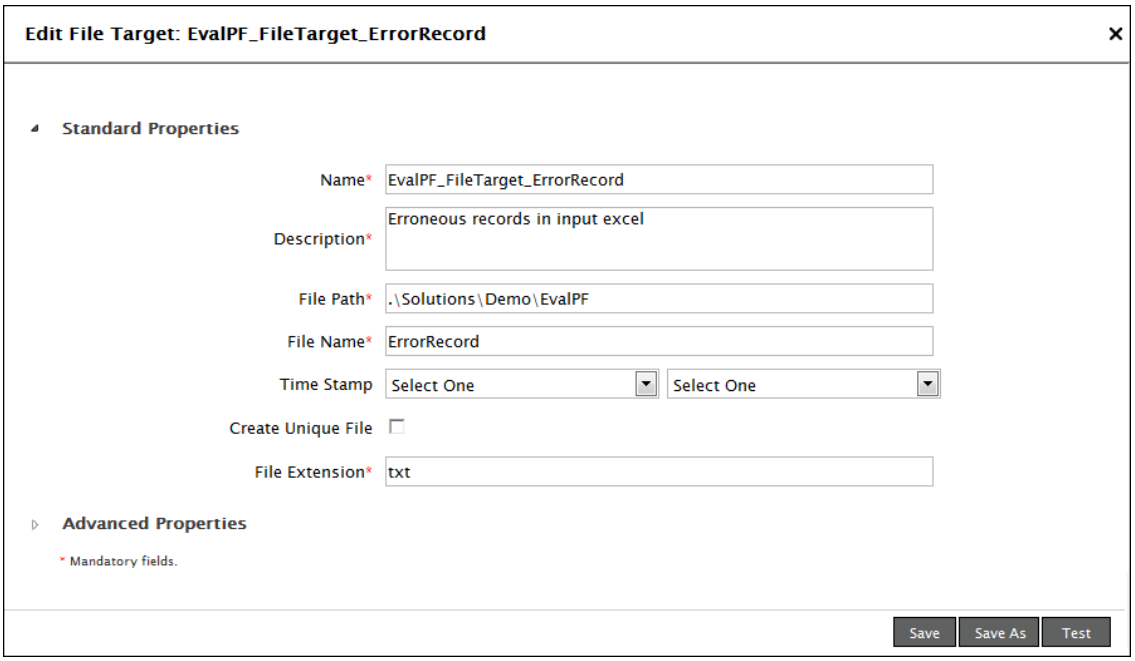

Figure 130: Edit *EvalPF_FileTarget_ErrorRecord* Activity

The following table defines the fields on this screen.

Table 31: Details of Fields on Edit File Target Screen

Field Name	Description
Name	Name of the File Target
Description	Description of the File Target
File Path	Path of the target file. For example: **../../Solutions/Demo/EvalPF/** In the activities *EvalPF_FileTarget_InsertError* and *EvalPF_FileTarget_UpdateError*, the path will be ../../Solutions/Demo/EvalPF/ and ../../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 175).
7. Enter 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.

Repeat the same steps to edit EvalPF_FileTarget_InsertError and EvalPF_FileTarget_UpdateError activities.

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

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 32: 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation and then click Data Mapping.

   The *Manage Data Mapping* screen is displayed with the list of existing activities (refer to Figure 26).
3. Click the **Create New** link. The *Create Data Mapping* screen is displayed (refer to Figure 27).
4. Enter the name and description of the new mapping activity in the textboxes **Name** and **Description** respectively.
5. Click the **Data Mapper** button. This displays the Data Mapper applet (refer Figure 28).
6. Click **Open Source Schema** button to load the source schema. This displays the **Select Schema** screen (see Figure 29).
7. Select **Source Schema (EvalPF_ExcelSchema_Format1)** and select **Target Schema** (EvalPF_ExcelSchema_Format1, EvalPF_DatabaseSchema_Database1, EvalPF_DatabaseSchema_Database2).
8. Click  button to load the schemas. This loads the selected schemas in the Source and Target Panel of the Data Mapper applet.

9. Expand the tree structure for Source Schema and the Target Schema (see Figure 131).

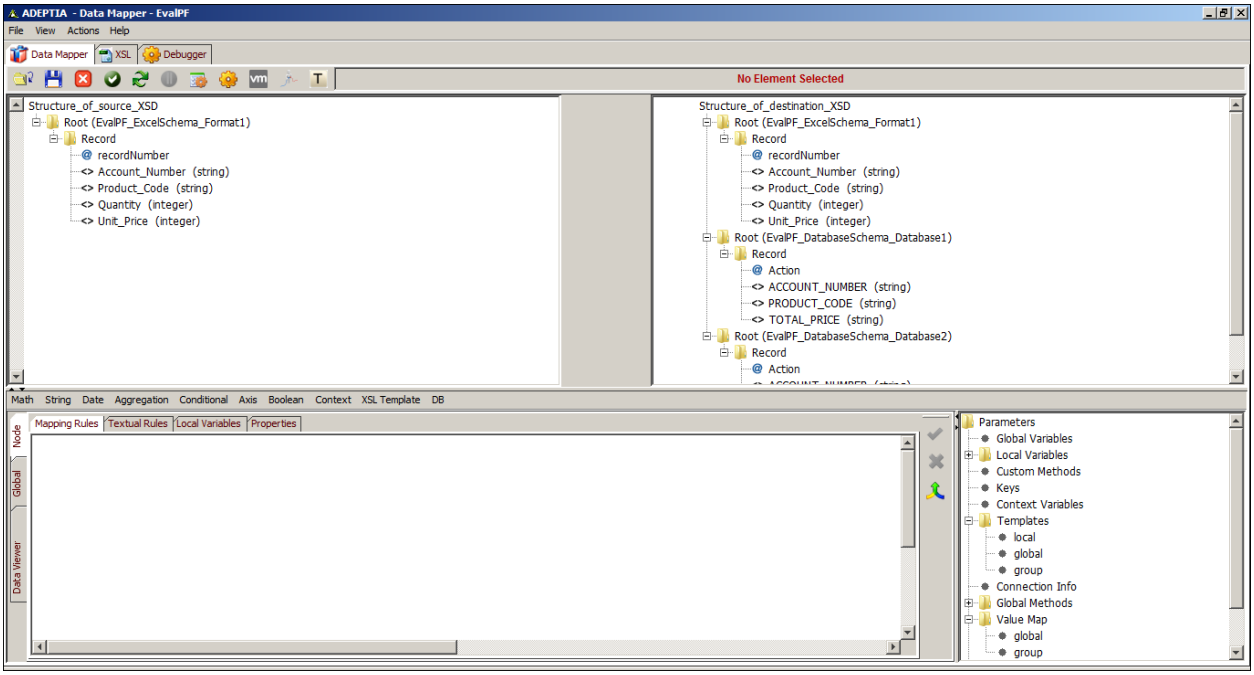

Figure 131: Expand Source and Target Schemas

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

17. Click the **For Each** field and then double-click the **Record** element of the source schema.

18. Click **Save Properties** to save the *For Each* property (see Figure 132).

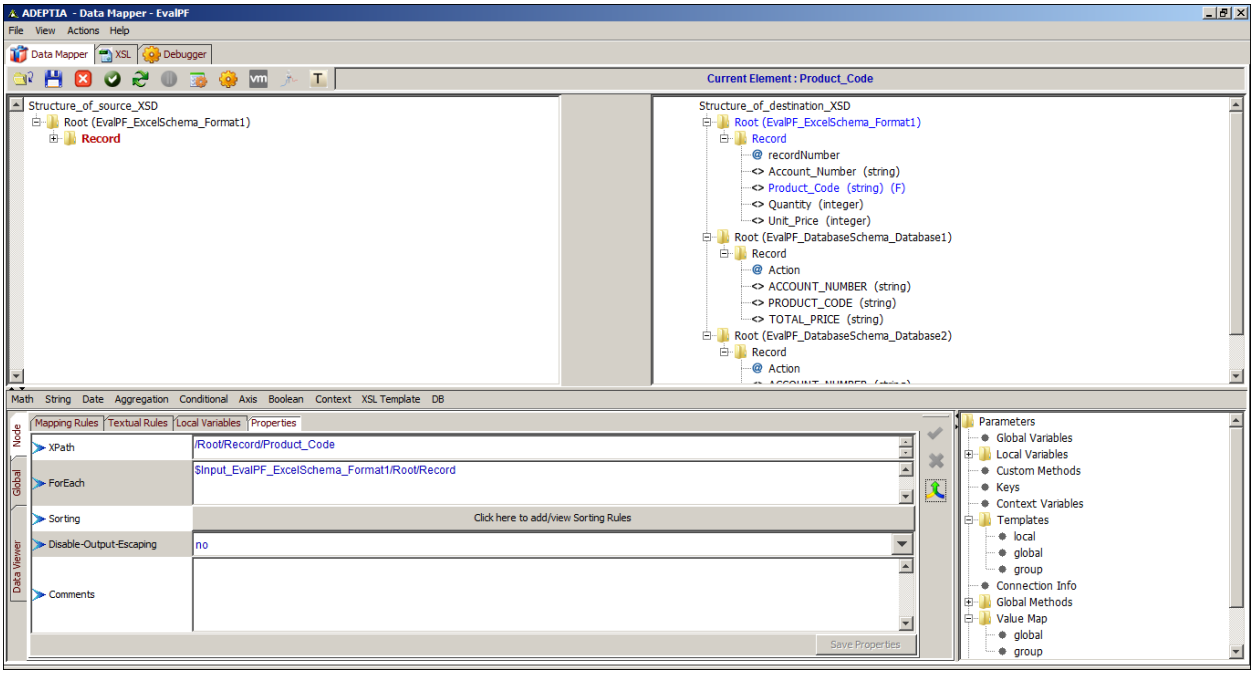

Figure 132: Apply For Each Property

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

20. 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 133).

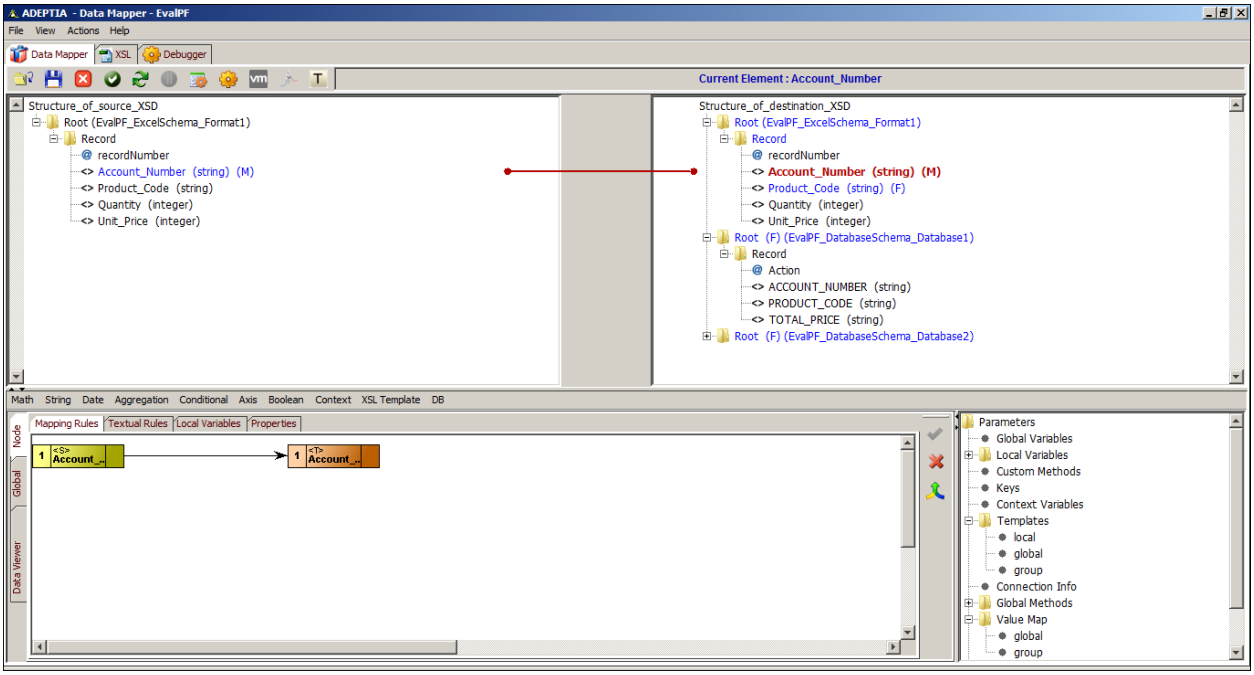

Figure 133: Map Account Numbers

21. 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.

22. 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.

23. 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 134).

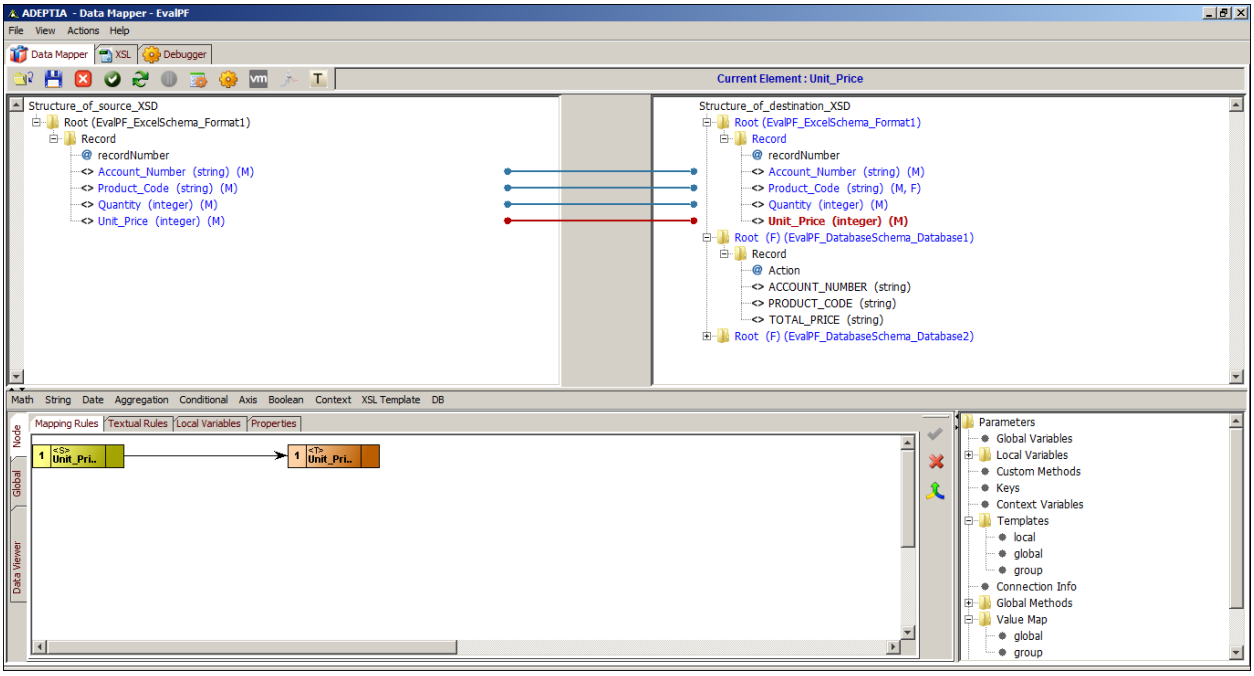

Figure 134: Map Source and Target Elements of First Target Schema

24. 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 135).

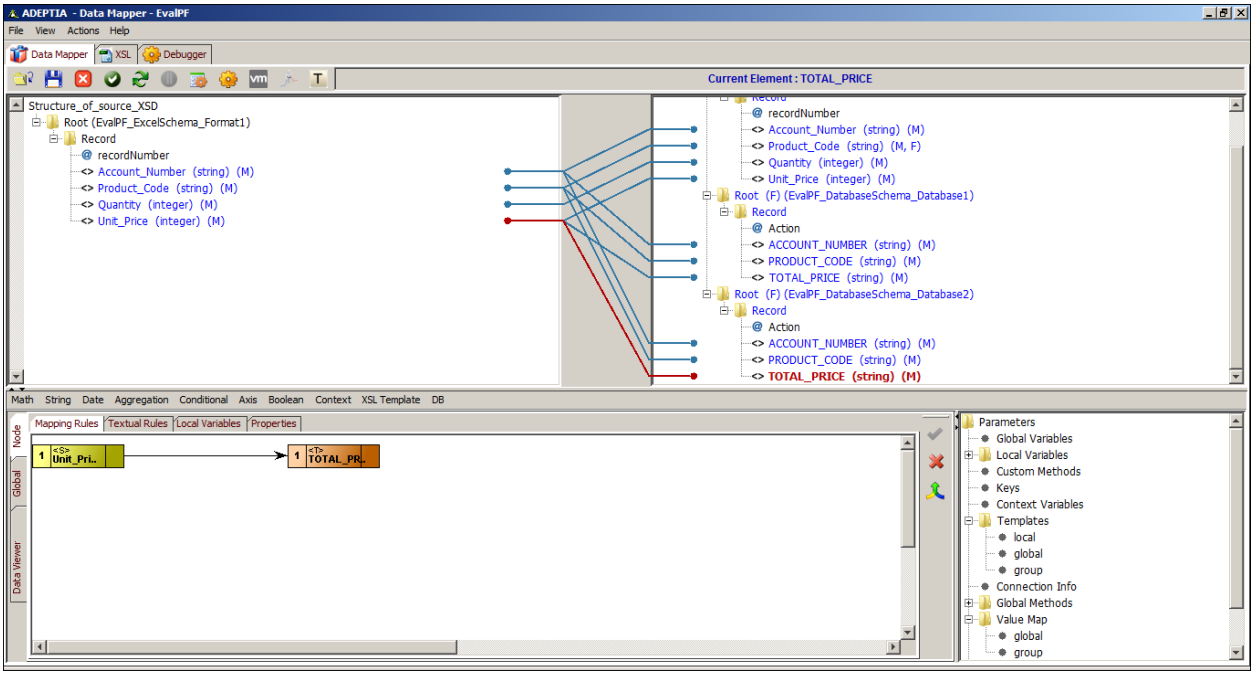

Figure 135: Mapping Source and Target Elements of All Target Schemas

25. To filter the records based on *Account_Number* field of the source schema, Custom Method is used.
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 136).

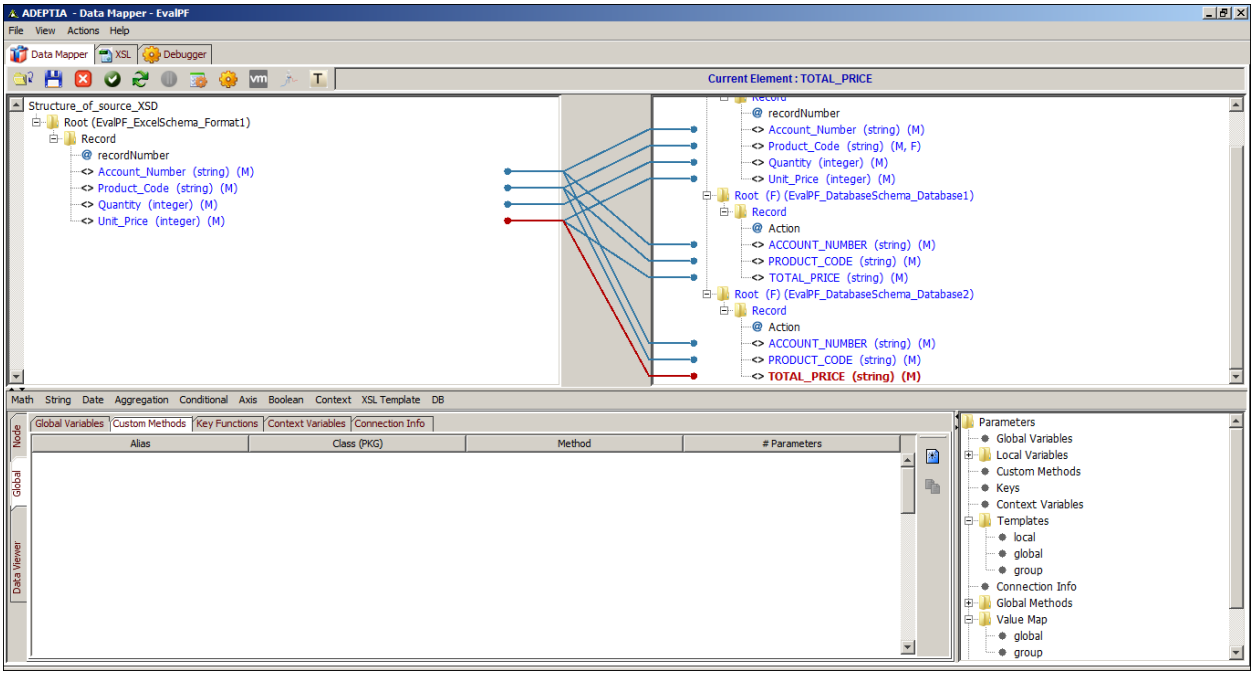

Figure 136: Custom Methods Pane

28. Click the **Add Method** ( ) button to add a Custom Method. A row is inserted with following columns (see Figure 137):

- Alias
- Class (PKG)
- Method

- # Parameters

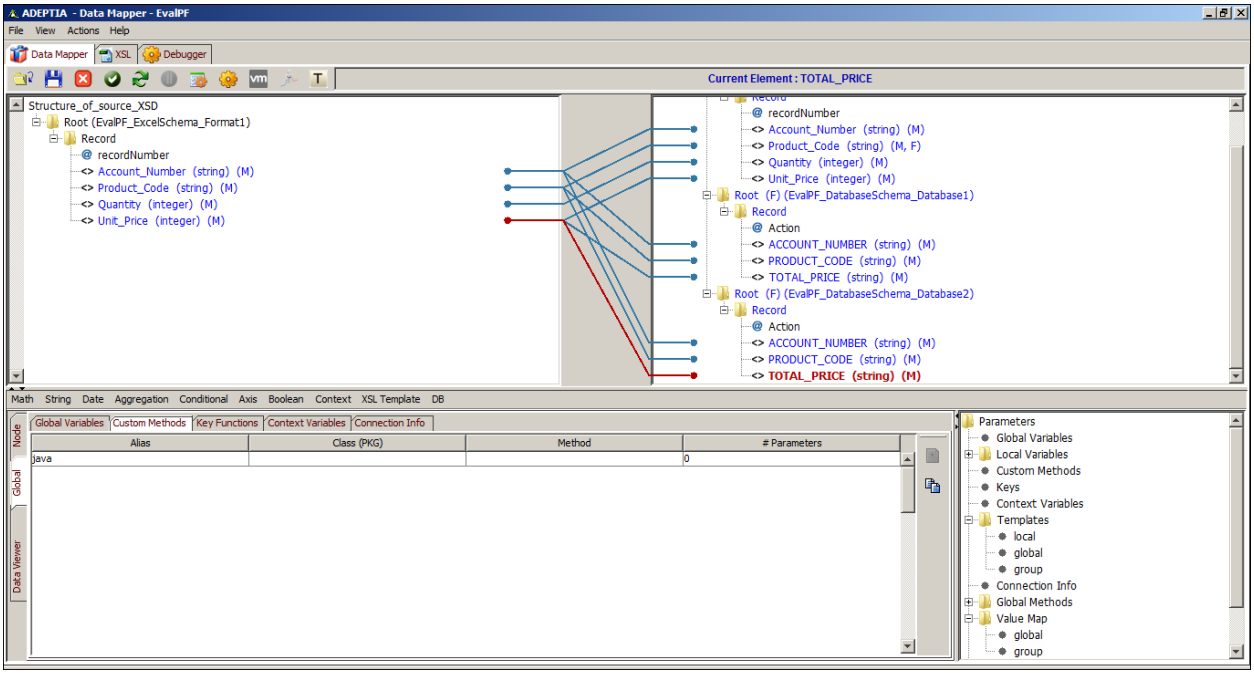

Figure 137: Create Custom Method

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 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.

33. 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.

34. Press the [**Tab**] or [**Enter**] key to return to the *Alias* field. This will save the added Custom Method (see Figure 138).

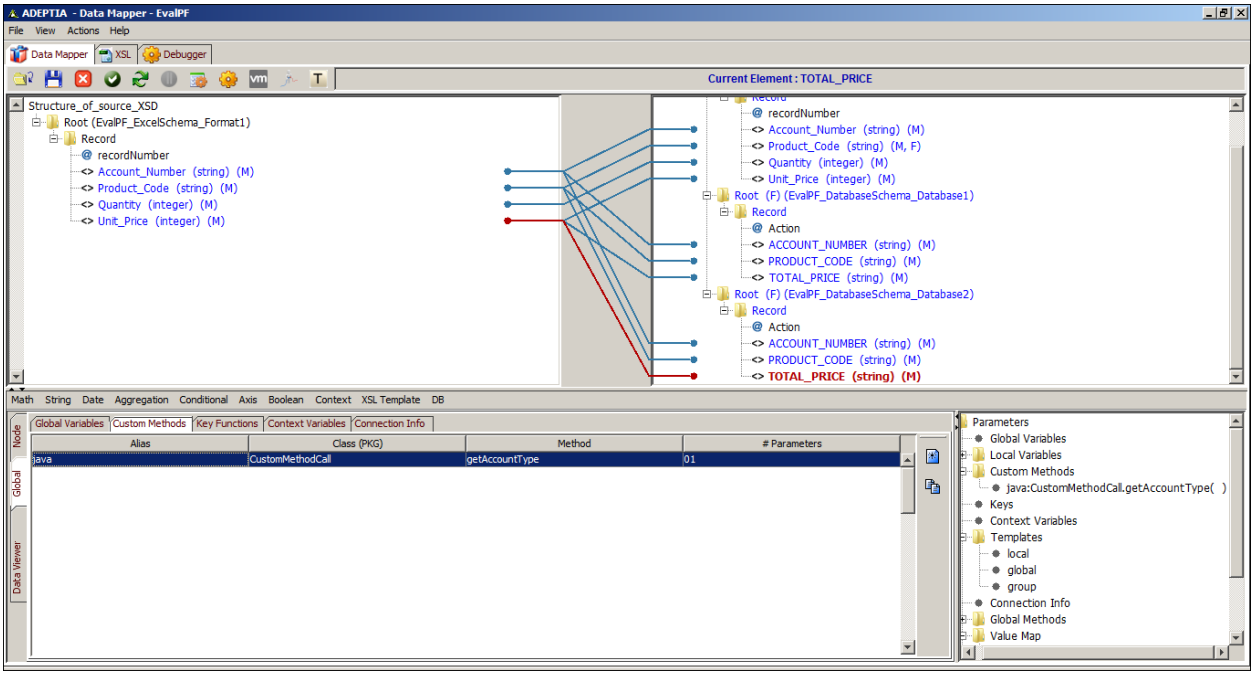

Figure 138: Save Custom Method

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

35. Click the **Node** tab to map above declared Custom Method to the *Account_Number* node. The Mapping Graph Area is displayed.
36. Select the node **Record** element of first target schema. The *Record* node is shown in the Mapping Graph Area.
37. Double-click the **Account_Number** node in the Source Panel. The *Account_Number* node is shown 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 139).

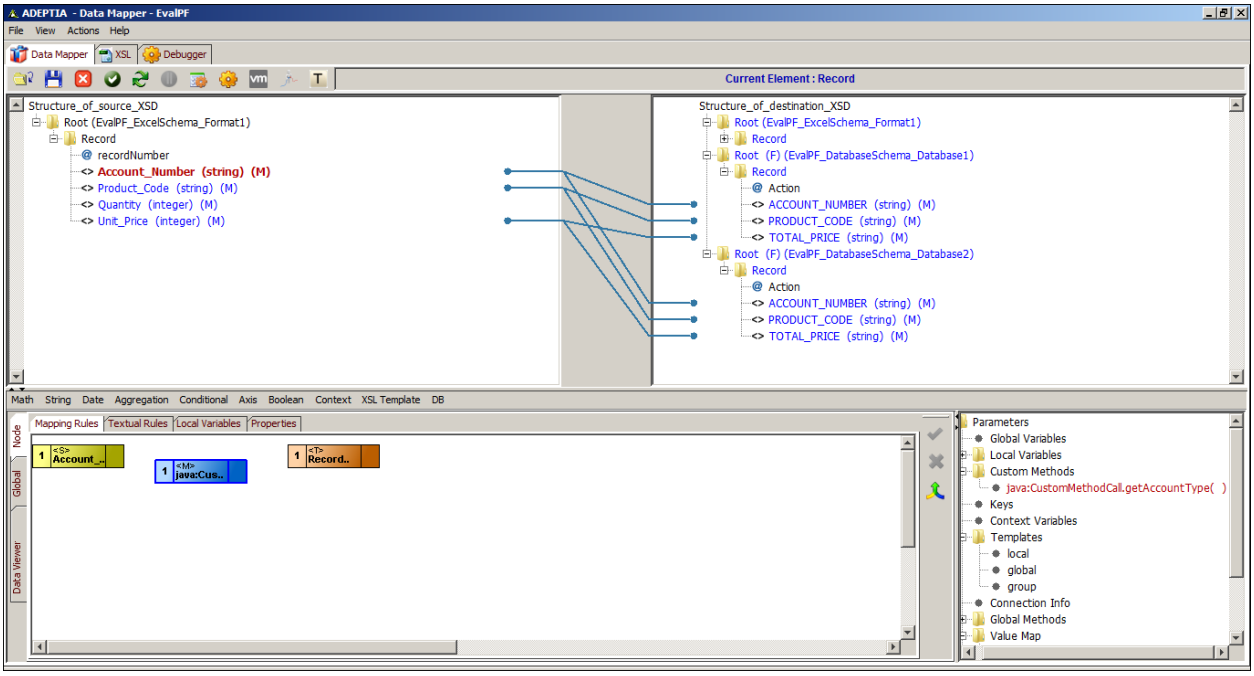

Figure 139: Select Custom Method

39. 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.

40. Double-click the **Constant** node. The Input dialog box is displayed (refer to Figure 49).

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

42. Check the *Add Quotes* checkbox and click the **OK** button. The entered value is shown in the *Constant* node (see Figure 140).

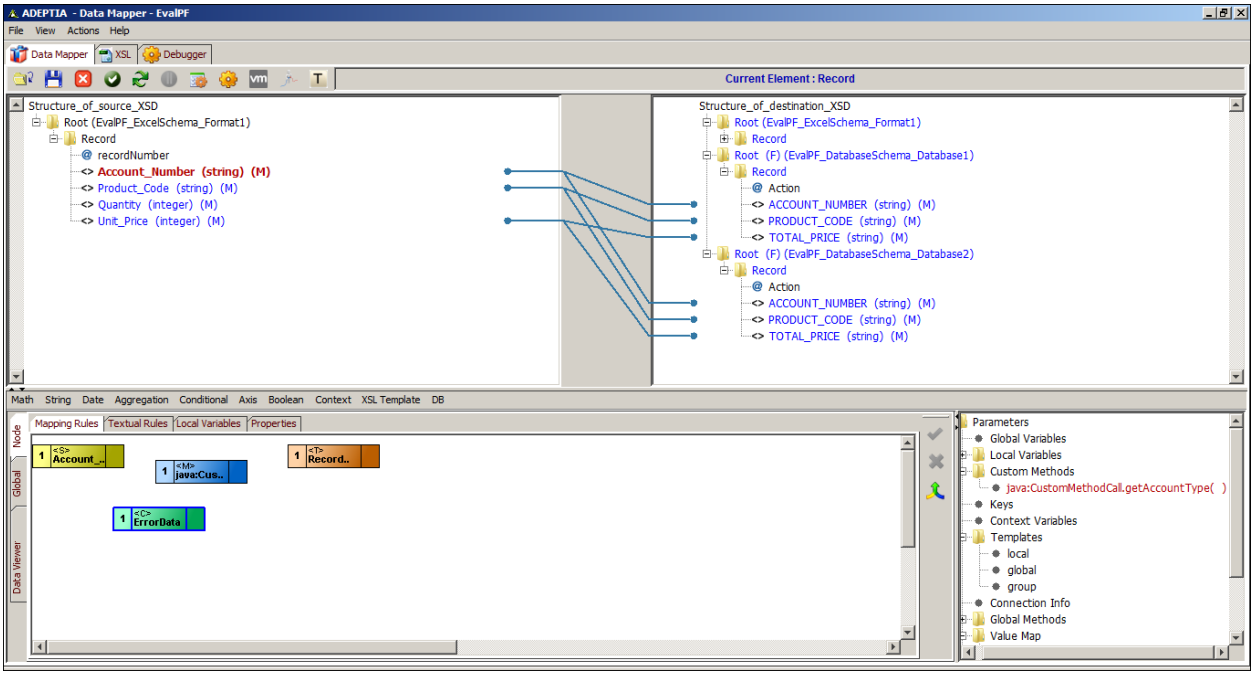

Figure 140: Constant Value Entered

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

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

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

46. Create a link from the output of the *Constant* function node to the second input of the *Equal* function node (see Figure 141).

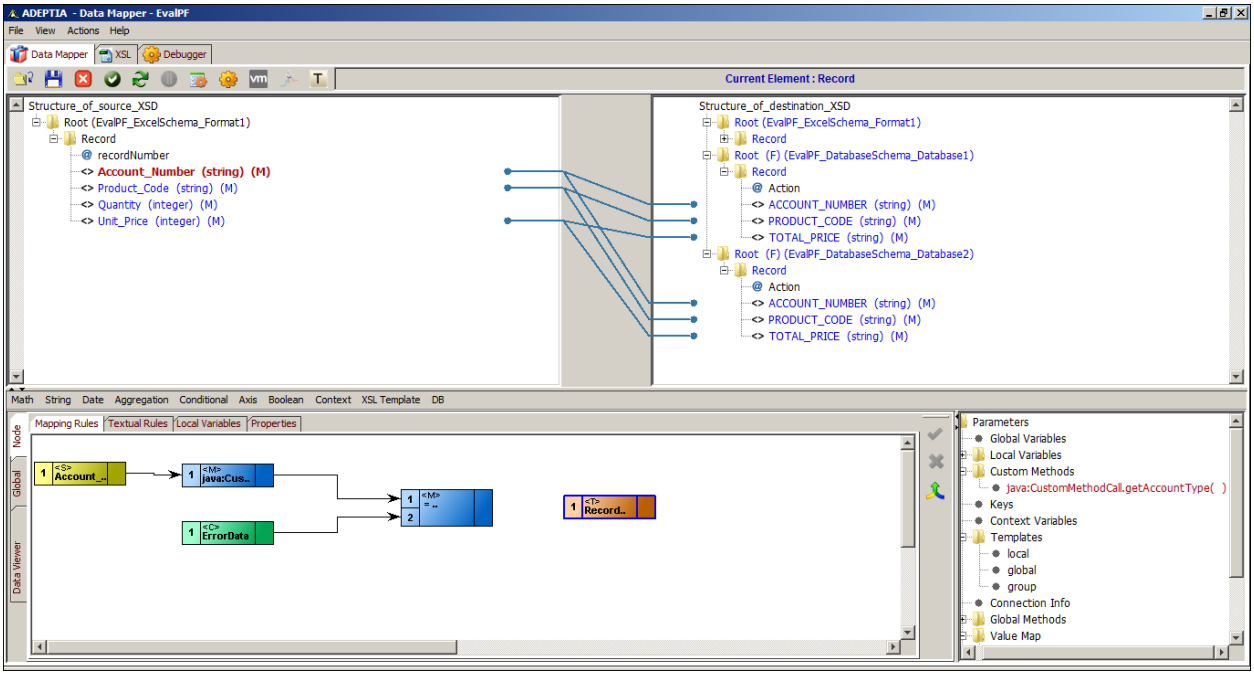

Figure 141: Create Links

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

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

48. 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 142).

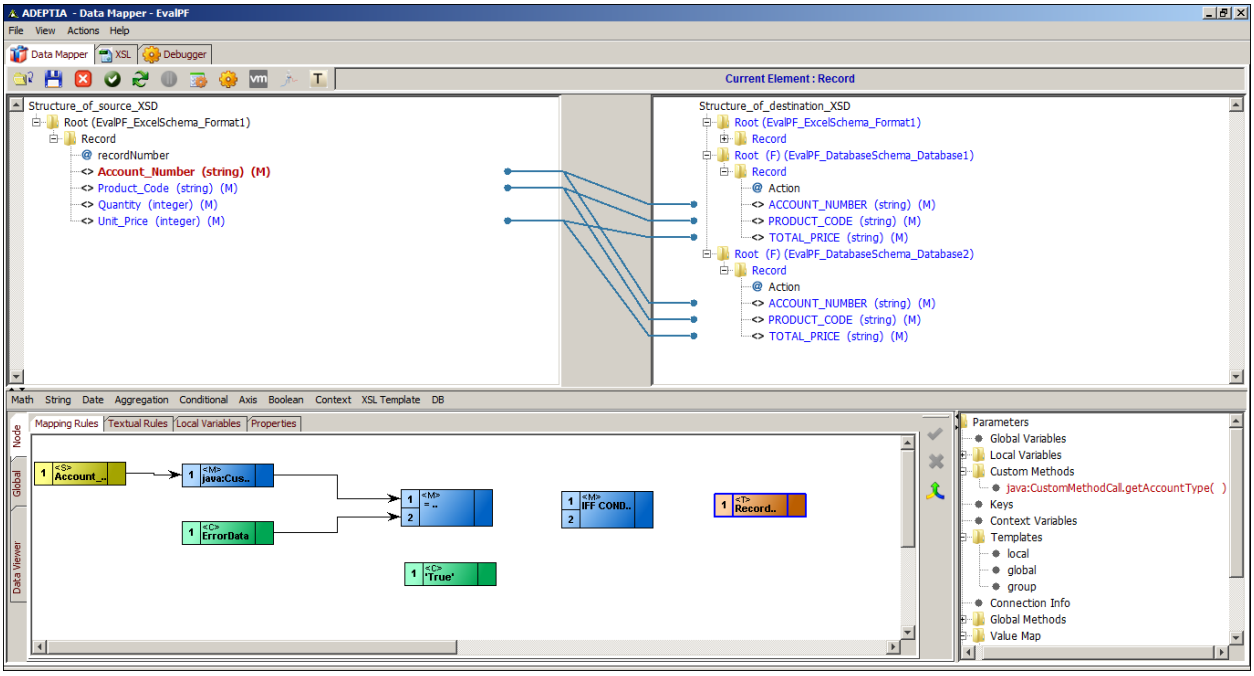

Figure 142: Select IF Condition

49. Create a link from the output of the *Equal* function node to the first input of the *IFF Condition* node.
50. Create a link from the output of the *Constant* node (‘*True’*) to the second input of the *IFF Condition* node.
51. Create a link from the output of the *IFF Condition* node to the input of the *Record* node.

52. Click the **Apply Mapping** ( ) button to apply the mapping. Lines indicating mapping between the source and target nodes are displayed (see Figure 143).

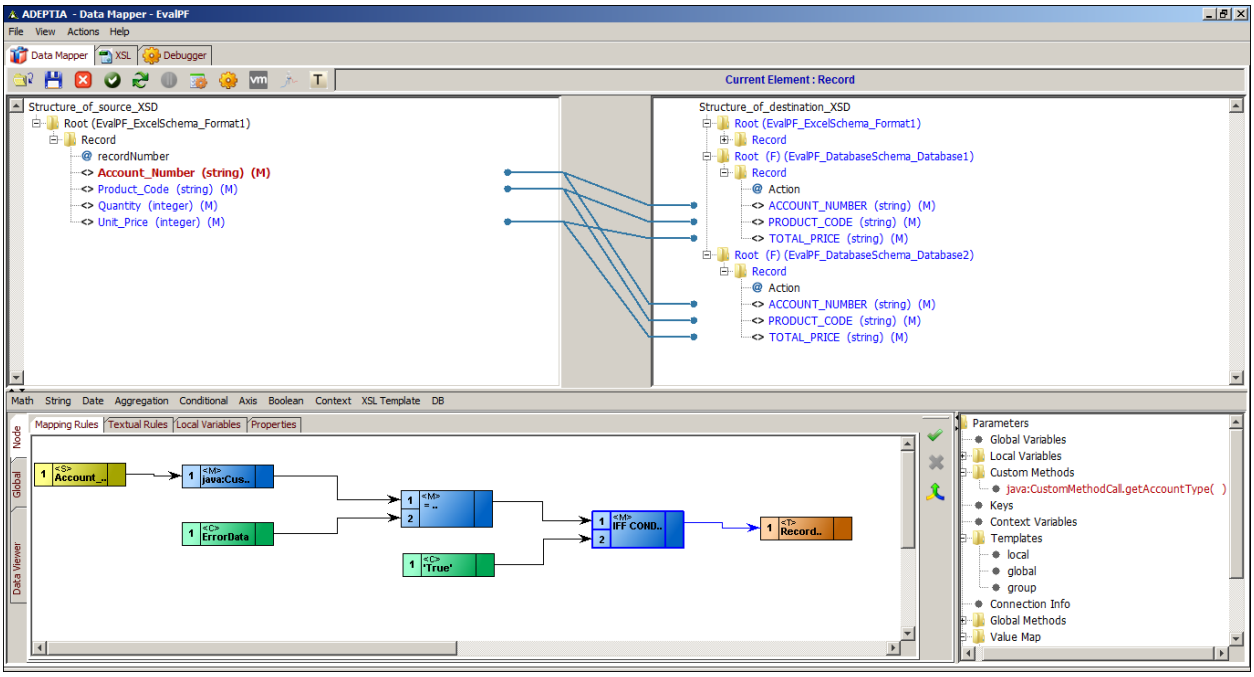

Figure 143: Apply Mapping

53. 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 c *ErrorData.*

54. 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.

55. Right-click the **Root** of the second target schema and select **Assign Stream** option. The **Assign Streams** dialog box is displayed (see Figure 144).

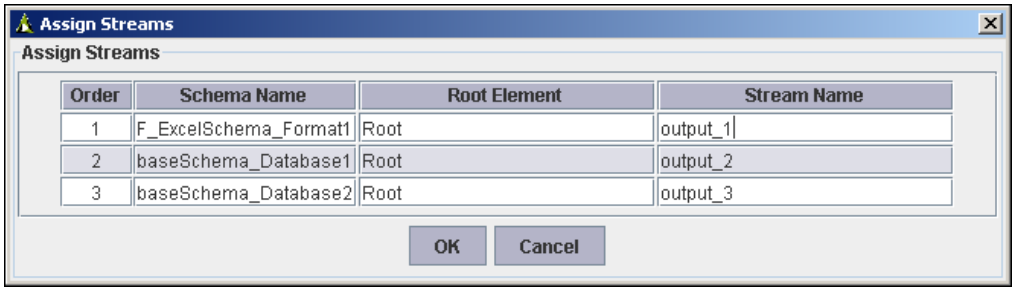

Figure 144: Assign Streams

56. Enter any stream name (*default*) for the Root having Occurrence Order *1*.
57. Enter any stream name (*database1*) for the Root having Occurrence Order *2*.
58. Enter any stream name (*database2*) for the Root having Occurrence Order *3* (see Figure 145).

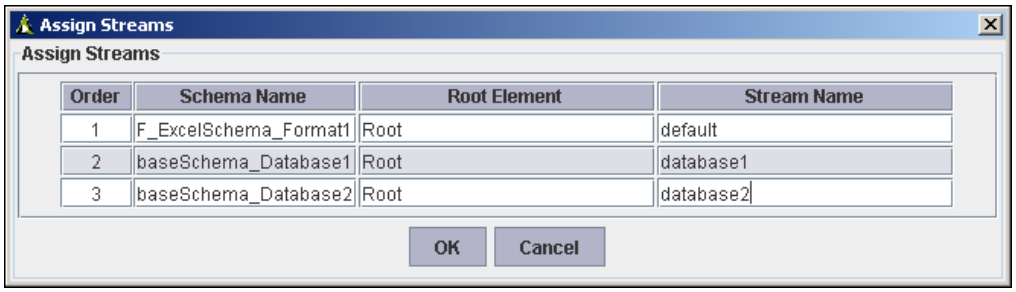

Figure 145: Enter Stream Name

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.

59. Click on **OK** to close the Assign Stream dialog box.
60. 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.
61. 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 181).
62. Enter the comments in the Specify comments for mapping object (object name) field.

The comment should be at least 1 character in length.

63. Click **OK** to save the comments. This displays a dialog box confirming that the mapping has been saved successfully.
64. 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (refer to Figure 55).
3. Click the **Create New** link. The *Create Process Flow* screen is displayed (refer to Figure 56).
4. Enter the name and description of the Process Flow.
5. Click the **Process Designer** button to open Process Designer. The *Process Designer* screen is displayed (refer to Figure 57).

For more information on how to create process flows and on *Process Designer* screen, refer to the Creating Process Flow section.

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

8. Select **EvalPF_MailSource** and drag it to the Graph Canvas Area (see Figure 146).

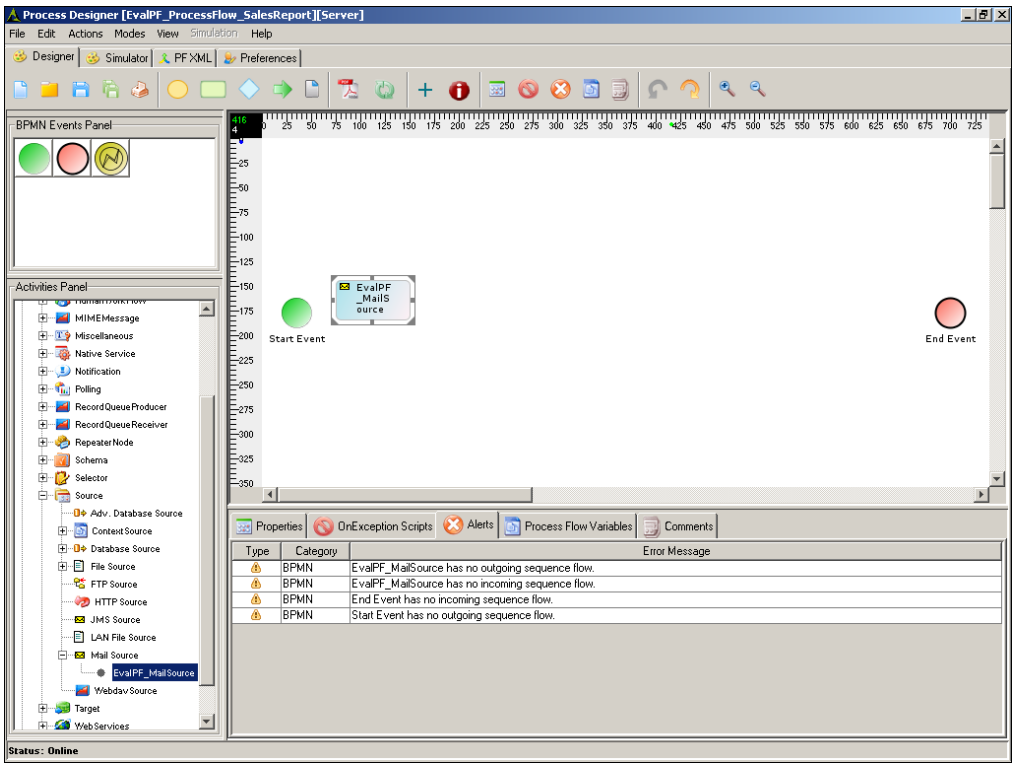

Figure 146: Drag File Source Activity to Graph Canvas Area

9. 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.

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

11. To select a BPMN Gateway, click the **Gateway** ( ) icon in the Palette and drag it to the Graph Canvas Area (see Figure 147). In this Process Flow, **Gateway** is used to check the subject of the incoming mail and to decide which schema is to be used.

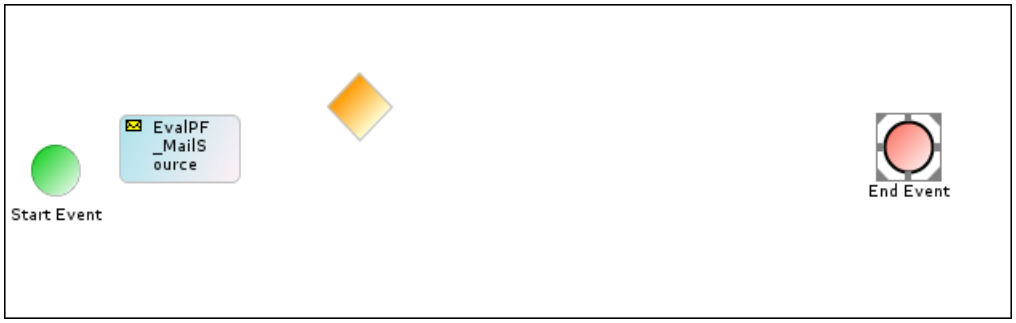

Figure 147: Drag BPMN Gateway Element to Graph Canvas Area

12. Click **[+] Schema** in the Activities Panel, and then click **[+] Excel Schema**. Select **EvalPF_ExcelSchema_Format1** activity and drag it to the Graph Canvas Area.

13. Similarly, drag **EvalPF_ExcelSchema_Format2** activity to the Graph Canvas Area.

14. Similarly, click **[+] DataTransform** and then **[+] Data Mapping**. Select **EvalPF_MappingTransformation_Format1** activity and drag it to the Graph Canvas Area.
15. Select **EvalPF_MappingTransformation_Format2** activity and drag it to the Graph Canvas Area.
16. After these activities are dragged to the Graph Canvas Area, they must be connected using appropriate BPMN Flows or Control Flows.
17. Click the **Sequence Flow** (⟶) icon in the Palette. The Sequence flow is selected.
18. To connect *Start Event* with *EvalPF_MailSource*, drag mouse pointer from *Start Event* to *EvalPF_MailSource* (see Figure 148).

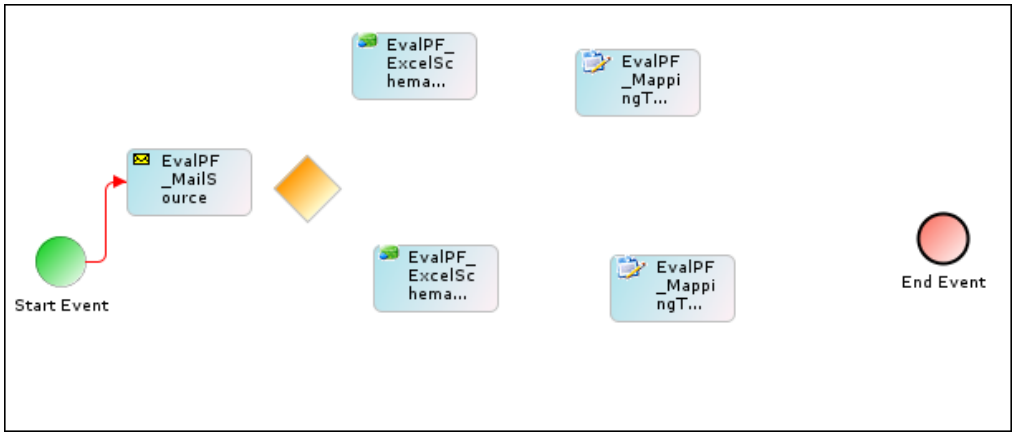

Figure 148: Connect Start Event to File Source Activity

19. Similarly, connect all other activities (see Figure 149).

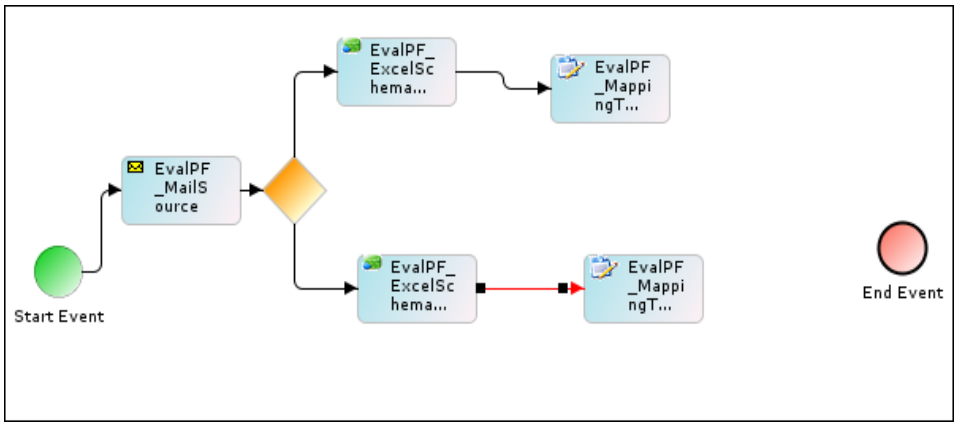

Figure 149: Connect all Activities

20. 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 150).

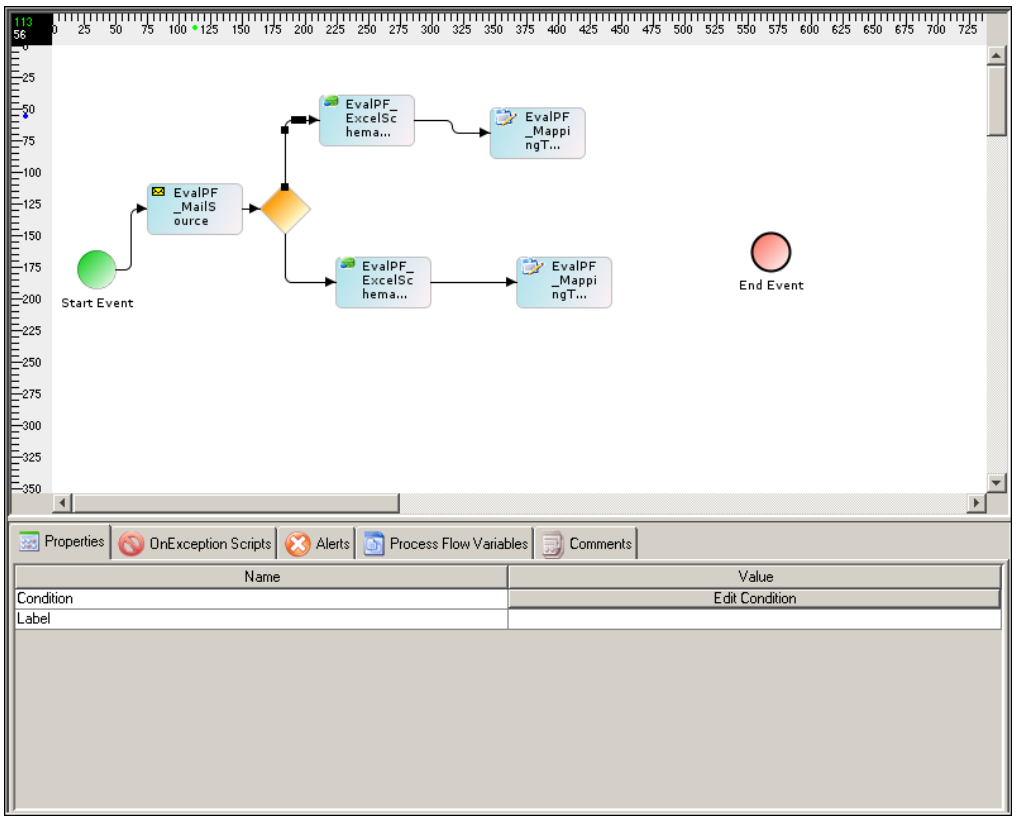

Figure 150: Define Decision Criteria

21. Click **Edit Condition** in the Properties Panel. The Condition Wizard is displayed (Refer to Figure 224).

22. Select **Process Flow Variable Condition** and click the **Next** button. The Process Flow Variable Condition type screen is displayed (see Figure 151).

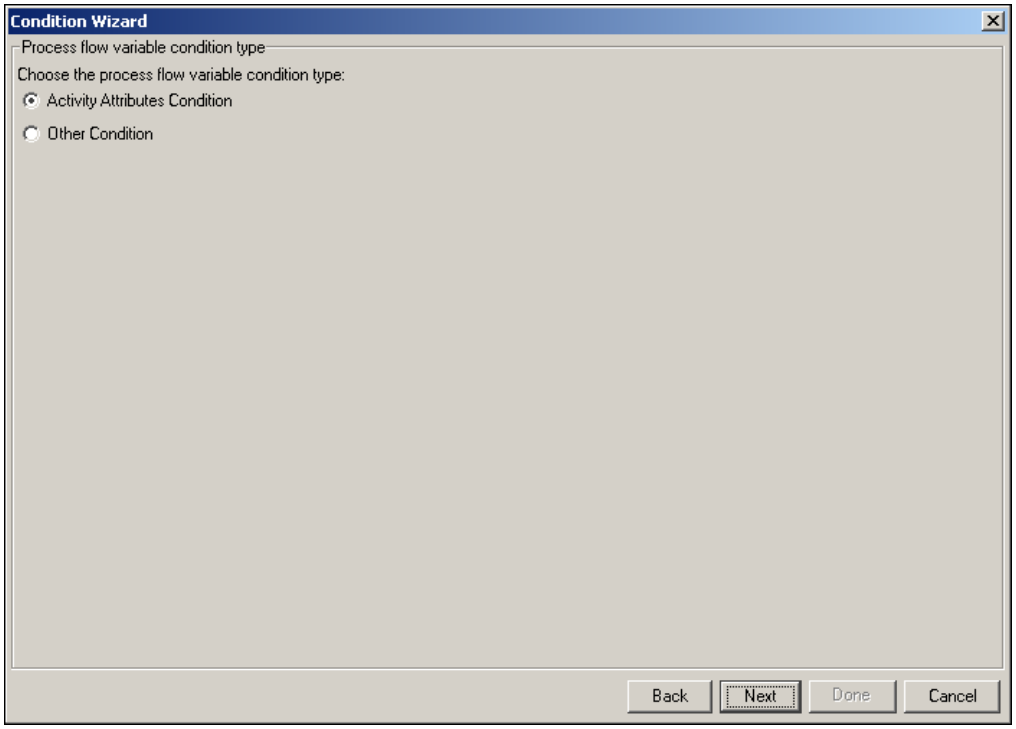

Figure 151: Process Flow Variable Condition Type

23. Select **Activity Attributes Condition** and click the **Next** button. The **Activity Attribute Condition** screen is displayed (see Figure 152).

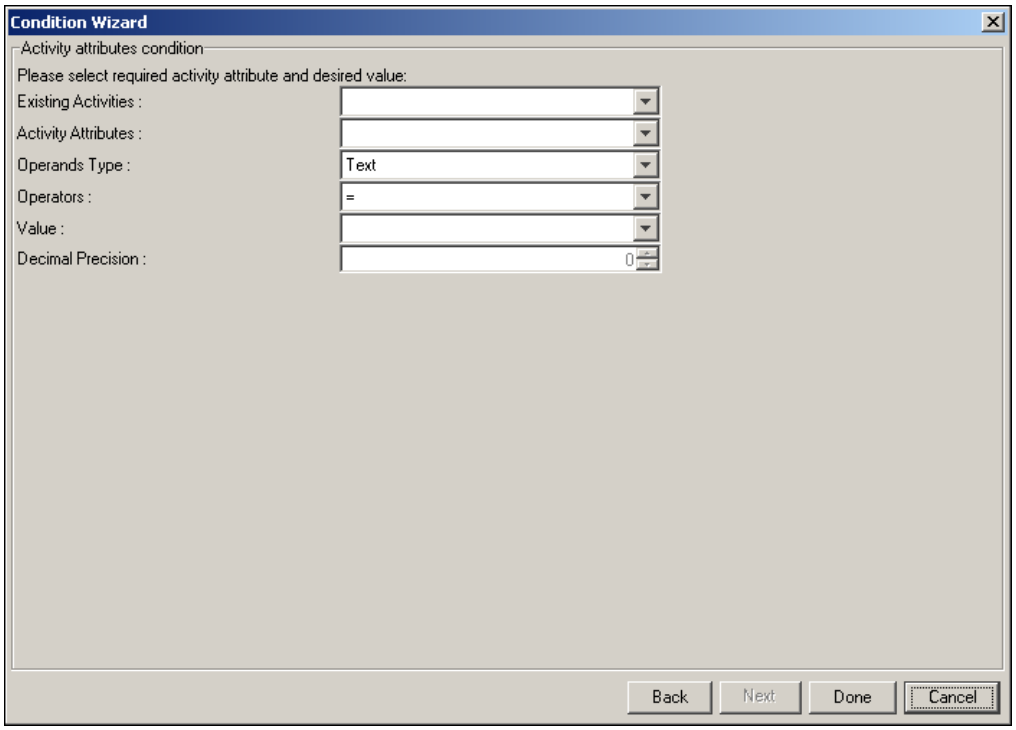

Figure 152: Select Activity Attributes

24. Select **EvalPF_MailSource** from the *Existing Activities* dropdown list, and select **mailSubject** from *Activity Attributes* dropdown list.
25. Select **Text** from the **Operand Type** dropdown list, and select = from the *Operators* dropdown list.
26. Enter *Format1* in the **Value** field (see Figure 153).

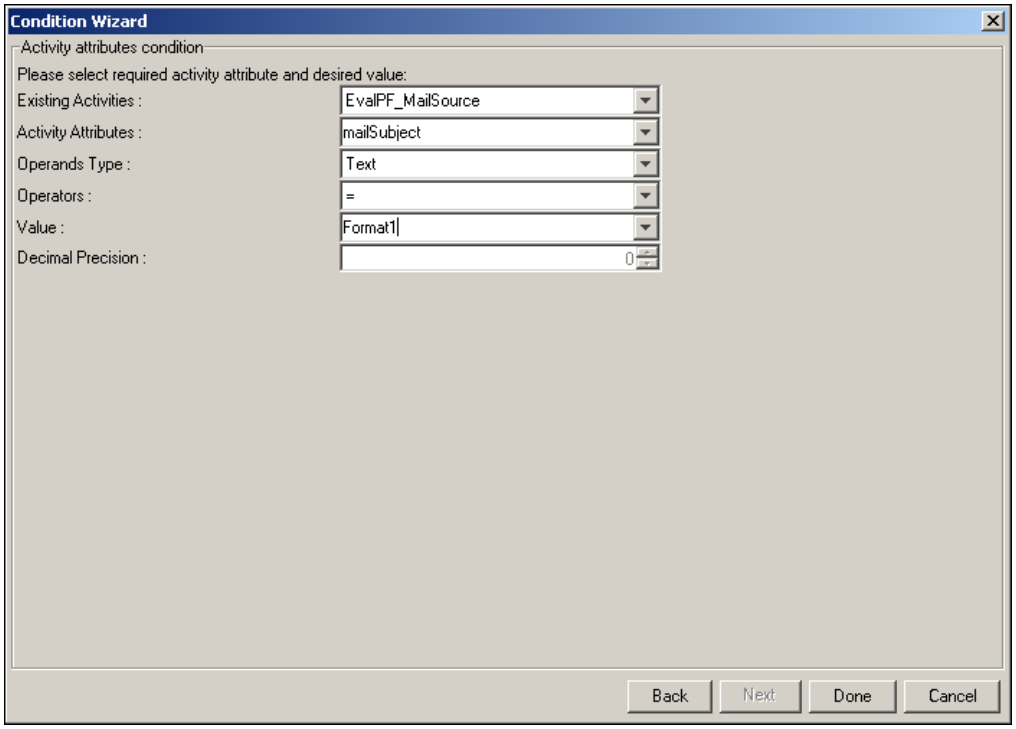

Figure 153: Select Attribute Value for *Format1*

27. Click the **Done** button to close the **Condition Wizard**.
28. Similarly, right-click the **BPMN Flow**, which is connecting *Gateway* and *EvalPF_ExcelSchema_Format2* activity and select **View Properties.**

29. Repeat the steps 26 to 30 and enter *Format2* in *Value* field (see Figure 154).

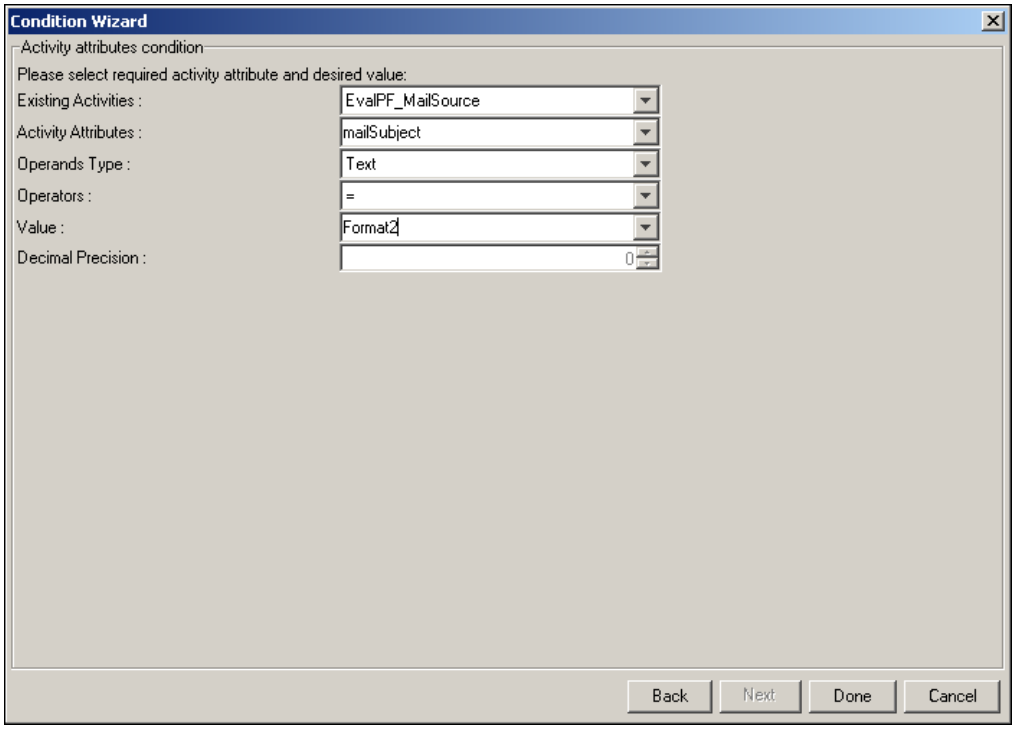

Figure 154: Select Activity Attributes for Format2

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

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

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

> An Action name is always unique.

33. Select **Put-Context-Var** and drag it to the Graph Canvas Area and change its *label* to **Seq1_File_Path** (see Figure 155).

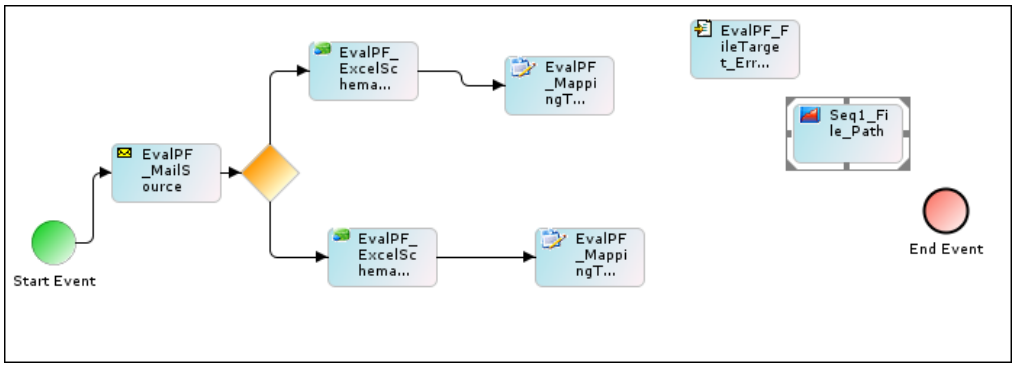

Figure 155: Drag *Put-Context-Var* to Graph Canvas Area

34. Connect EvalPF_MappingTarnsformation_Format1 with Seq1_File_Path and then Seq1_File_Path with EvalPF_FileTargetErrorRecord (see Figure 156).

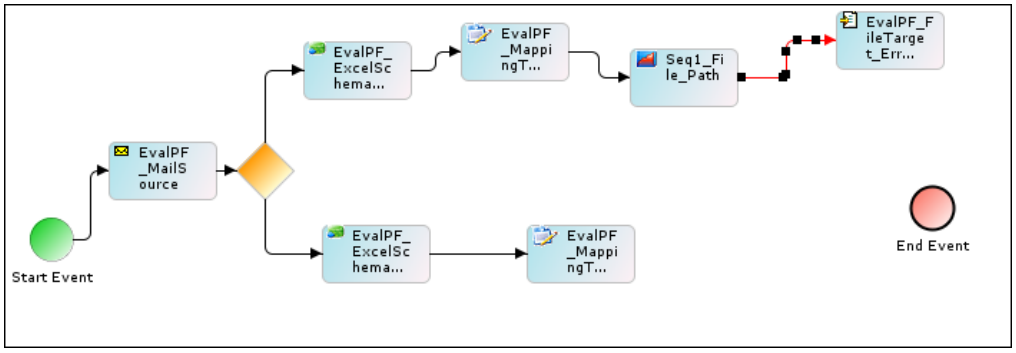

Figure 156: Connect Activities

35. 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 157).

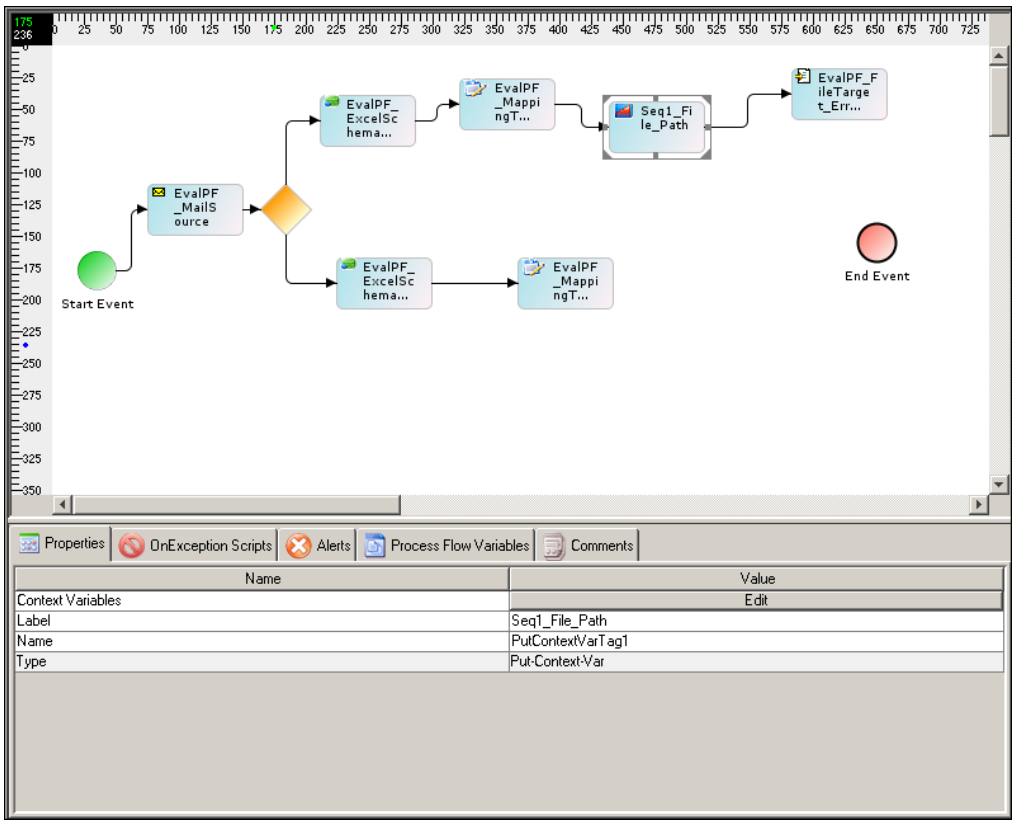

Figure 157: View Properties of *Seq_File_Path*

36. Click *Edit* from the *value* column of the *Context Variable* property. The **Edit Context Variables** screen is displayed (see Figure 158).

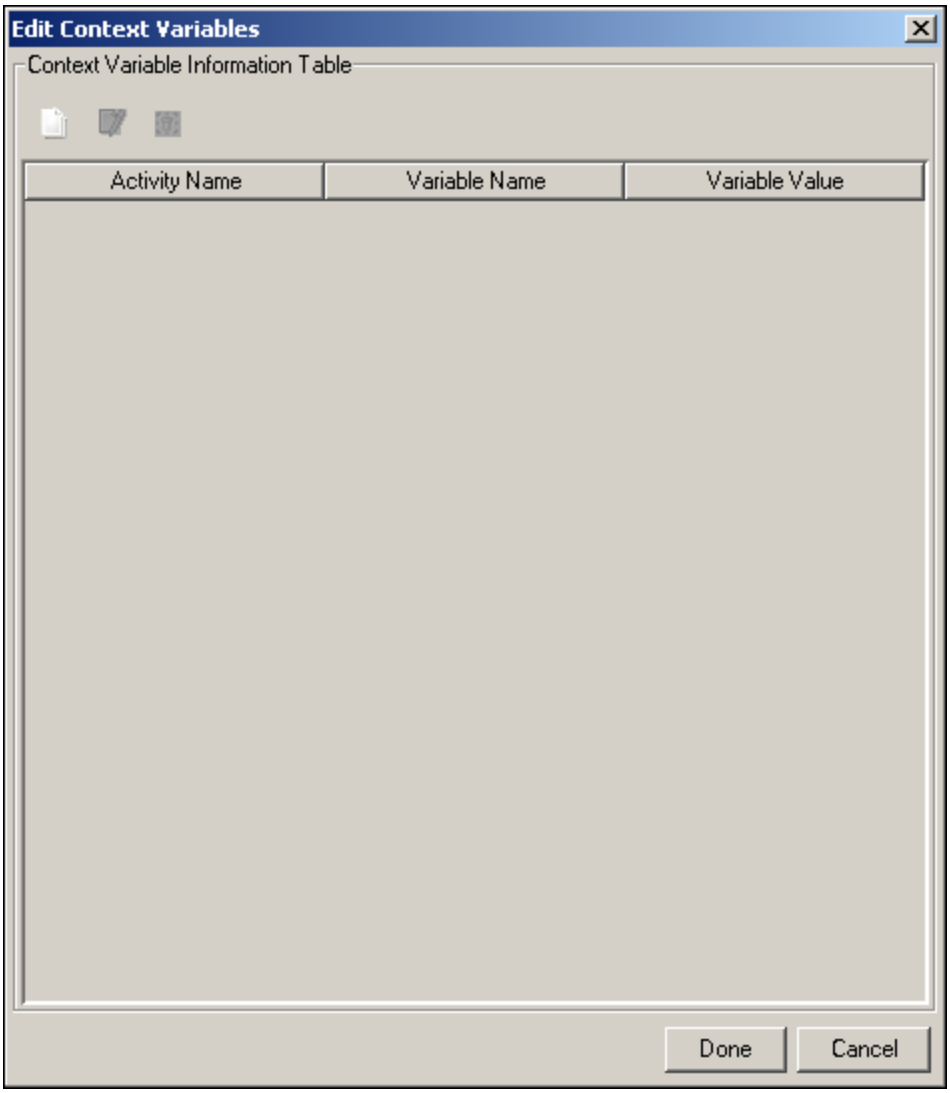

Figure 158: Edit Context Variable

37. Click the **New Context Variable** ( ) button. The **Context Variable Information** dialog box is displayed (see Figure 159).

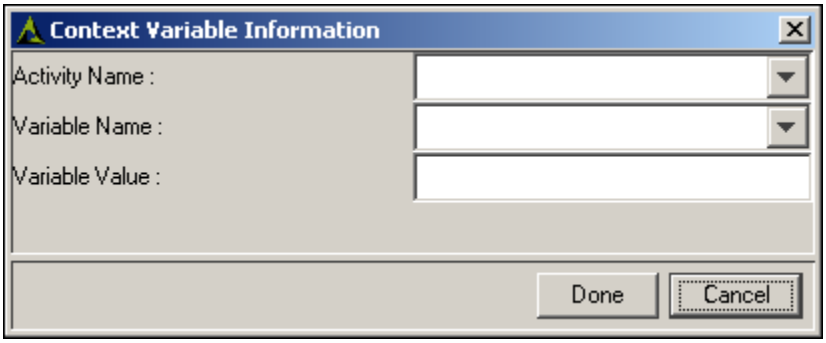

Figure 159: Add Context Variable

38. Select **EvalPF_FileTarget_ErrorRecord** from the Activity Name dropdown list.
39. Select **filepath** from the **Variable Name** dropdown list.

40. 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*

41. 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*

42. Click **[+] Target** and then **[+] Database Target**. Select **EvalPF_DatabaseTarget_Database1** activity and drag it to the Graph Canvas Area.

43. Similarly, drag **EvalPF_DatabaseTarget_Database2** activity to the Graph Canvas Area.

44. Connect these activities as shown in Figure 160.

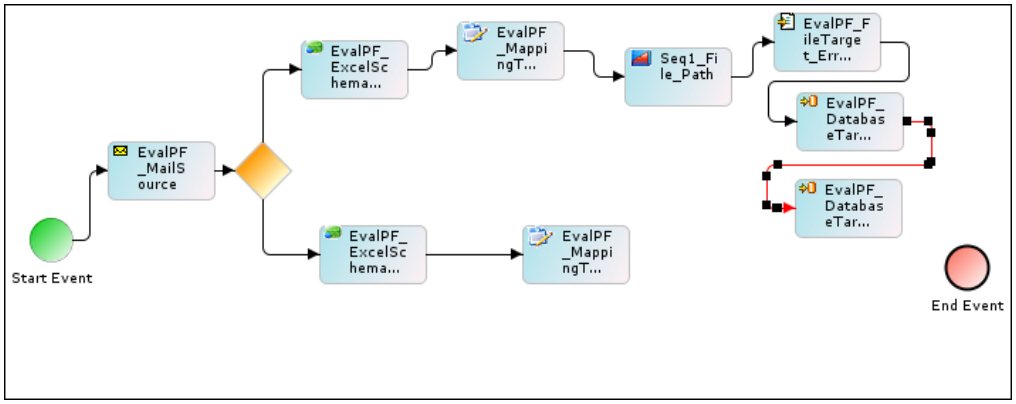

Figure 160: Connect Activities

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

46. Similarly, drag **EvalPF_FileTarget_UpdateError** activity to the Graph Canvas Area.

47. Click **Action** and drag the **Put-Context-Var** action in the Graph Canvas Area (see Figure 161).

    Right-click it and change the properties as shown in the table below.

Table 33: Changed Properties

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

48. 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 34: Changed Properties

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

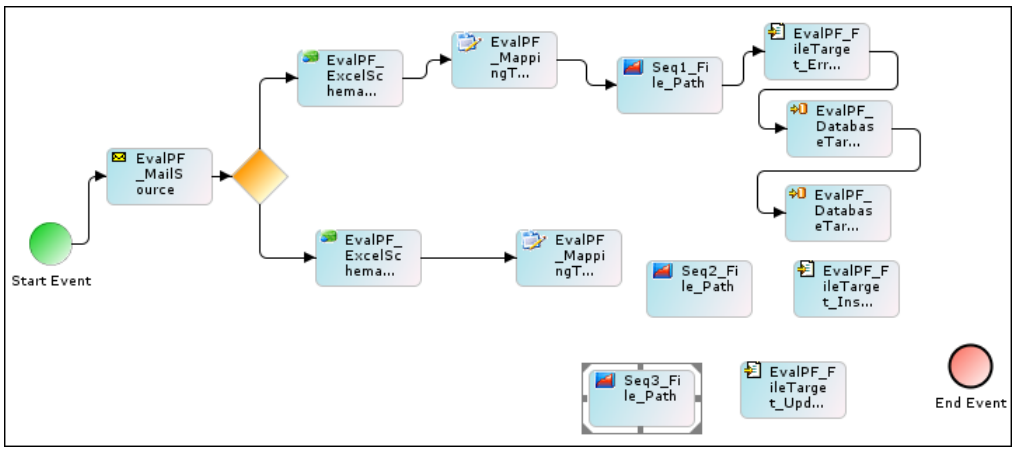

Figure 161: Drag *Put-Context-Var* to Graph Canvas Area

49. Connect these activities as shown in the Figure 162.

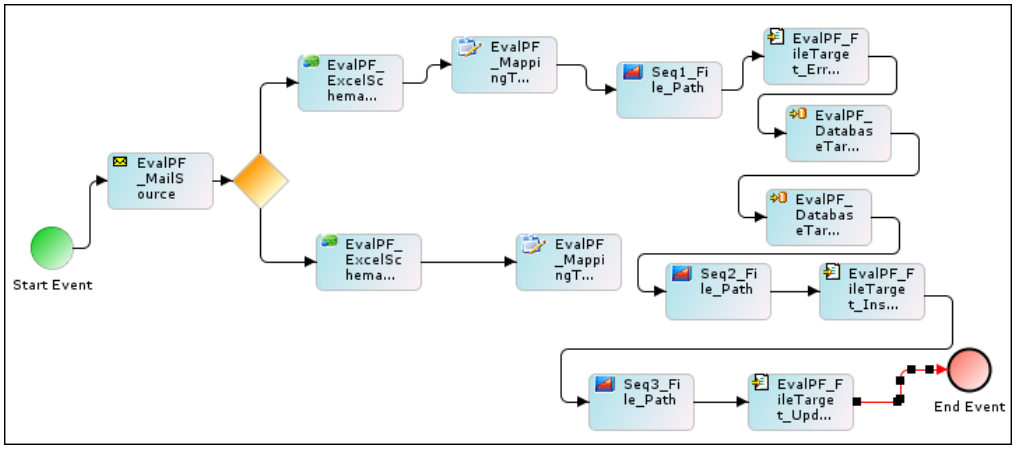

Figure 162: Connect Activities

50. Repeat the steps 36 to 55 to make another sequence of activities from **EvalPF_MappingTarnsformation_Format2** (see Figure 163).

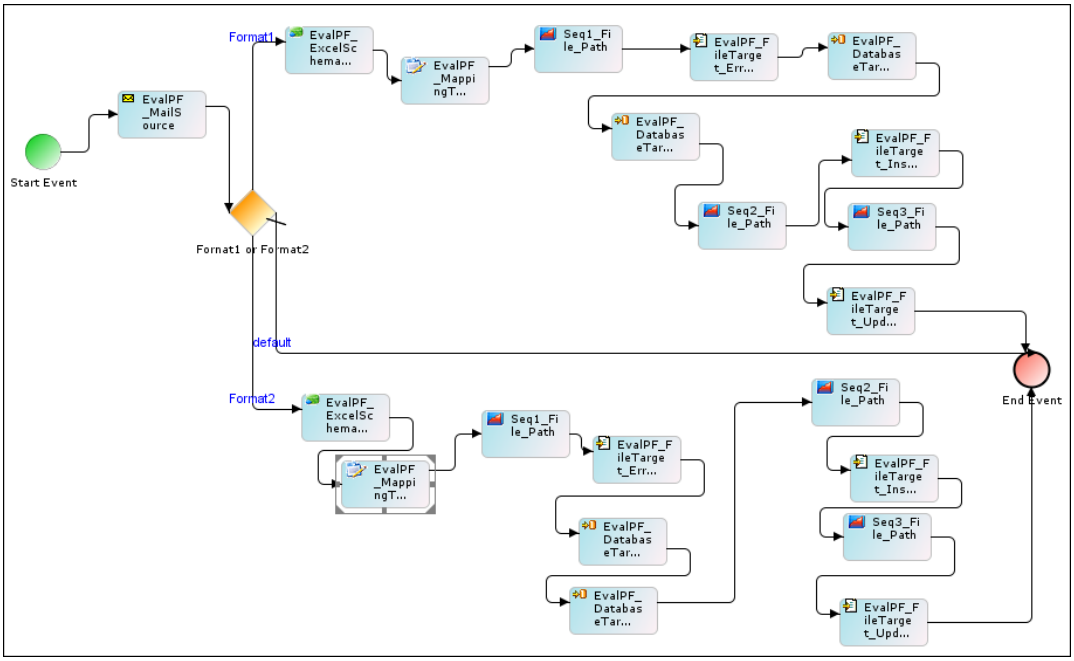

Figure 163: Connect Activities

- 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 163 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.

51. To create multiple streams, right-click the **EvalPF_MailSource** activity and select **Multiple Stream**. The Multiple Stream dialog box is displayed (Refer to Figure 83).
52. Enter any name for the Data Stream (default) in the **Steam Name** field and click **Add Stream**.
53. Ensure that **EvalPF_ExcelSchema_Format1** is selected in the **Activities** dropdown list.
54. Click the **Map** button. A stream between **EvalPF_MailSource** and **EvalPF_ExcelSchema_Format1** is created.

55. Make sure the **Explicit Stream** checkbox is selected. A stream between **EvalPF_MailSource** and **EvalPF_ExcelSchema_Format1** is created.

56. To create another stream, click the **Activities** dropdown list and select **EvalPF_ExcelSchema_Format2** and then click the **Map** button. A dialog box is displayed (Refer to Figure 108).

57. Click the **Yes** button. Another stream between **EvalPF_MailSource** and **EvalPF_ExcelSchema_Format2** is created.

58. Click the **Done** button to close the **Multiple Stream** Dialog box. Streams created are shown in the Graph Canvas area (see Figure 164).

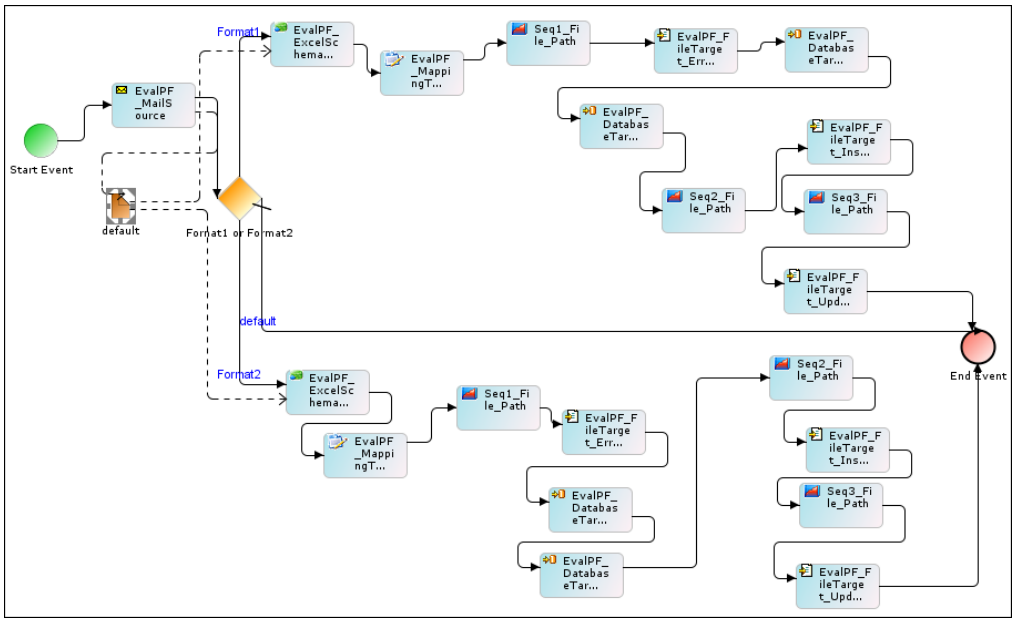

Figure 164: Multiple Streams Created

59. Repeat steps 57 to 64 to create another streams between activities listed below (see Figure 165):

- 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

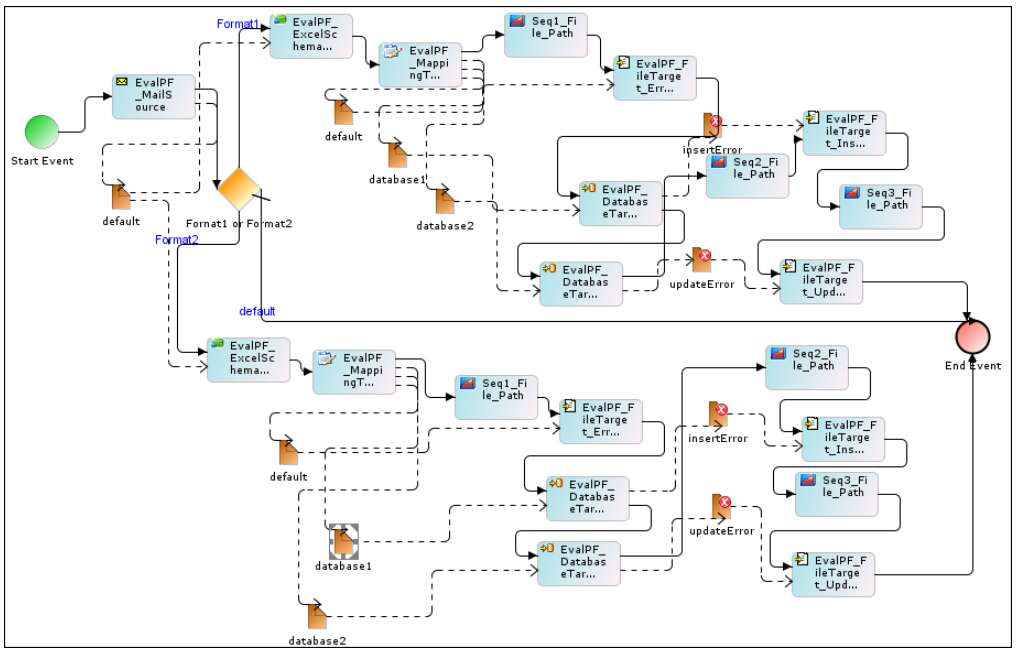

Figure 165: Multiple Streams Created for Multiple Activities

60. 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.

61. Click the **Done** button to close the dialog box.

62. 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. On the Adeptia Suite homepage menu, click the **Develop** tab.

2. Go to **Events > Event Registry**. This displays the *Manage Event Registry* page with the list of existing Event Registry activities (see Figure 166).

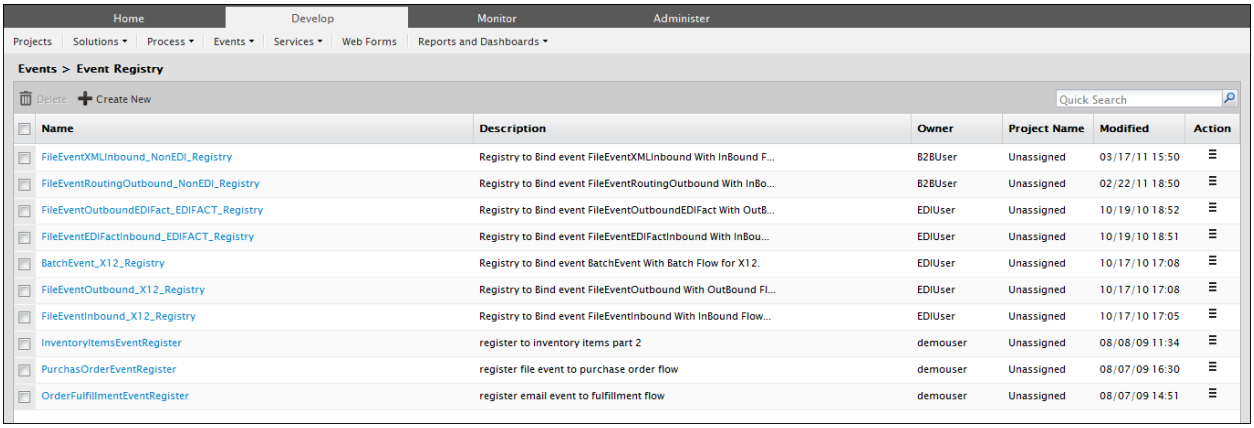

Figure 166: Manage Event Registry

3. Select the required Event Registry activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Event Registry activity in edit mode (see Figure 167).

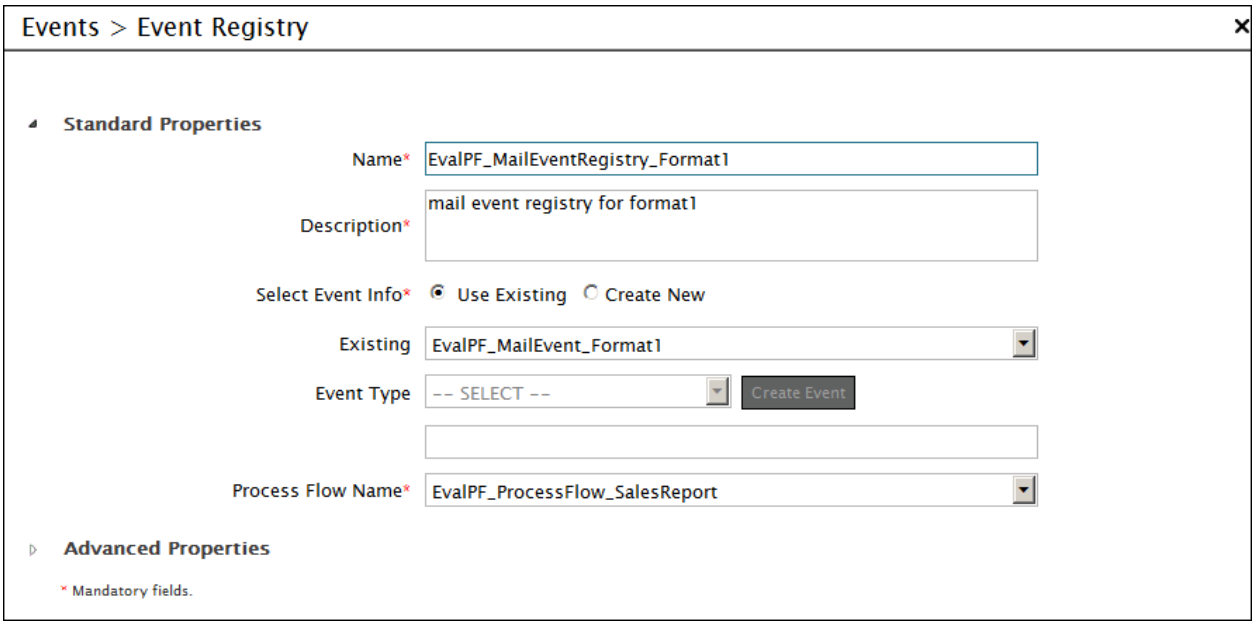

Figure 167: Edit *EvalPF_MailEventRegistry_Format1* Activity

The following table defines the fields on this screen.

Table 35: Details of Fields on Edit Event Registry Screen

Field Name	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

Field Name	Description
Process Flow Name	Name of the Process Flow, which is triggered by JMS Event

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 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 175).
7. Enter comments in the **Add Comments** field.
8. 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.

8

JMS EVENT DRIVEN PROCESS FLOW

This section describes the JMS Event Driven Process Flow.

In the Adeptia Suite this process flow is available in:

EBIM Suite	BPM Suite	ESB 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 168):

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

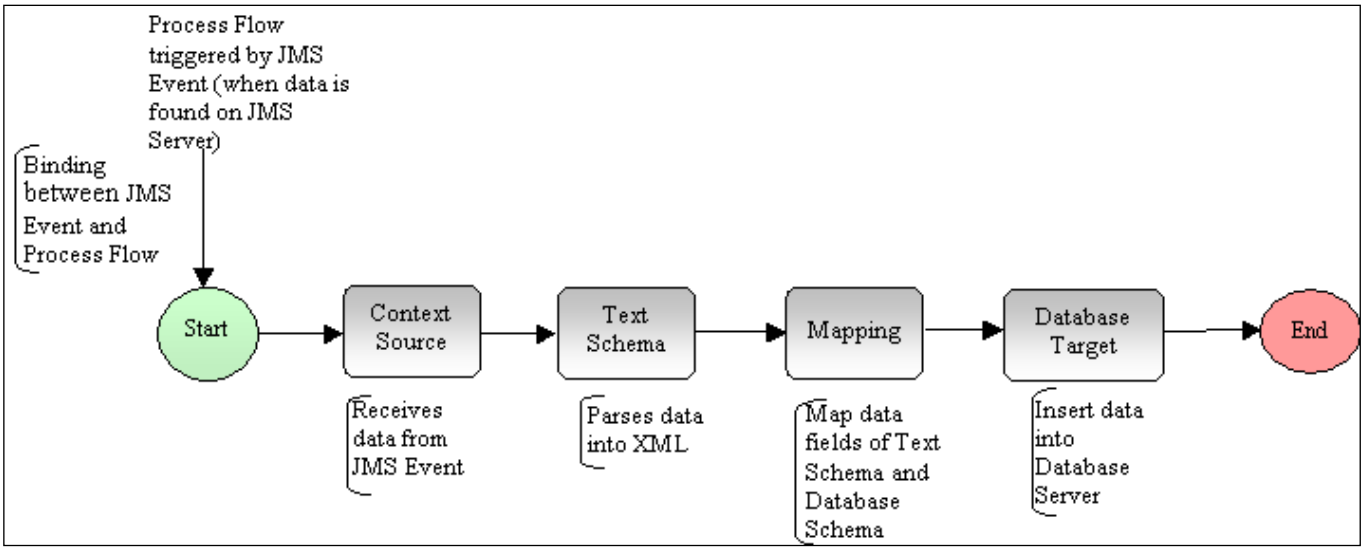

Figure 168: Flow Chart showing JMS Driven Process Flow

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:

1. **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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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.

7. **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 36: 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

Field Name	Description	Data Type
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

The comment should be at least 1 character in length.

Database Tables also have same fields as shown in the Table 36.

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/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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Events > JMS**. The *Manage JMS Event* screen is displayed with the list of existing JMS Events (see Figure 169).

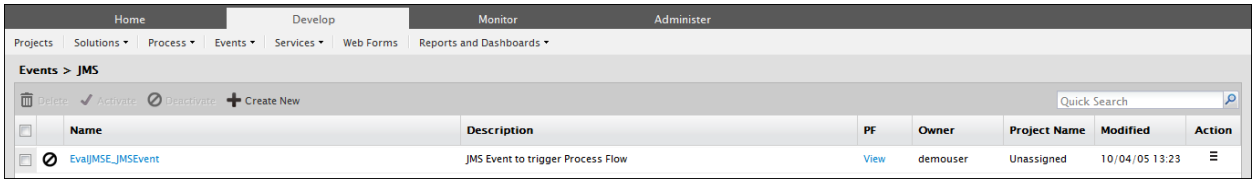

Figure 169: Manage JMS Event

4. Select the required JMS Event activity.
5. Click the **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 170).

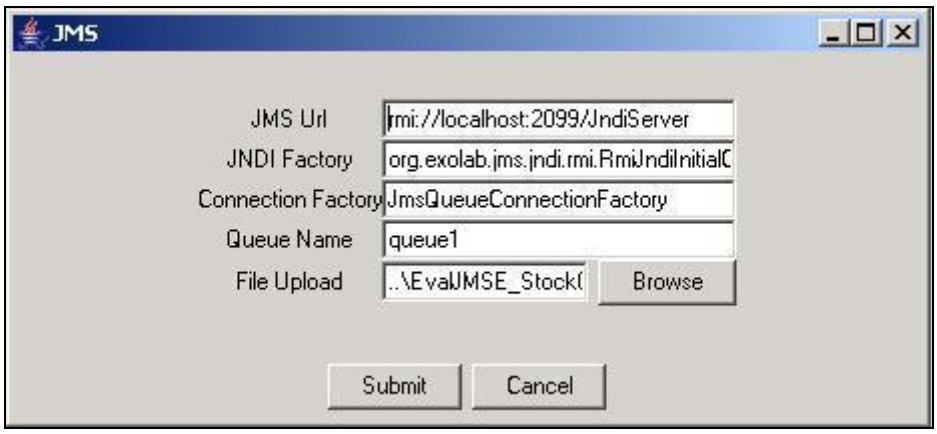

Figure 170: Send File to OpenJMS Server

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. On the Adeptia Suite homepage menu, click the **Monitor** tab.

   Go to History > Process Flows Logs (see Figure 171).

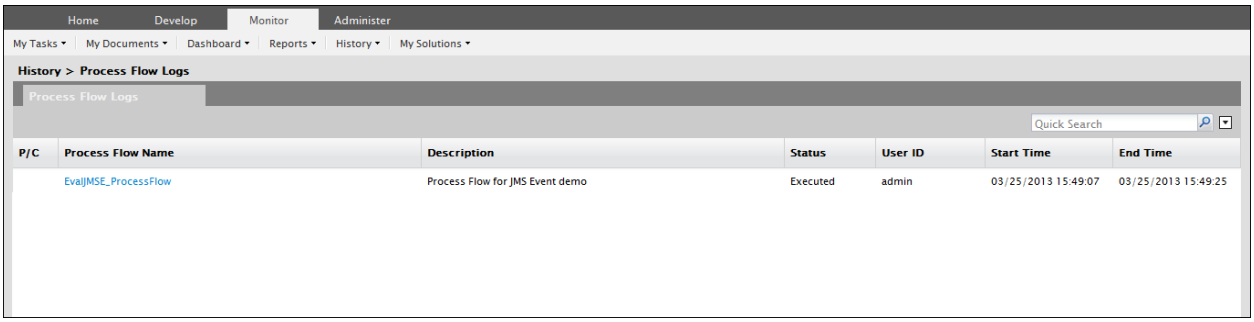

Figure 171: Process Flow Log

2. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 172).

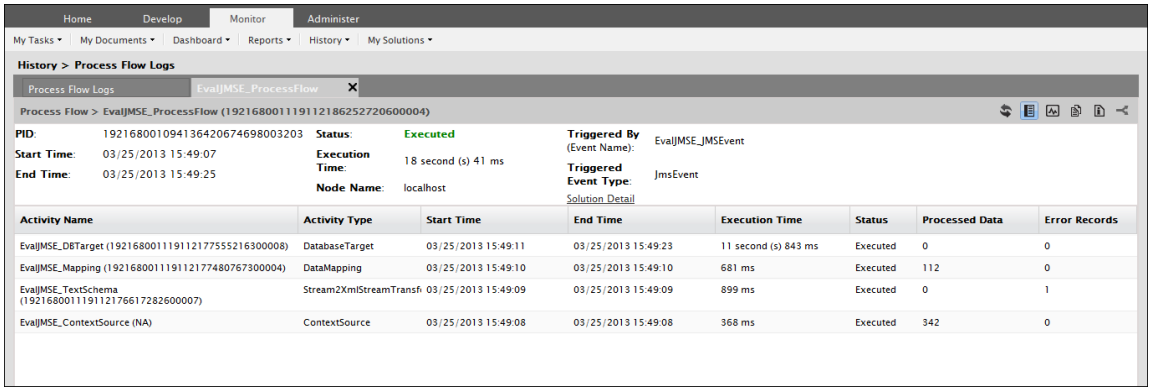

Figure 172: Searched Process Flows

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. On the Adeptia Suite Home page, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **JMS Provider**. The *Manage JMS Provider* screen is displayed with the list of existing JMS Provider activities (see Figure 173).

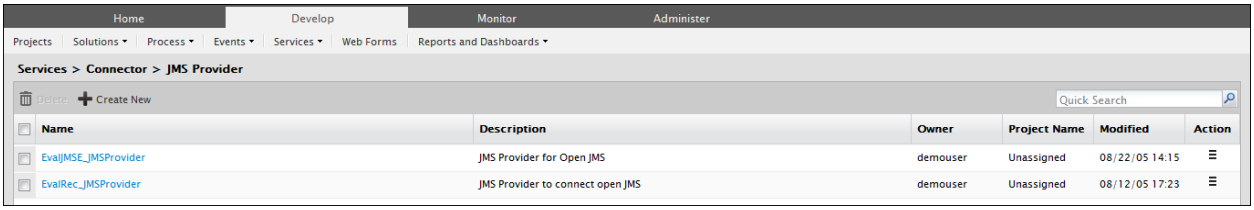

Figure 173: Manage JMS Provider

4. Select the required JMS activity and right-click the same to view the **More Actions** menu.
5. Click **Edit** to open the respective JMS Provider activity in edit mode (see Figure 174).

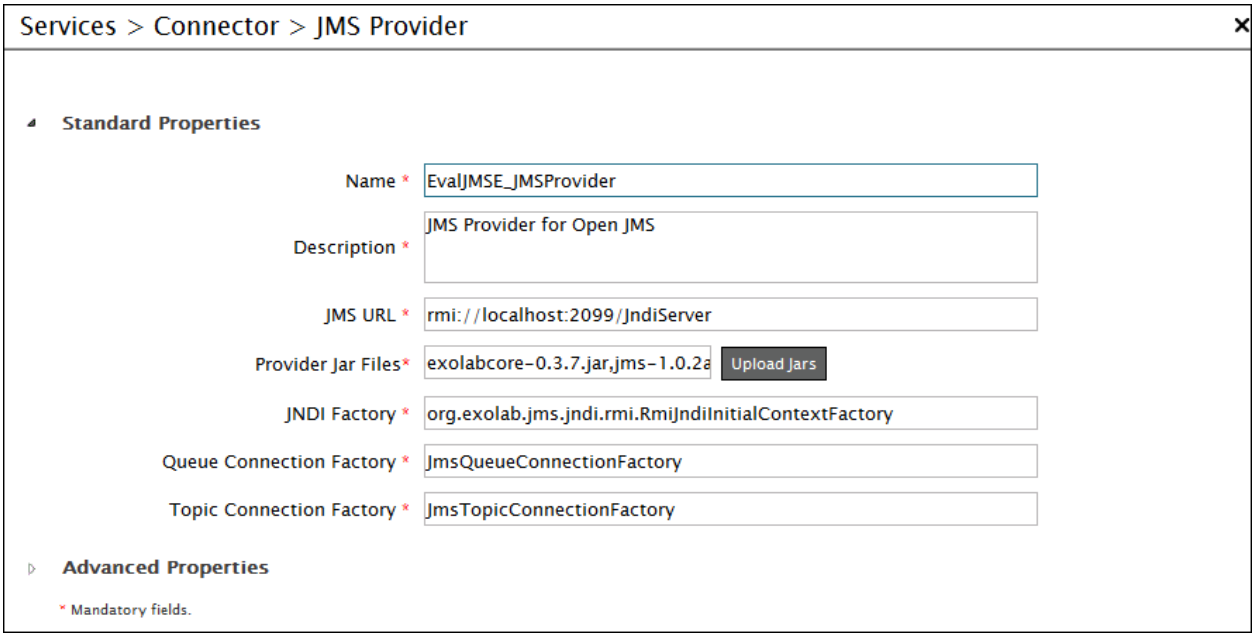

Figure 174: Edit JMS Provider Activity

A detailed The following table defines the fields on this screen.

Table 37: Details of Fields on Edit JMS Provider Screen

Field Name	Description
Name	Name of the JMS Provider
Description	Description of the JMS Provider
JMS URL	URL of the JMS Server

Field Name	Description
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.

6. Make the necessary changes.
7. 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 175).

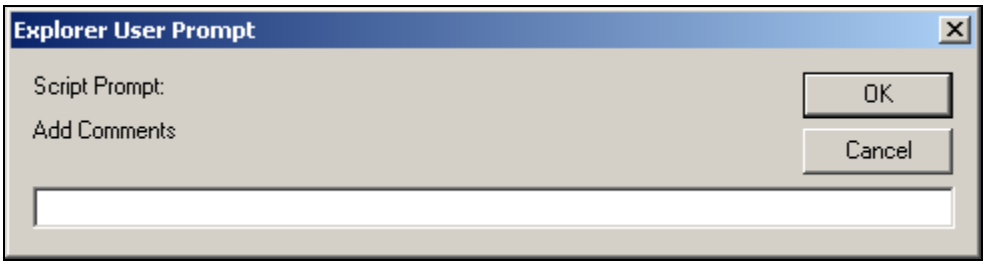

Figure 175: Enter Comments

8. Enter comments in the **Add Comments** field.
9. 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Events > JMS**. The *Manage JMS Event* screen is displayed with the list of existing JMS Event activities (Refer to Figure 169).
3. Select the required JMS Event activity and right-click the same to view the More Actions menu.
4. Click **Edit** to open the respective JMS Event activity in edit mode (see Figure 176).

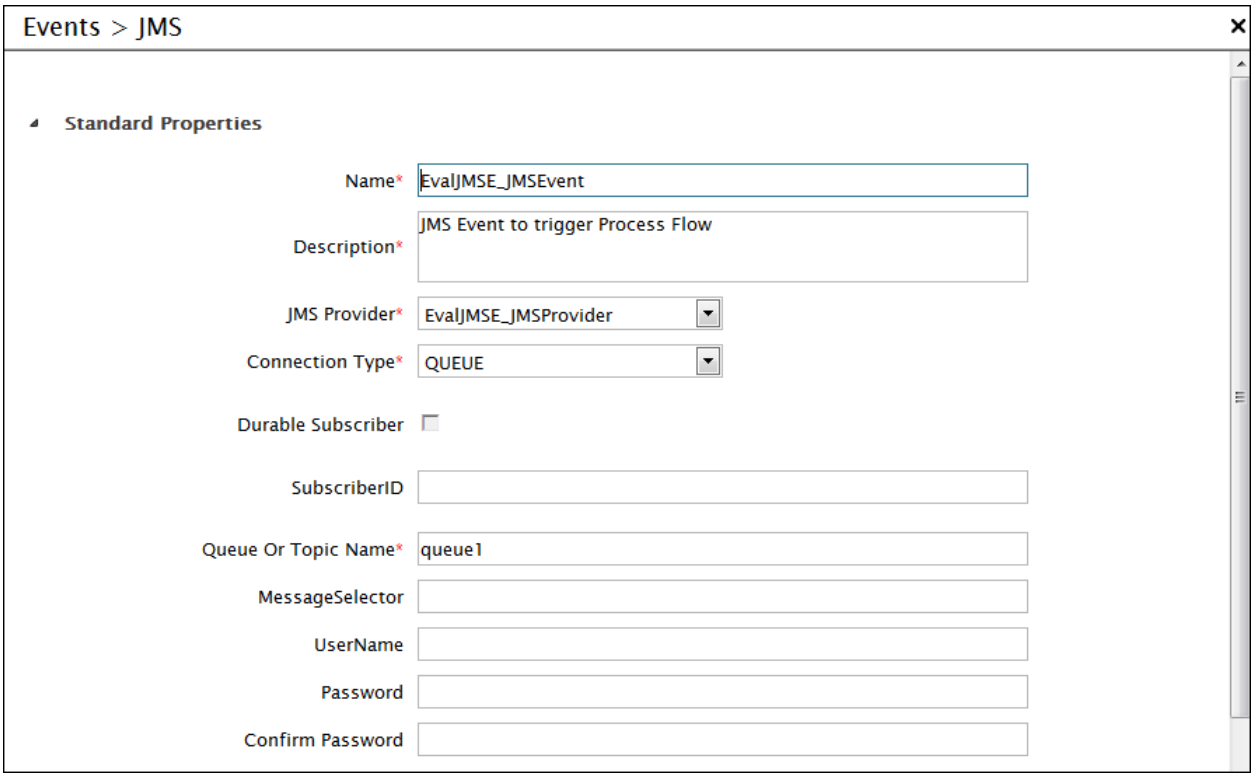

Figure 176: Edit JMS Event Activity

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

Table 38: Details of Fields on Edit JMS Events Screen

Field Name	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 TOPIC: Used for one to many messaging. It supports publish subscribe

Field Name	Description
	model of messaging. QUEUE: Used for one-to-one messaging. It supports Point-to-Point Messaging.
Durable Subscriber	If a client needs to receive all the messages published on a 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

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 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 175).
7. Enter comments in the **Add Comments** field.
8. 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Services** > **Schema**, and then click **Text**. The *Manage Text Schema* screen is displayed with the list of existing Text Schemas (refer to Error! Reference source not found.).
3. Select the required Text schema activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective Text schema activity in edit mode (see Figure 177).

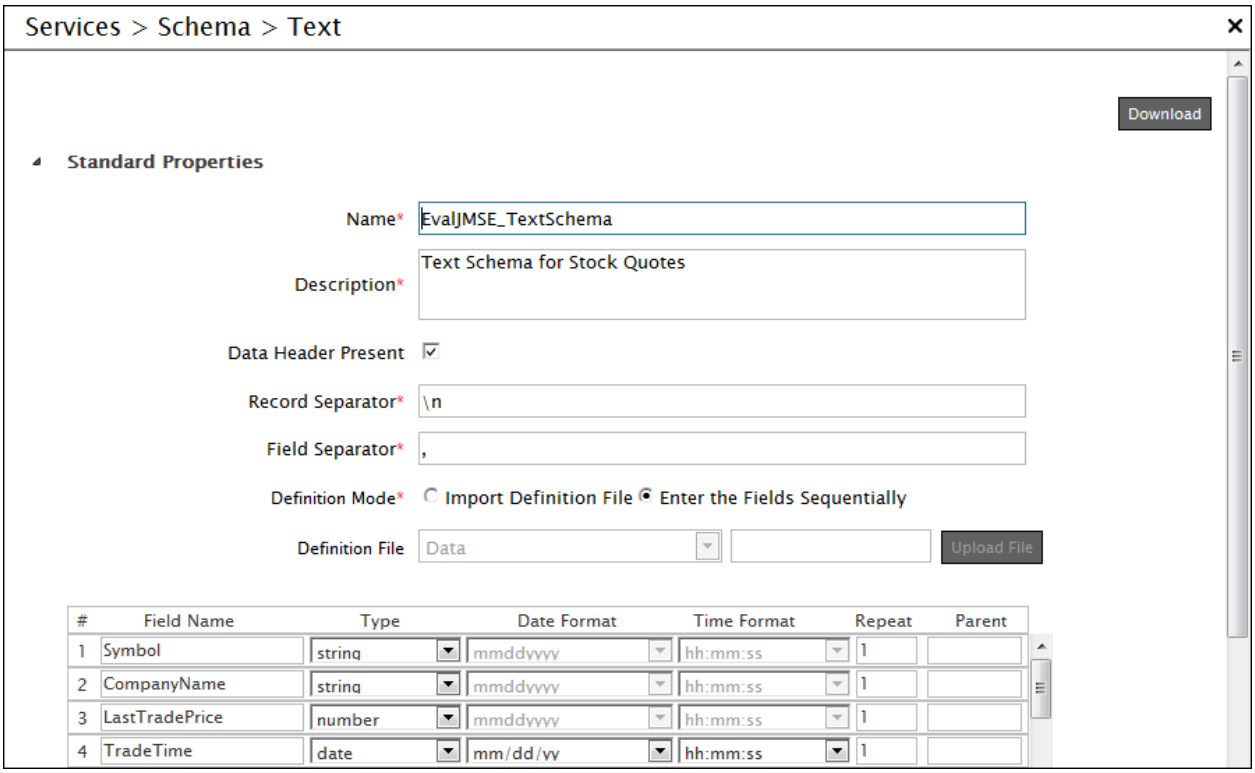

Figure 177: Edit Text Schema Activity

The following table defines the fields on this screen.

Table 39: Details of Fields on Edit Text Schema Screen

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

Field Name	Description
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: Use Definition File Enter the Field 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 following data types: String: String can be used for any type of data Number: Contains numbers 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 (Chocolate20perpack) 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. 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 175).

7. Enter comments in the **Add Comments** field.
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 178).

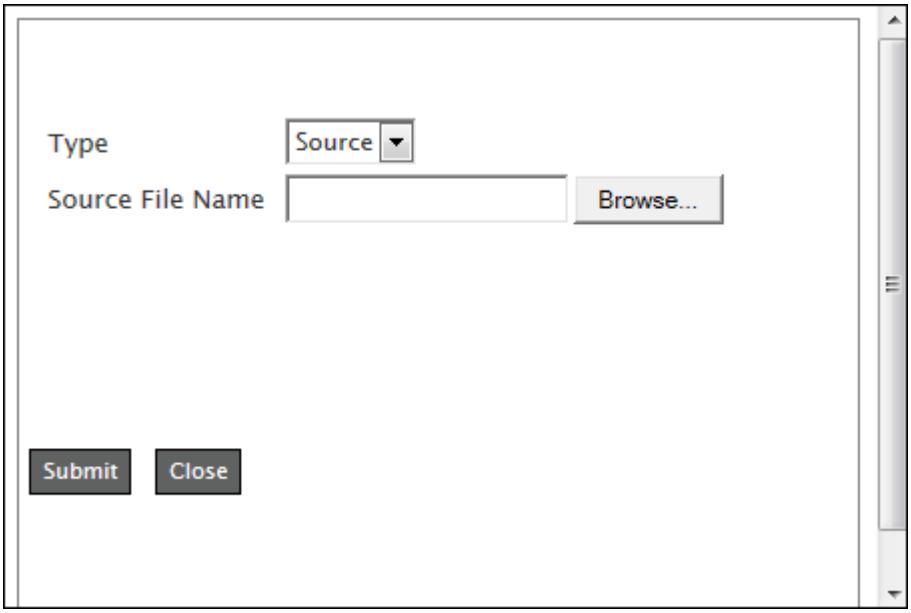

Figure 178: Test Schema

2. Select the type of schema to test, from the **Type** dropdown 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to Services > Data Transformation and then click Data Mapping.

The *Manage Data Mapping* screen is displayed with the list of existing activities (refer to Figure 26).

3. Select the required mapping activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to view open the respective mapping activity in edit mode(see Figure 179).

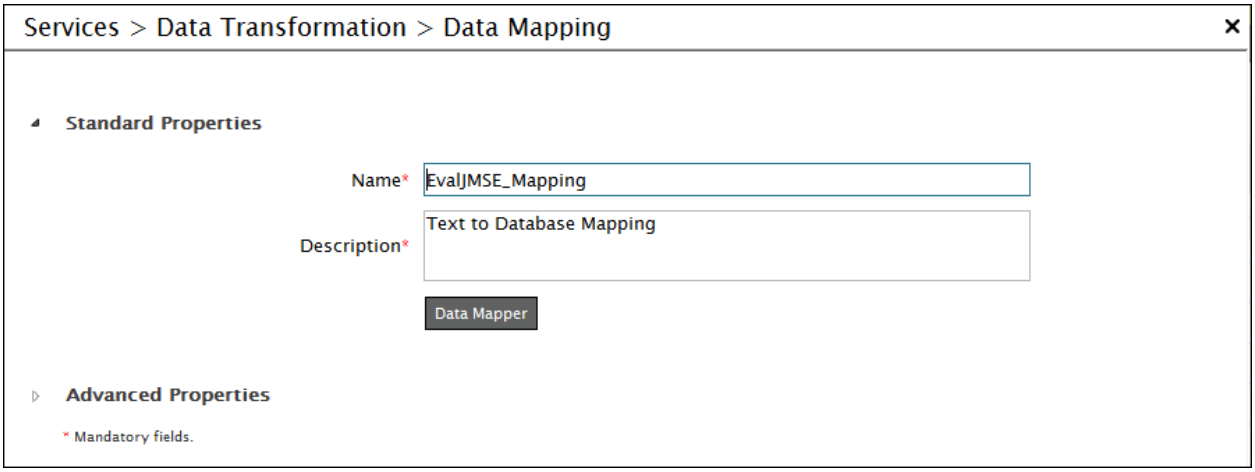

Figure 179: Edit *EvalJMSE_Mapping* Activity

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 180).

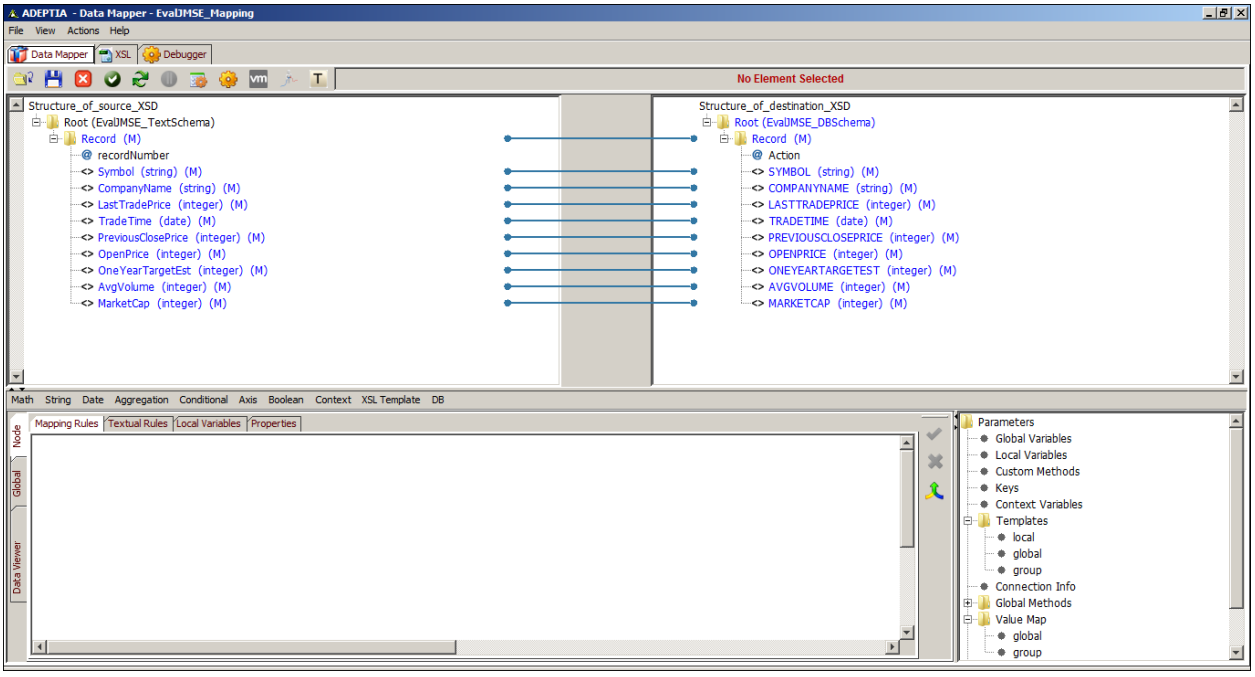

Figure 180: Mapping between Source and Target Schemas

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 181).

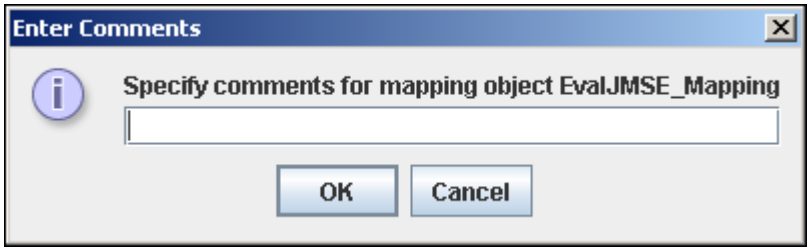

Figure 181: Enter Comments (Mapping)

9. Enter the comments in the Specify comments for mapping object (object name) field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **Database Driver**. The *Manage Database Driver* screen is displayed with the list of existing database drivers (refer to Figure 6).
4. Select the required Database Driver activity and right-click it to view the **More Actions** menu.
5. Click **Edit** link. This opens the respective Database Driver activity in the edit mode (see Figure 182).

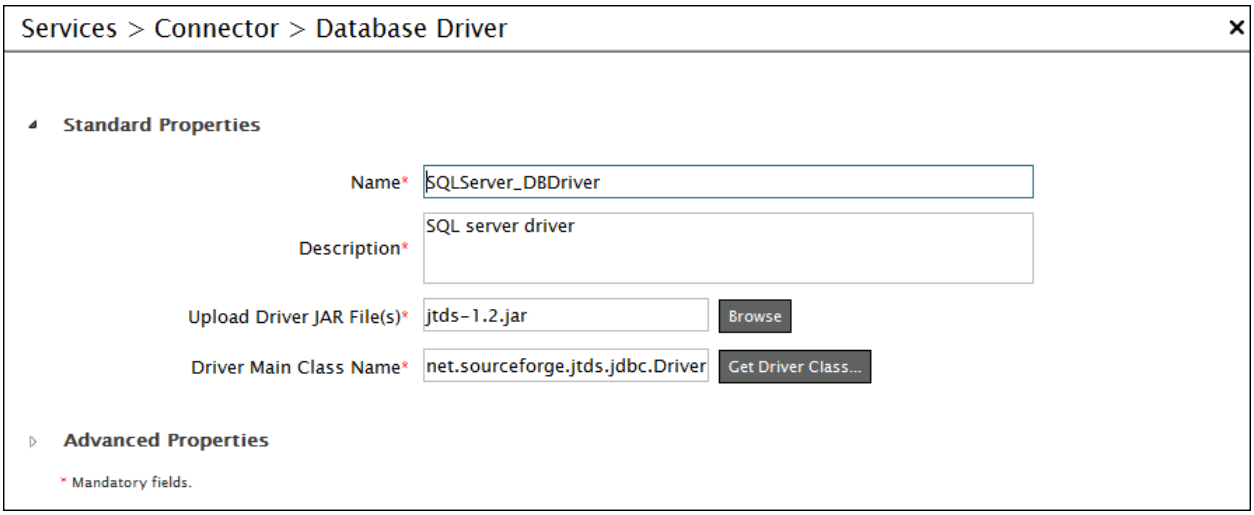

Figure 182: Edit *SQLServer JTDS Driver*

The following tables define the fields on this screen.

Table 40: Details of Fields on Edit Database Driver Screen

Field Name	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: Oracle – ojdbc5.jar IBM DB2 (Ver 7.1) – db2java.zip (7.1 version) IBM DB2 (Ver 8.1) – db2jcc.jar MS SQL – msbase.jar, mssqlserver.jar and msutil.jar JTDS- SQL Server – Jtds.jar 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 dropdown 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

6. Make the necessary changes.
7. 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 175).

8. Enter comments in the **Add Comments** field.
9. 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **Database Info**. The *Manage Database Info* screen is displayed with the list of existing Database Info (refer to Figure 8).
4. Select the required Database Info activity and right-click the same to view the **More Actions** menu.
5. Click **Edit** to open the respective Database Info activity in edit mode (see Figure 183).

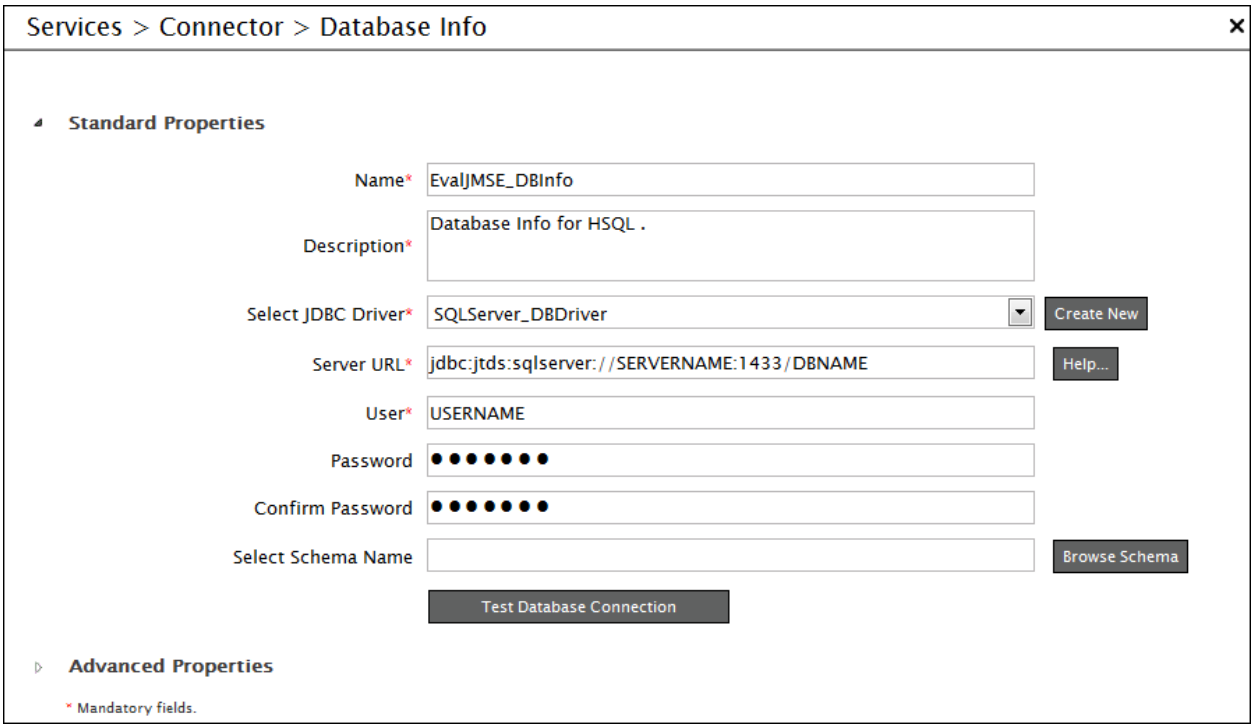

Figure 183: Edit *EvalJMSE_DBInfo*

6. Make the necessary changes.
7. 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 175).
8. Enter comments in the **Add Comments** field.

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

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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Schema** and then click **Database**.

   The *Manage Database Schema* screen is displayed (refer to Figure 10).
3. Select the required database activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Database schema activity in edit mode (see Figure 184).

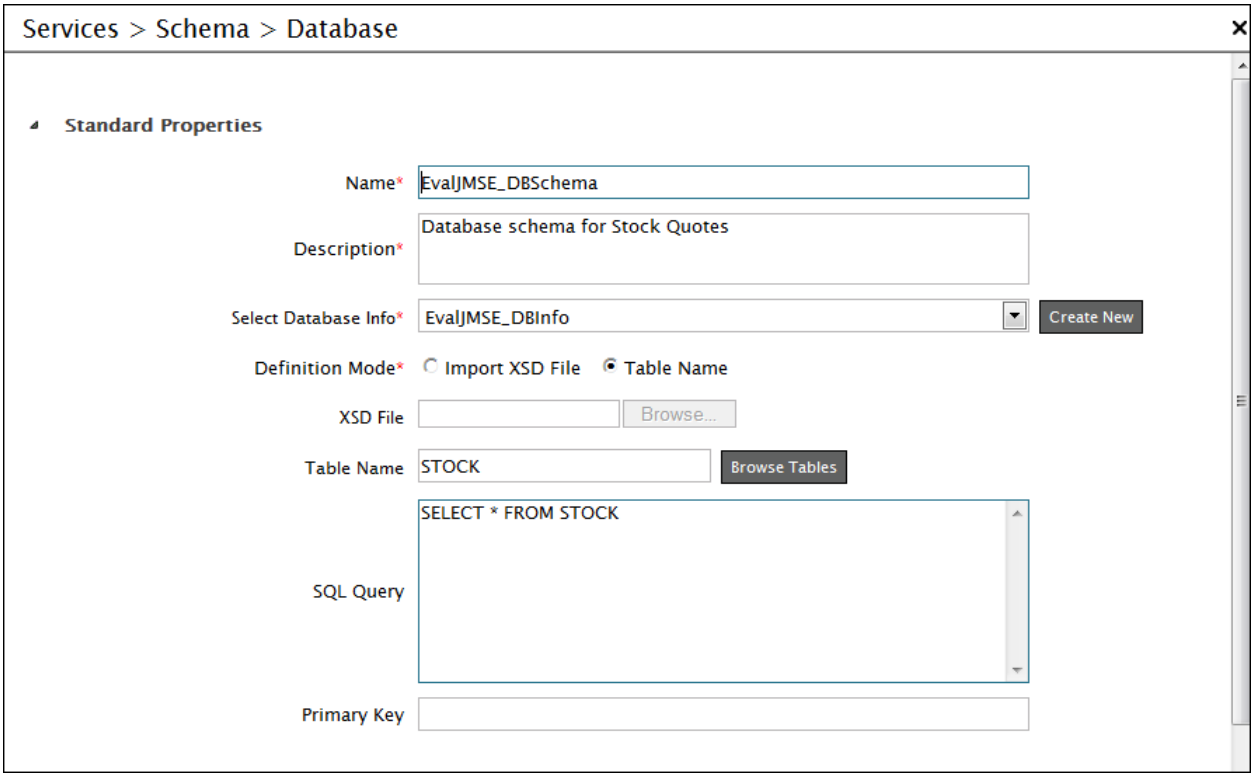

Figure 184: Edit *EvalJMSE_DBSchema* Activity

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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **Database**. The *Manage Mail Database* screen is displayed with a list of existing Database Target activities (refer to Figure 128).
3. Select the required Database target activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Mail target activity in edit mode (see Figure 185).

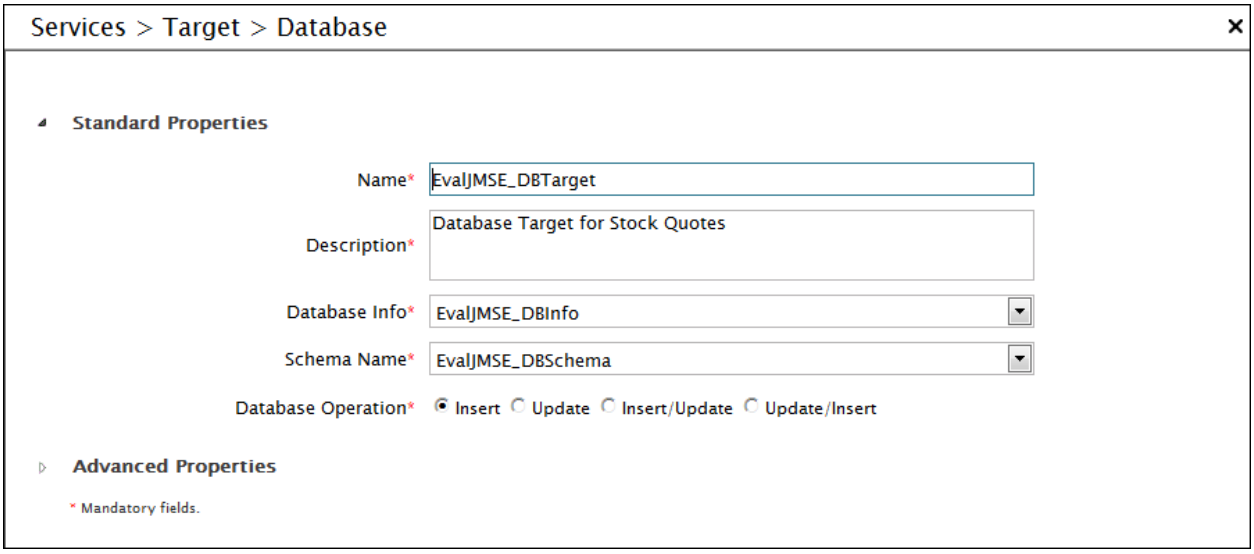

Figure 185: Edit *EvalJMSE_DBTarget* Activity

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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (refer to Figure 55).
3. Click the **Create New** link. The *Create Process Flow* screen is displayed (refer to Figure 56).
4. Enter the name and description of the Process Flow.
5. Select the logging level from the *Logging Level* dropdown list.
6. Select repository file retention from the **Repository File Retention** option.

7. Click the **Process Designer** button to open Process Designer. The *Process Designer* screen is displayed (refer to Figure 57).

> For more information on how to create process flows and on *Process Designer* screen, refer to the Creating Process Flow section.

> The Activities **Panel** has been renamed to **Repository View**. It is not reflected in this.

8. Click **OK** to synchronize.
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 186).

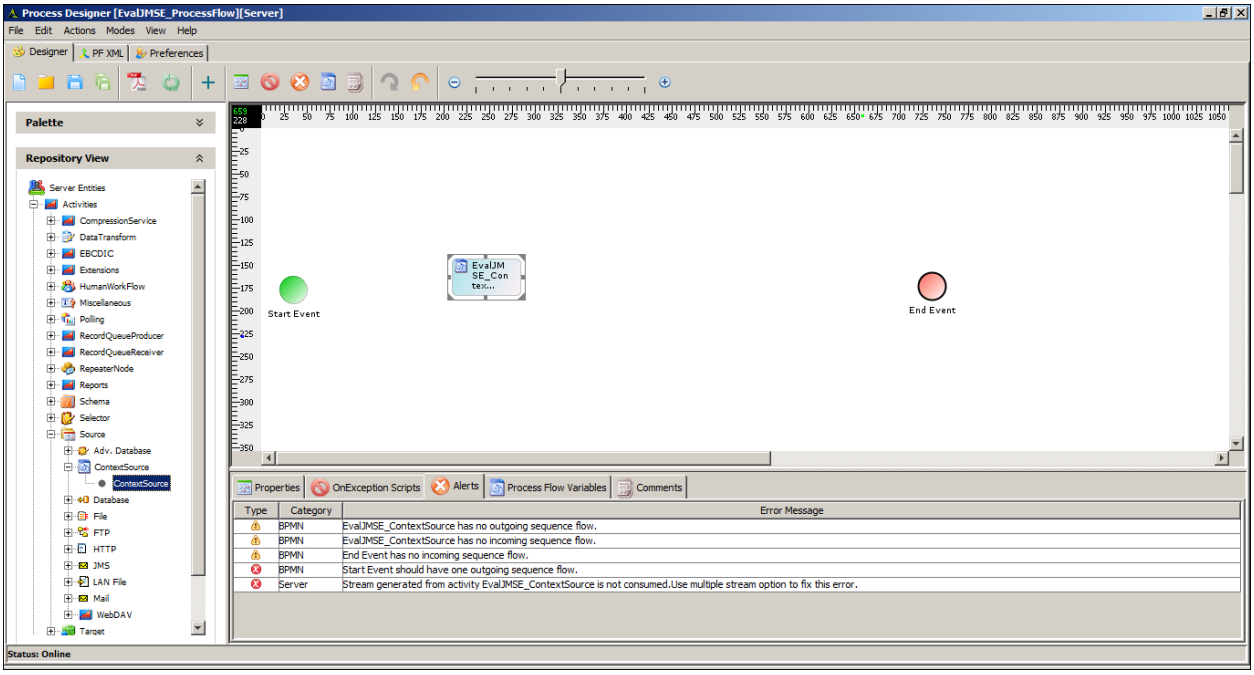

Figure 186: Drag Context Source to Graph Canvas

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 187).

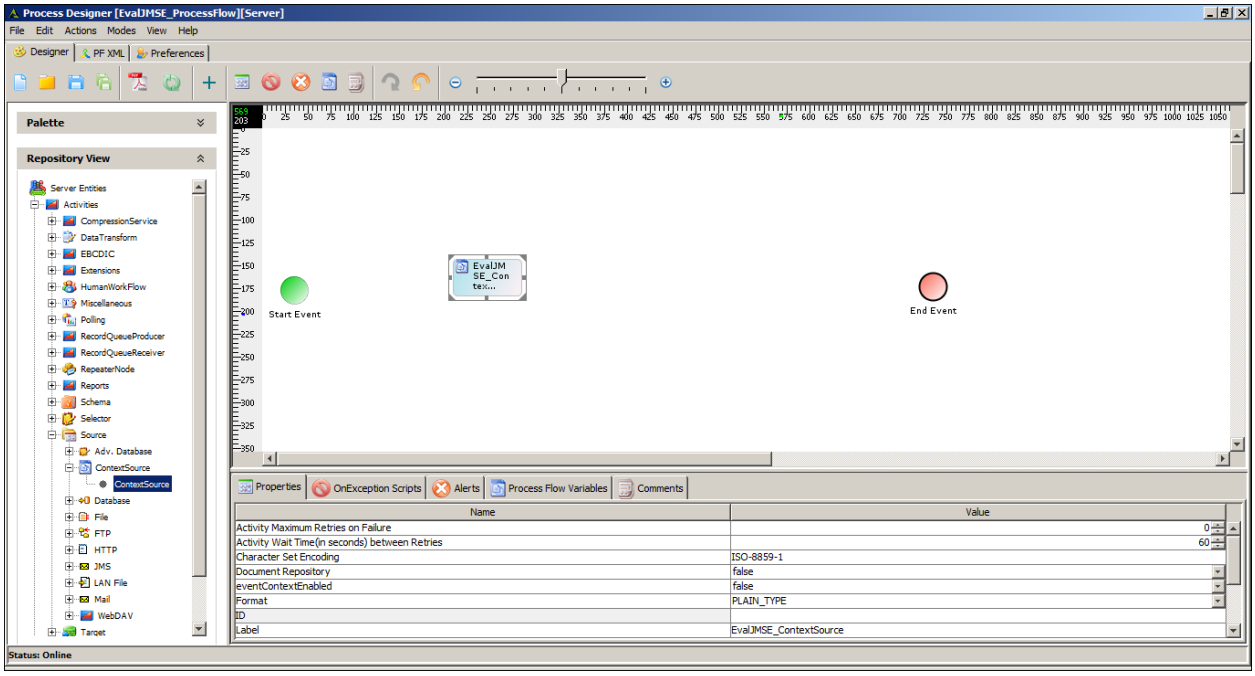

Figure 187: View Properties of Context Source Activity

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 188).

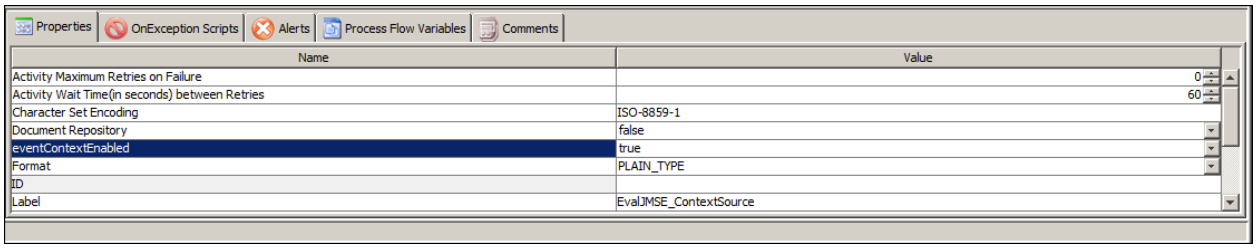

Figure 188: Context Source Name

15. Click **[+] Activities** in the Activities Panel to expand the list of services and then click **[+] Schema**. All 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 189).

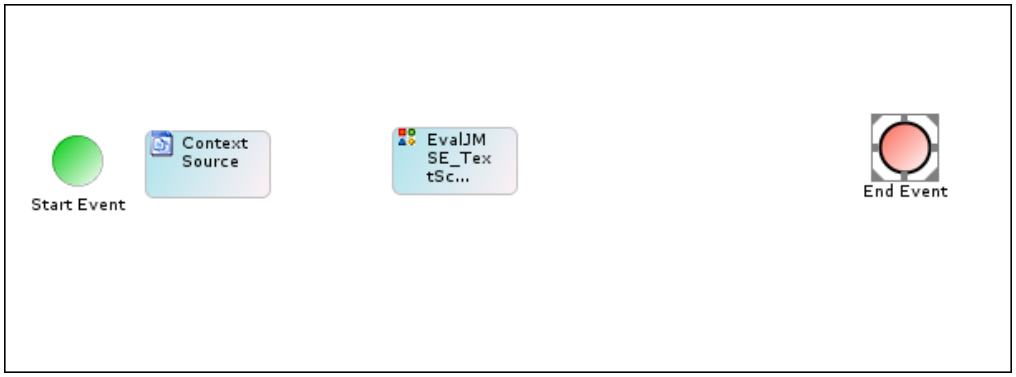

Figure 189: Drag Text Schema to Graph Canvas Area

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 190).

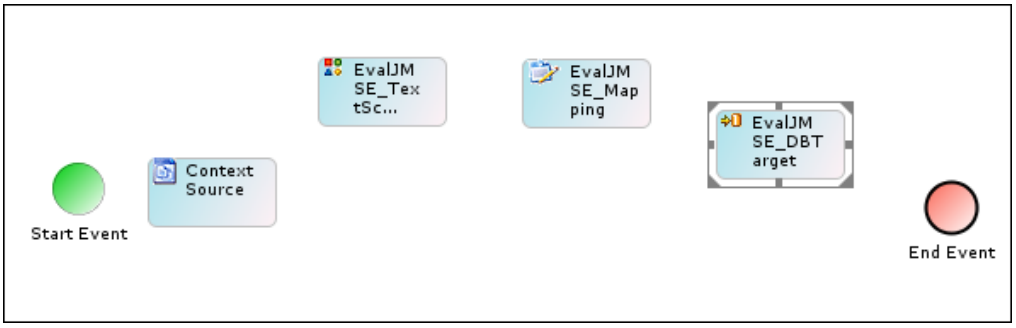

Figure 190: Drag Database Target to Graph Canvas Area

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 191).

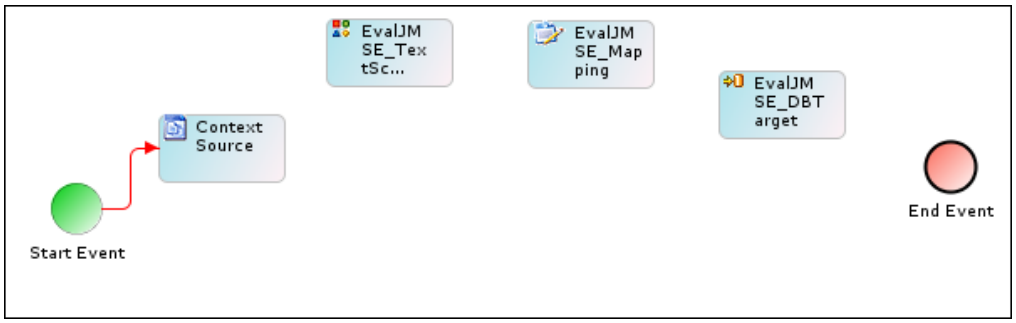

Figure 191: Connect Start Event to Context Source

23. Similarly, connect all other activities as shown in Figure 192.

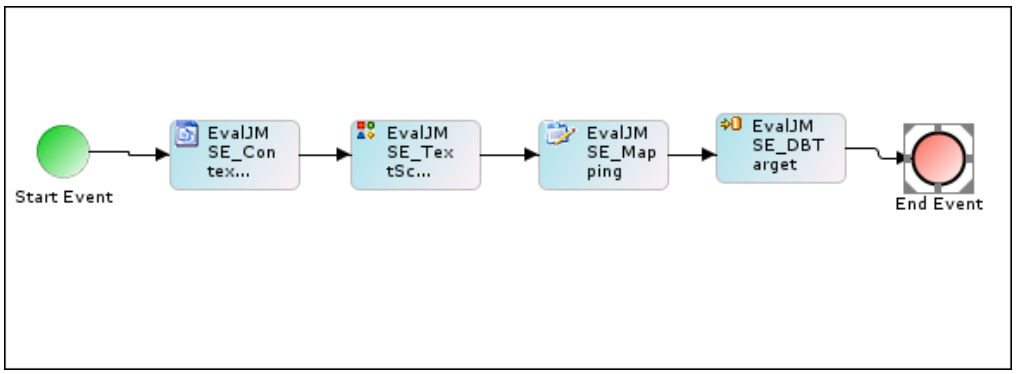

Figure 192: Connect all Activities

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 193).

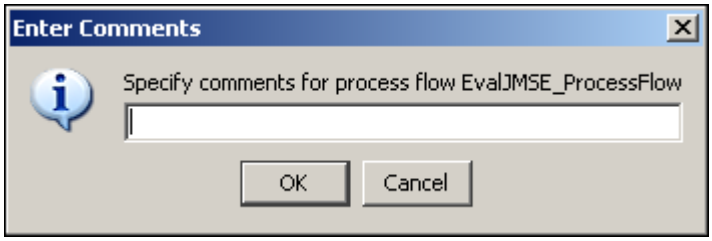

Figure 193: Enter Comments (Process Flow)

25. Enter the comments in the Specify Comments for process flow customer field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Events > Event Registry**. This displays the *Manage Event Registry* page with the list of existing Event Registry activities (go to Figure 166).
3. Select the required Event Registry activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective Event Registry activity in edit mode (see Figure 194).

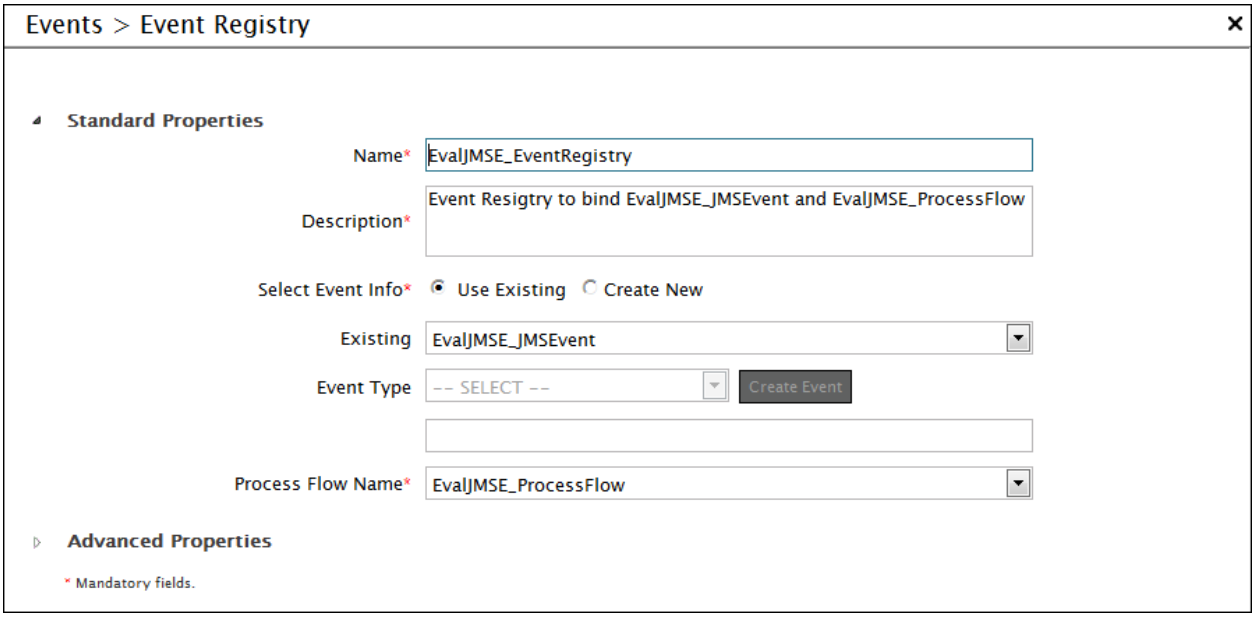

Figure 194: Edit *EvalJMSE_Event Registry*

The following table defines the fields on this screen.

Table 41: Details of Fields on Edit Event Registry Screen

Retention Option	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

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 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 175).
7. Enter comments in the **Add Comments** field.
8. Click **OK** to save the comments. This displays a screen confirming that the Event Registry has been updated successfully.

9

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:

EBIM Suite	BPM Suite	ESB 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:

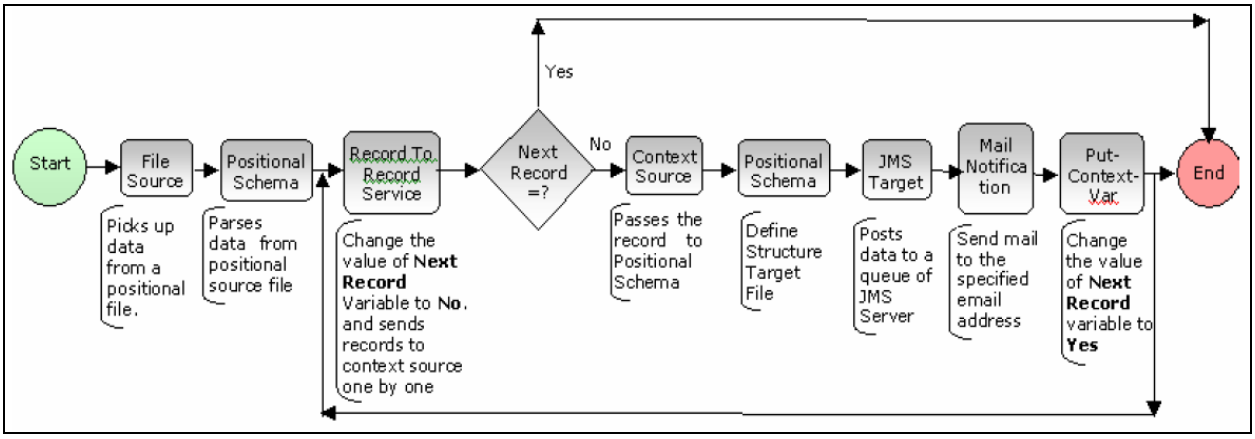

Figure 195: Flow Chart to show Process Flow

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 42: 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

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. On the Adeptia Suite homepage, click the **Develop** tab.

2. Go to Process > Process Flow. The *Manage Process Flow* screen is displayed (refer Figure 2)

3. Click the *Execute* button, which is adjacent to *EvalXform_ProcessFlow.* A confirmation dialog box is displayed (see Figure 3).

4. Click the button **Go to Logs** to view the status of the Process Flow execution. The Process Flow Logs are displayed (see Figure 196).

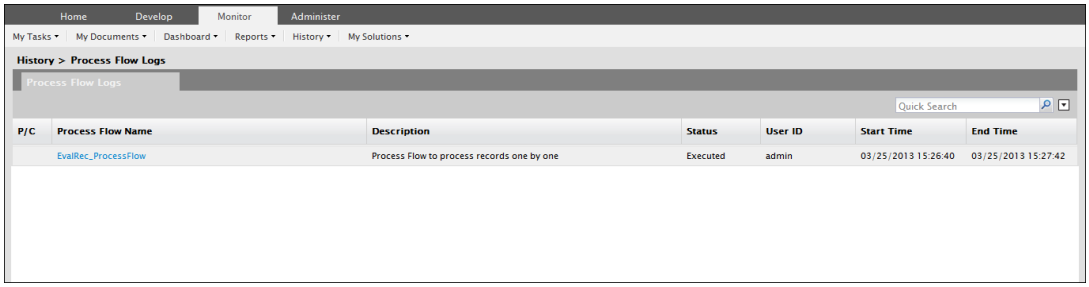

Figure 196: View Process Flow Logs

5. To view the execution details of the process flow, click the name of the process flow. The process flow execution details are displayed in a new tab (see Figure 197).

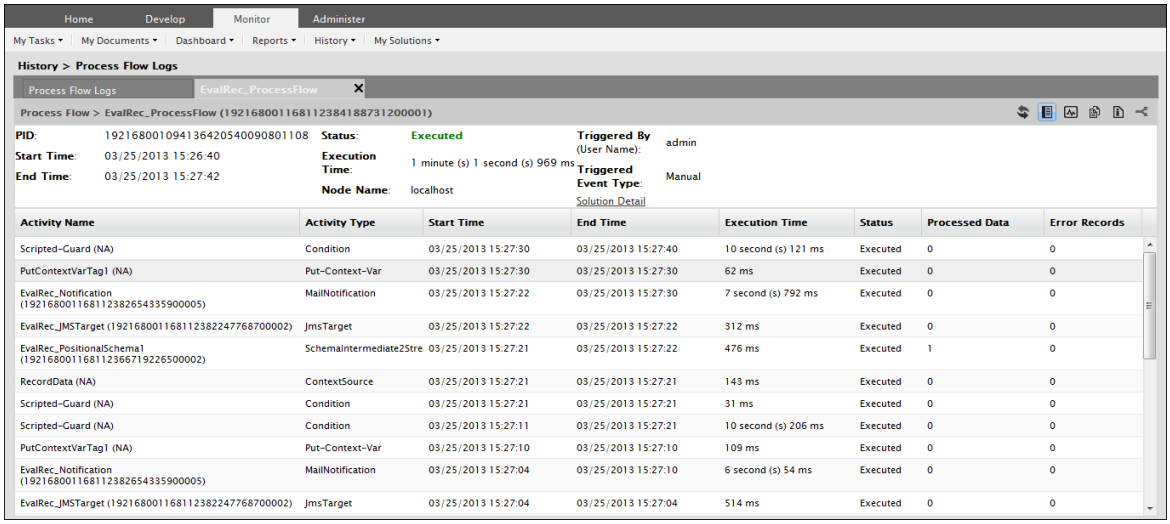

Figure 197: View Process Flow Log Details

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. On the Adeptia Suite homepage, click the **Develop** tab.

2. Go to **Services > Source** and then click **File**.

   The *Manage File Source* screen is displayed with the list of existing File source activities (refer to Figure 65).

3. Select the required File source activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective File source activity in edit mode (see Figure 198).

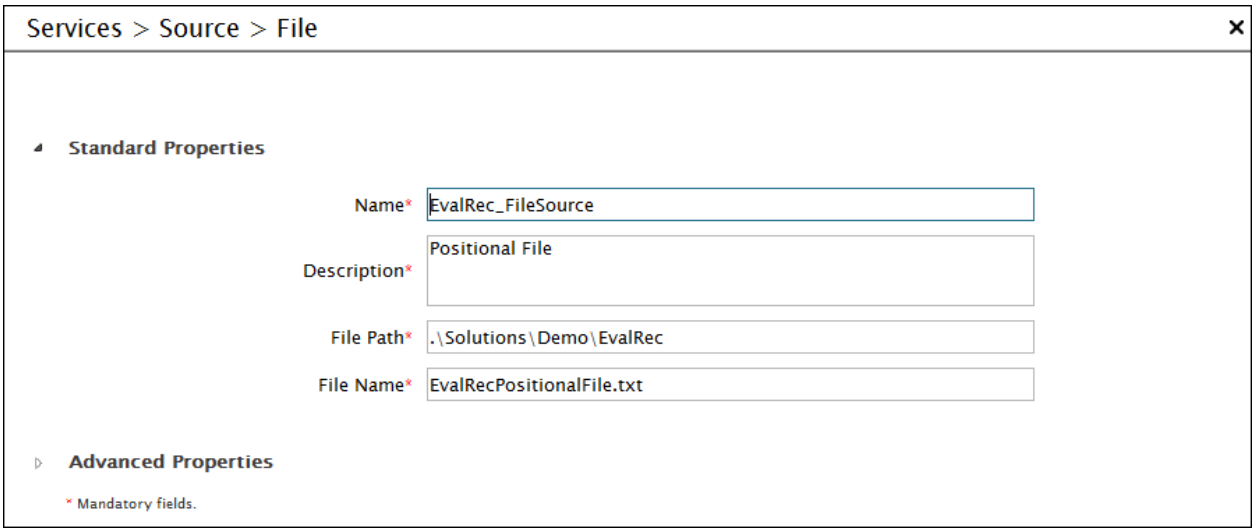

Figure 198: Edit *EvalRec_File Source* Activity

The following table defines the fields on this screen.

Table 43: Details of Fields on Edit File Source Screen

Field Name	Description
Name	Name of the File Source
Description	Description of the File Source
File Path	Path of the source file. For example: **`../../Solutions/Demo/EvalRec/`**
File Name	Name of the source file. For example: **`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 175).

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

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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Click **Services** to expand the tree. All 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 199).

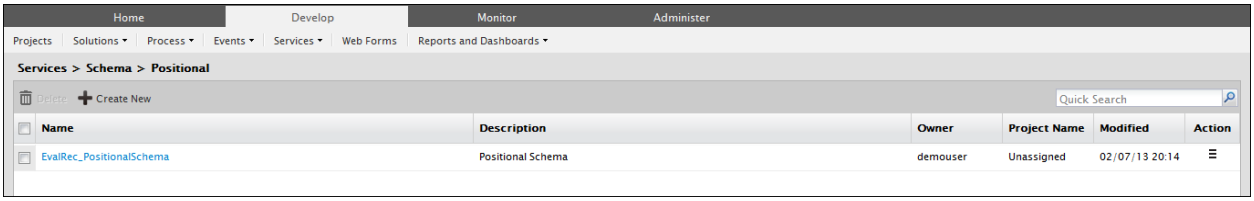

Figure 199: Manage Positional Schema

4. Select the required Schema activity and right-click the same to view **More Actions** menu.

This opens the respective Schema activity in edit mode (see Figure 200).

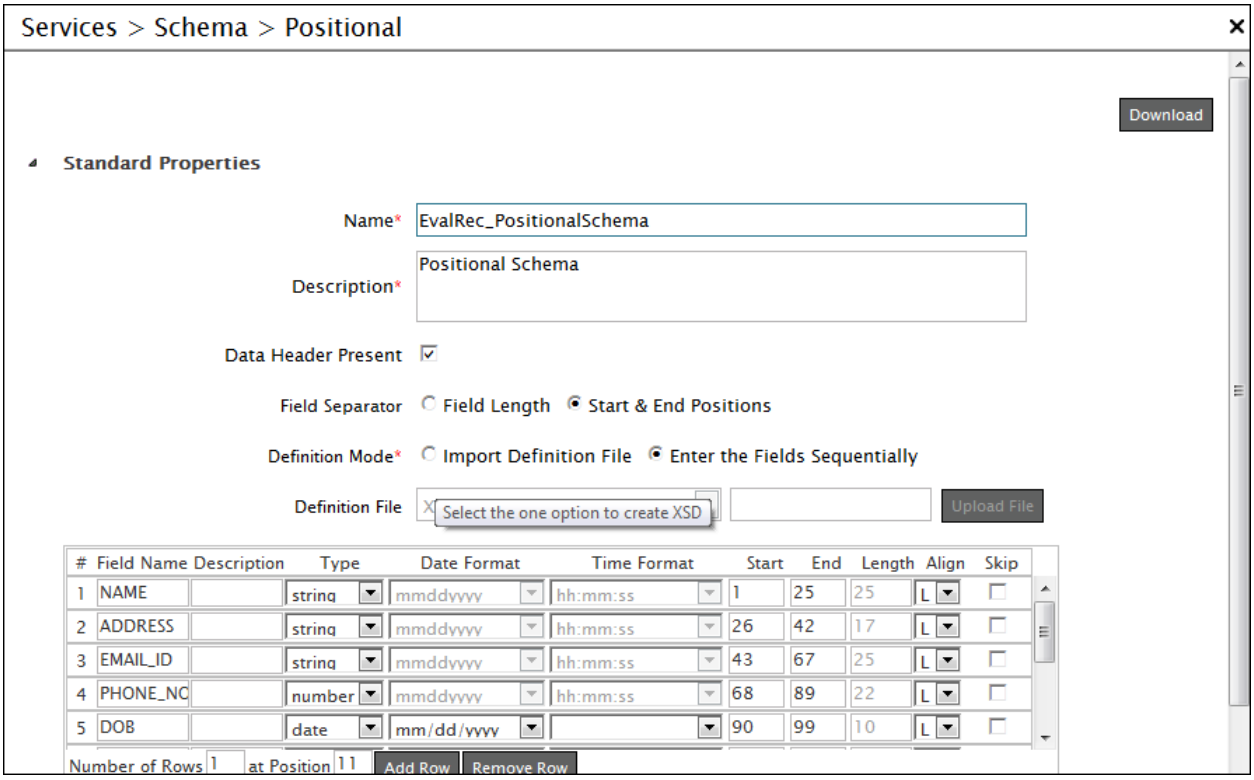

Figure 200: Edit *EvalRec_PositionalSchema* Activity

The following table defines the fields on this screen.

Table 44: Details of Fields on Edit Positional Schema Screen

Field Name	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: Use Definition File Enter the Field Sequentially Pre-created schema with this sample Process Flow is created using second option i.e. Enter the Field Sequentially

Field Name	Description
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
Start Position	Start position of the field
End Position	End position of the field
Length	Length of the field
Alignment	Alignment of the Filed **L** if the field is left aligned. **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 175).
7. Enter comments in the **Add Comments** field.
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 178).
2. Select the type of schema to test, from the *Type* dropdown 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* dropdown 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 201.

```
// 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 201: Sample JAVA Code

Steps to edit the Record to Record Service

1. In Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Services > Data Transformation** 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 202).

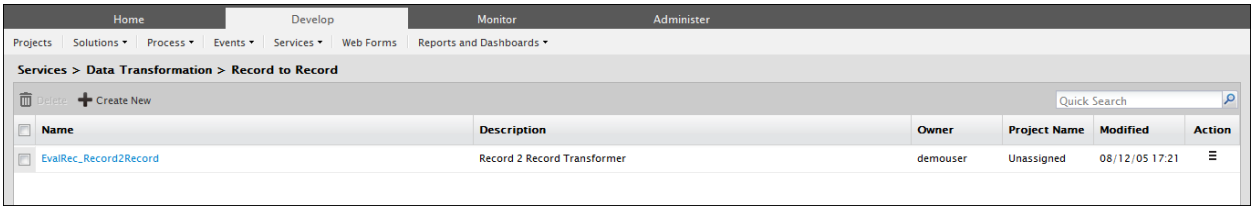

Figure 202: Manage Record to Record

3. Select the required activity and right-click the same to view the **More Actions** menu.
4. Click **Edit** to open the respective activity in edit mode (see Figure 203).

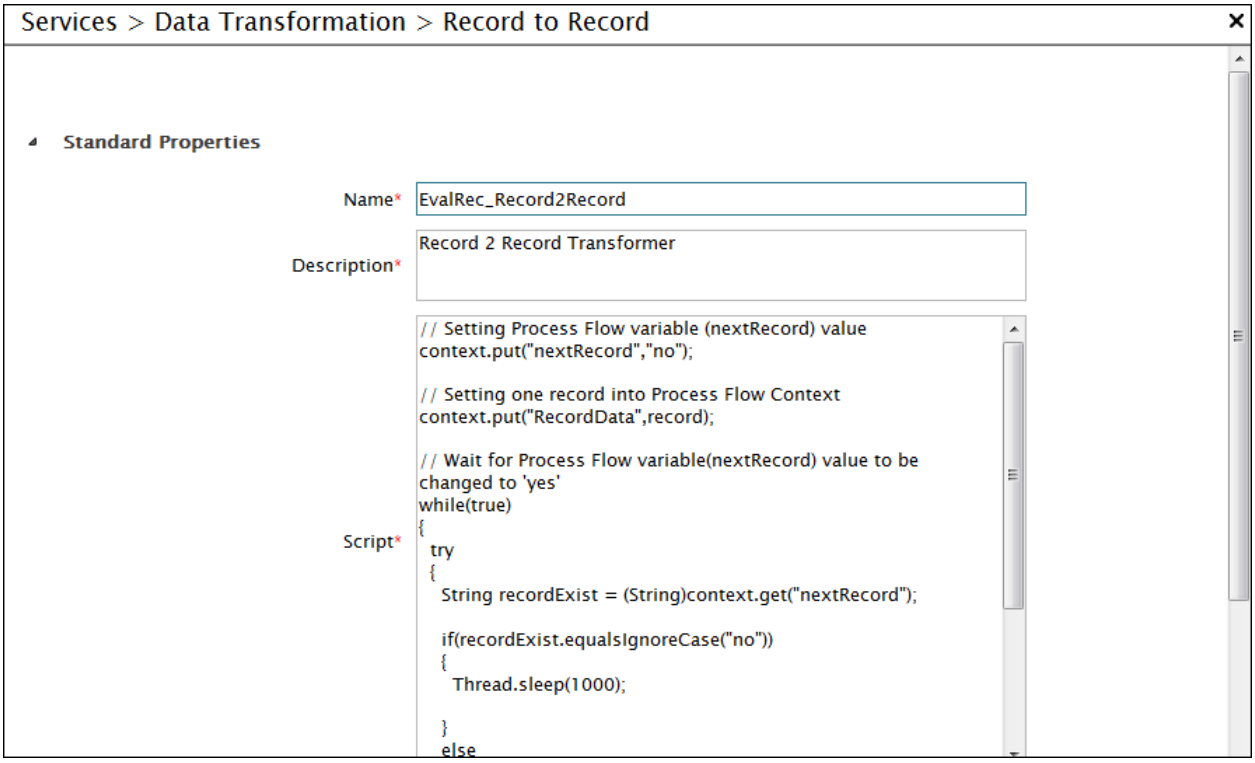

Figure 203: Edit *EvalRec_Record2Record* Activity

The following table defines the fields on this screen.

Table 45: Details of Fields on Edit Record to Record Screen

Field Name	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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite Home page, click the **Develop** tab.
2. Go to **Services > Connector**. All items in the **Connector** category are displayed.
3. Click **JMS Provider**. The *Manage JMS Provider* screen is displayed with the list of existing JMS Provider activities (refer to Figure 173).

4. Select the required JMS activity and right-click the same to view the **More Actions** menu.
5. Click **Edit** to open the respective JMS Provider activity in edit mode (see Figure 204).

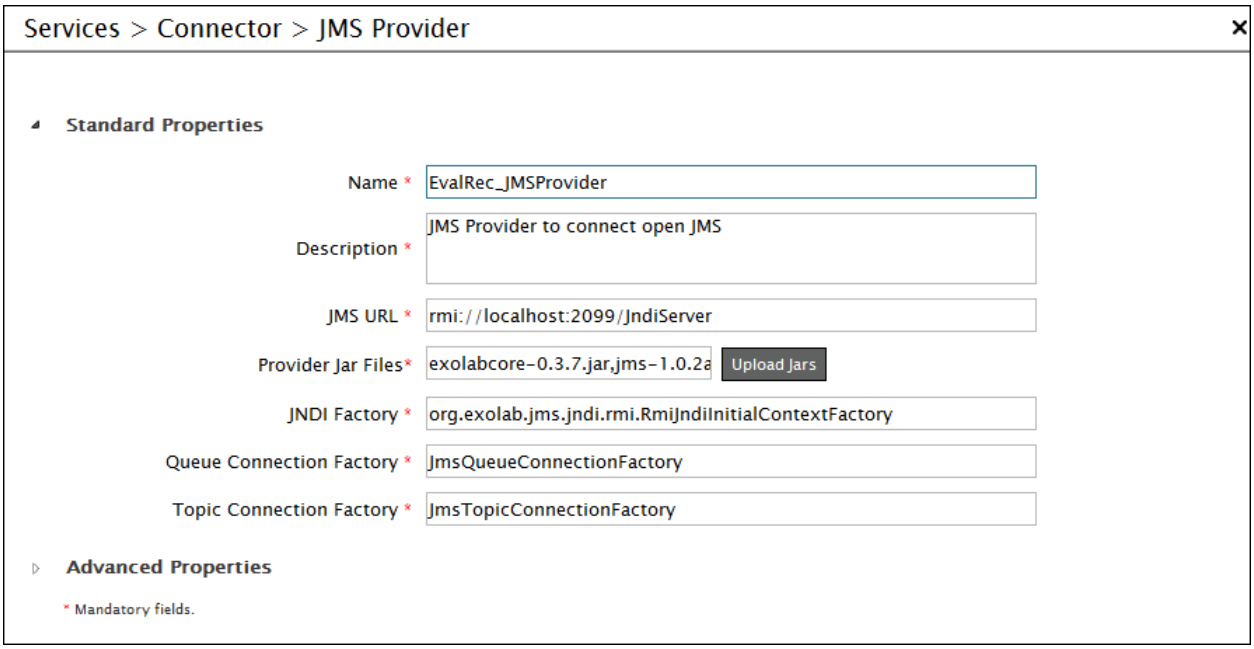

Figure 204: Edit *EvalRec_JMS Provider* Activity

The following table defines the fields on this screen.

Table 46: Details of Fields on Edit JMS Provider Screen

Field Name	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.

6. Make the necessary changes.

7. 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 175).
8. Enter comments in the **Add Comments** field.
9. 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. On the Adeptia Suite homepage, click the **Develop** tab.
2. Go to **Services > Target** and then click **File**.

   The *Manage File Target* screen is displayed with the list of existing File Target activities (refer to Figure 22).

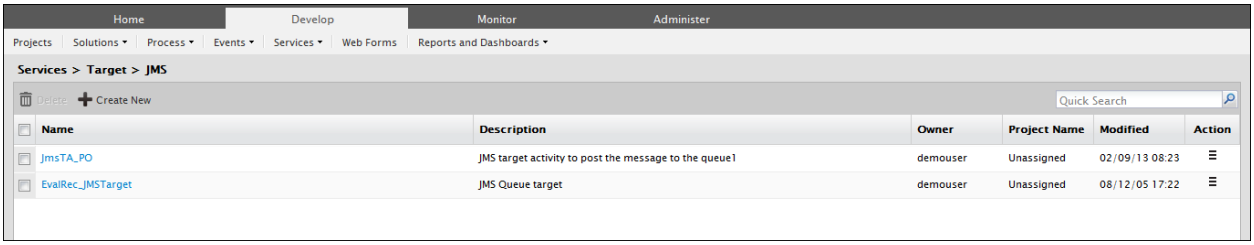

Figure 205: Manage JMS Target

3. Select the required File target activity and right-click the same to view the **More Actions** menu.

4. Click **Edit** to open the respective File target activity in edit mode.(see Figure 205).

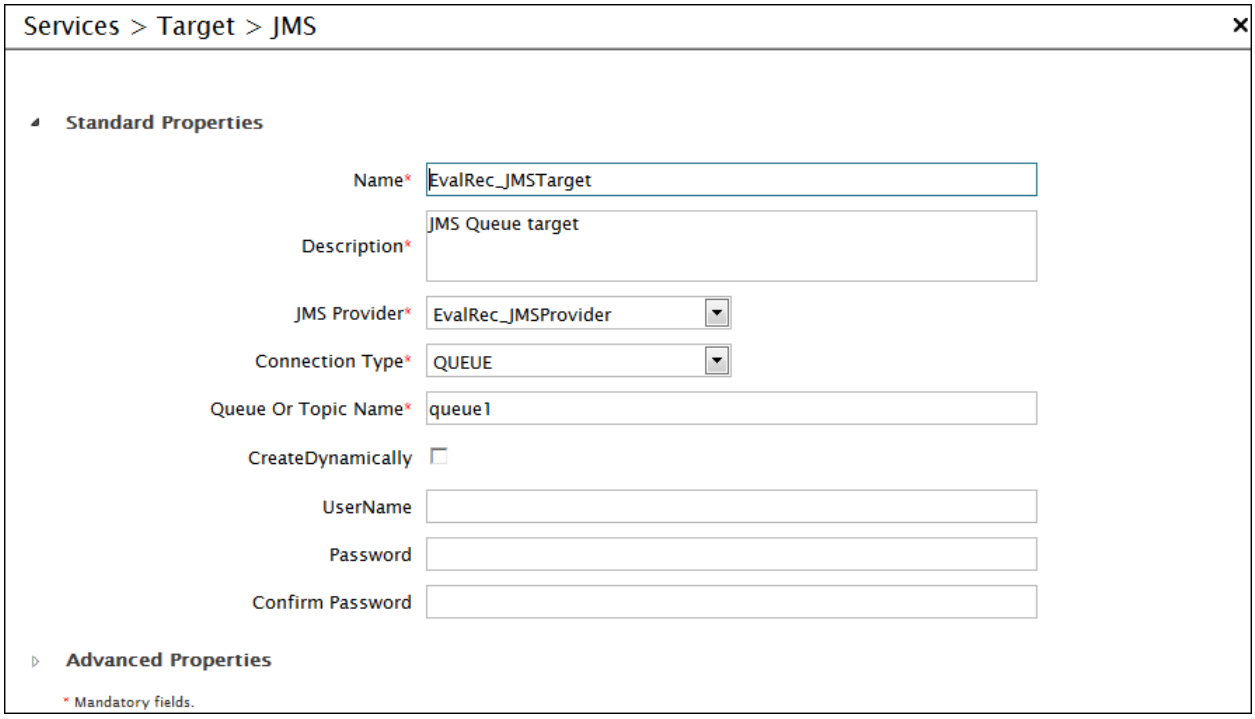

Figure 206: Edit *EvalRec_JMSTarget* Activity

The following table defines the fields on this screen.

Table 47: Details of Fields on Edit JMS Target Screen

Field Name	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 TOPIC: Used for one to many messaging. It supports publish subscribe model of messaging. 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

Field Name	Description
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 175).
7. Enter 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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Services > Miscellaneous**.

The *Manage Mail Notification* screen is displayed with a list of existing Mail Notification activities (see Figure 207).

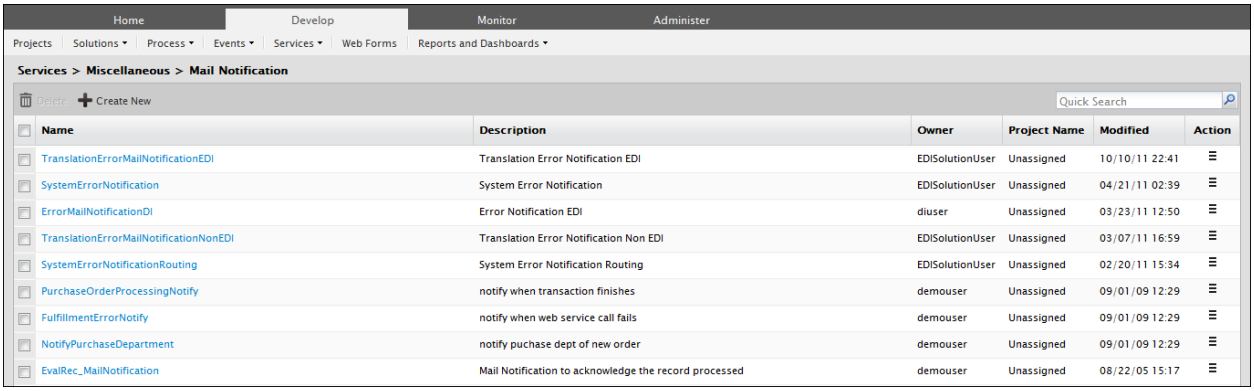

Figure 207: Manage Mail Notification

3. Select the required Mail Notification activity to view the **More Actions** menu.

4. Click **Edit** to open the respective activity in edit mode (see Figure 208).

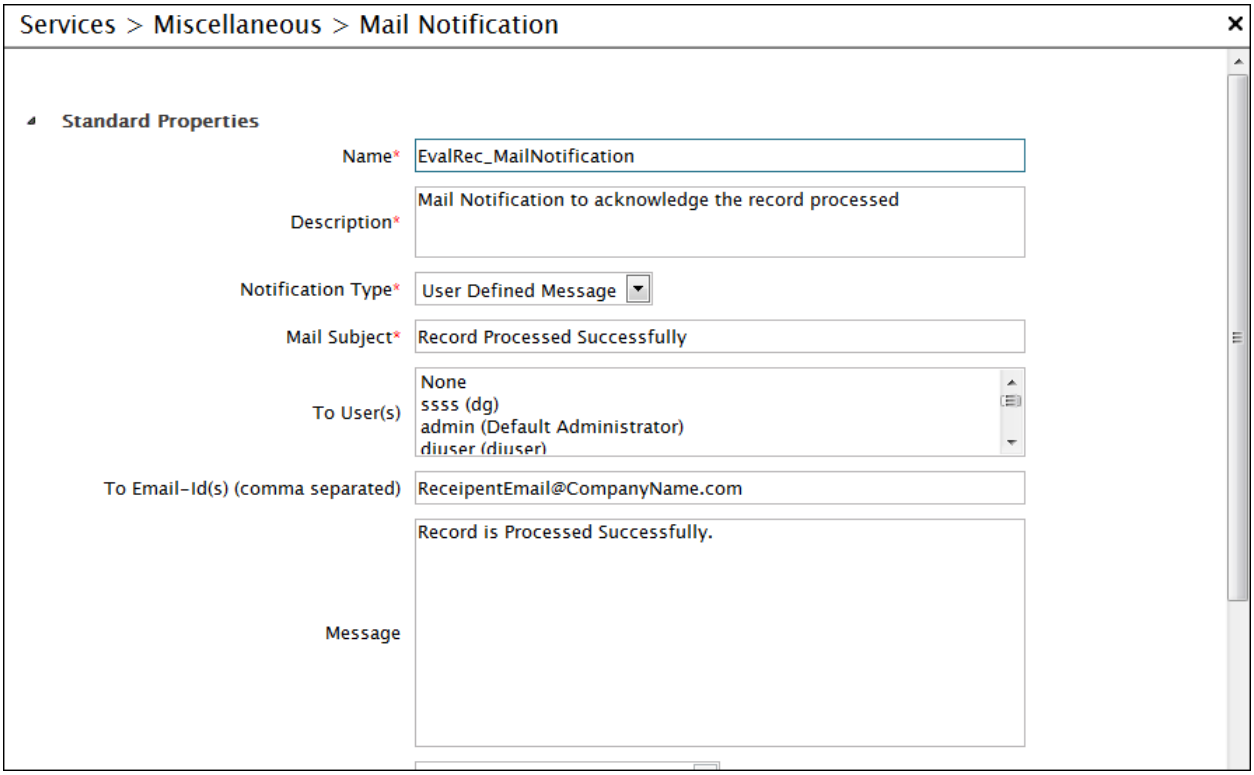

Figure 208: Edit *EvalRec_MailNotification* Activity

The following table defines the fields on this screen.

Table 48: Details of Fields on Edit Mail Notification Screen

Field Name	Description
Name	Name of the Mail Notification
Description	Description of the Mail Notification
Notification Type	There are two types of Mail Notification: • User Defined • Process Flow Summary 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: • Running or Executed Successfully • Failure • Always 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 175).
7. Enter comments in the **Add Comments** field.
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. On the Adeptia Suite homepage menu, click the **Develop** tab.
2. Go to **Process > Process Flow**. The *Manage Process Flow* screen is displayed with the list of existing Process Flows (refer to Figure 55).
3. Click the **Create New** link. The *Create Process Flow* screen is displayed (refer to Figure 56).
4. Enter the name and description of the Process Flow.
5. Select the logging level from the *Logging Level* dropdown list.
6. Select repository file retention from the **Repository File Retention** option.
7. Click the **Process Designer** button to open Process Designer.

For more information on how to create process flows and on *Process Designer* screen, refer to the Creating Process Flow section.

8. Click **OK** to synchronize.
9. Click **[+] Activities** in Repository View to expand the list of services and then click **[+] Source**. All 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 209).

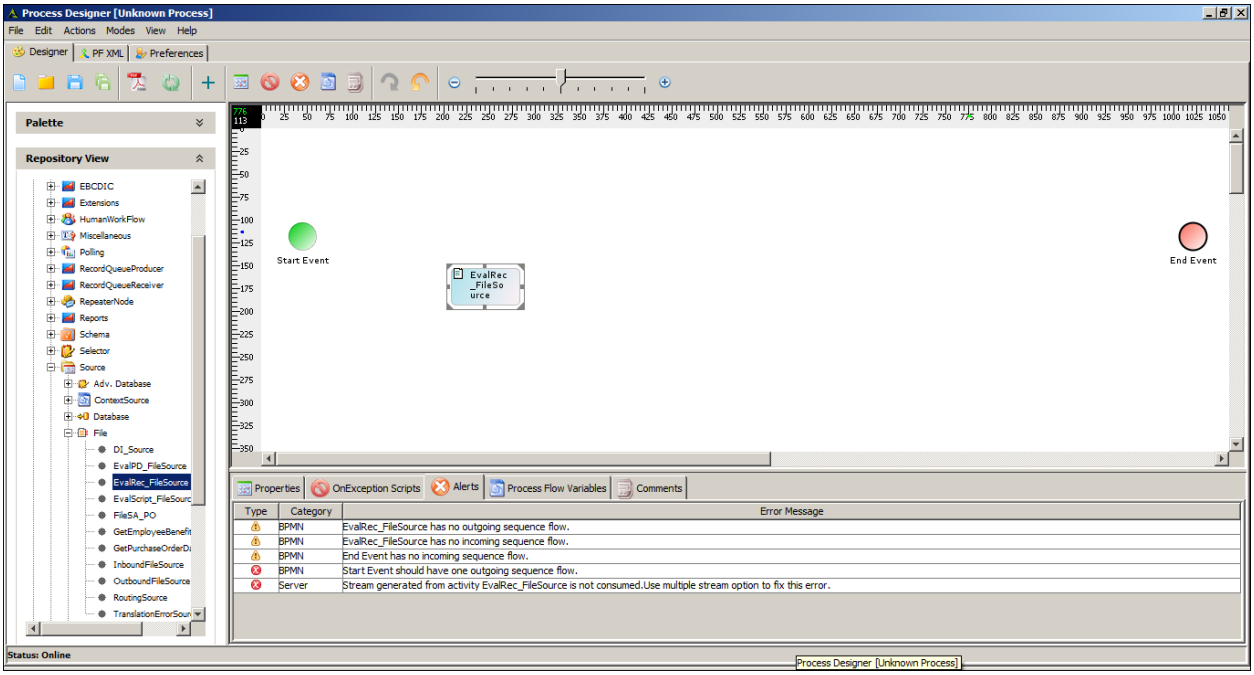

Figure 209: Select File Source Activity

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 210). In this Process Flow, Gateway is used to check the value of *NextRecord* variable and to decide which path to choose.

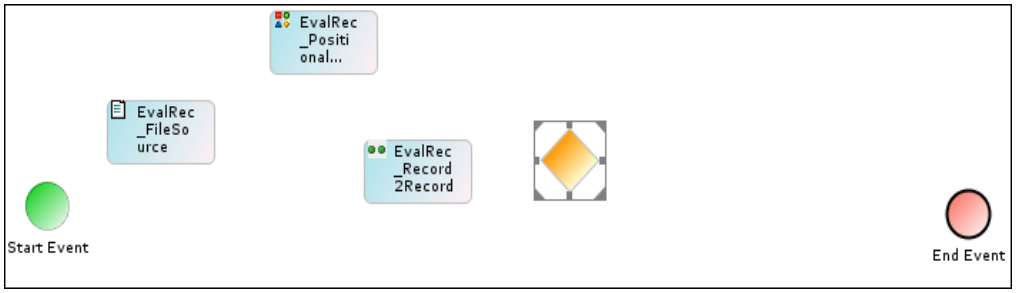

Figure 210: Drag BPMN Gateway to Graph Canvas Area

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 211)

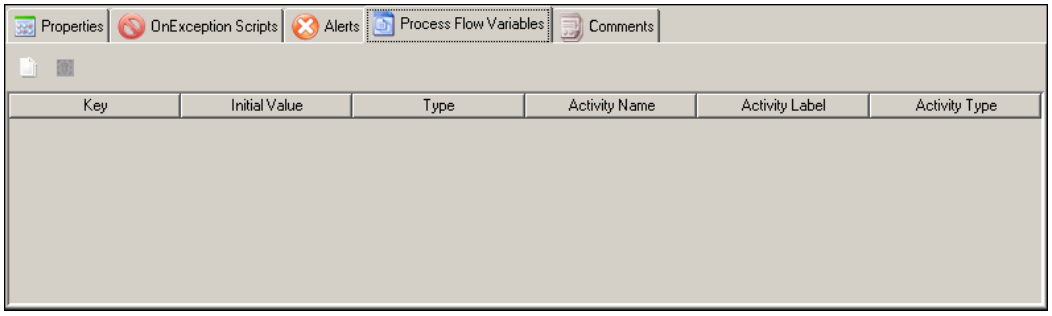

Figure 211: Process Flow Variables Panel

16. Click **New Process Flow Variable** ( ) button. The Process flow variable entry dialog box is displayed (see Figure 212).

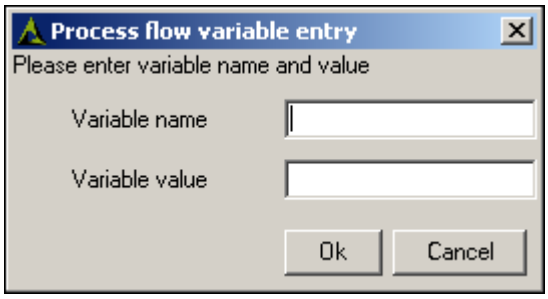

Figure 212: Create Process Flow Variable

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 213).

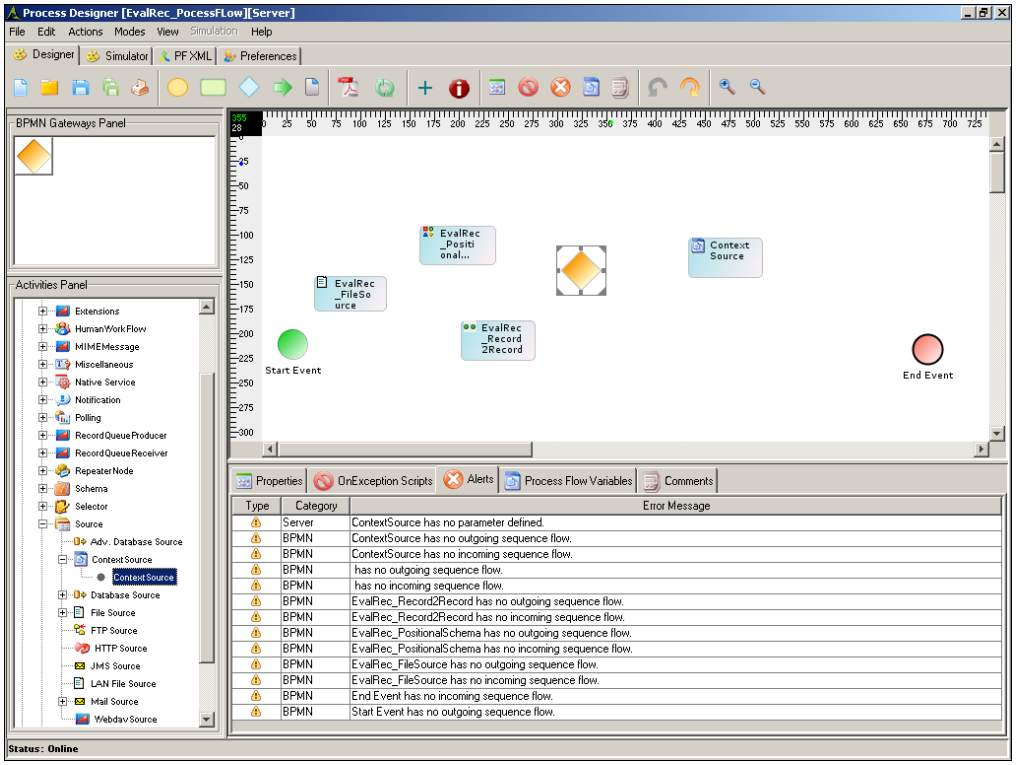

Figure 213: Drag Context Source to Graph Canvas Area

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 214).

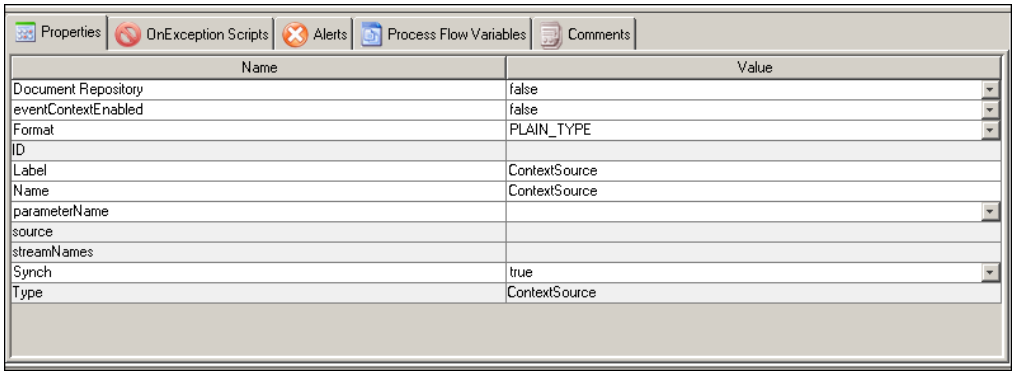

Figure 214: Drag Context Source to Graph Canvas Area

23. Change the Name and Label of the context source as *RecordData.*

24. Select *RecordData* from the *parameterName* dropdown list (see Figure 215).

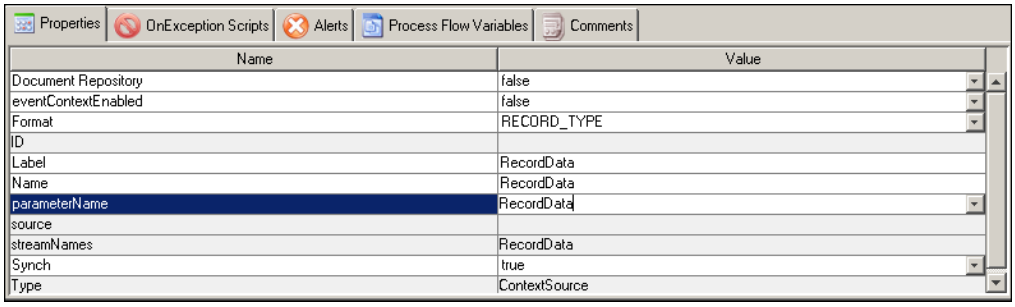

Figure 215: Enter Context Source Name

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 216).

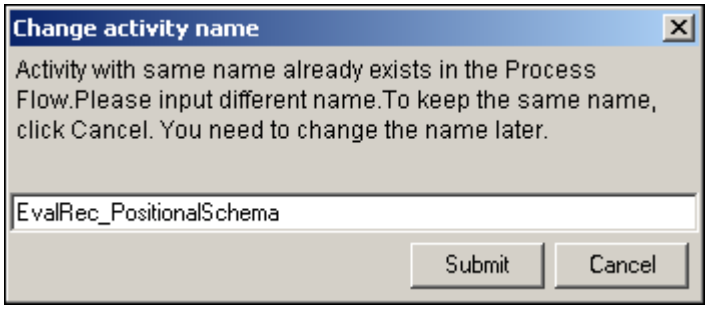

Figure 216: Change Activity Name

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 217).

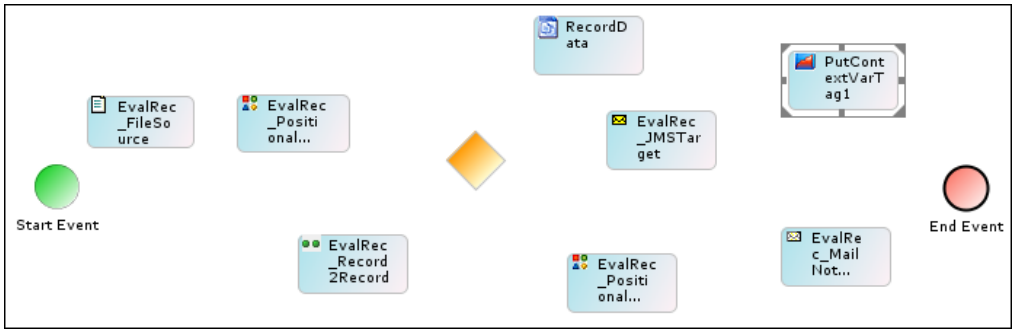

Figure 217: Select Put-Context -Var

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 218).

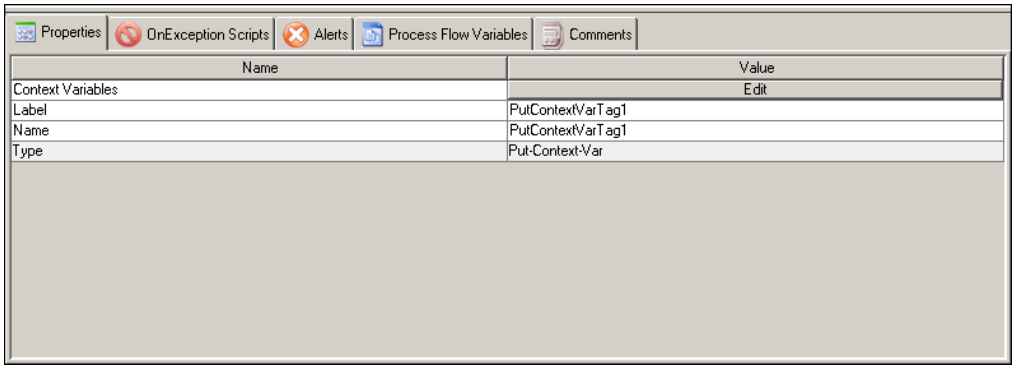

Figure 218: View Properties of Put-Context-Var

32. Click **Edit** from the value field of the *context Variable* properties. The Edit Context Variable screen is displayed (see Figure 219).

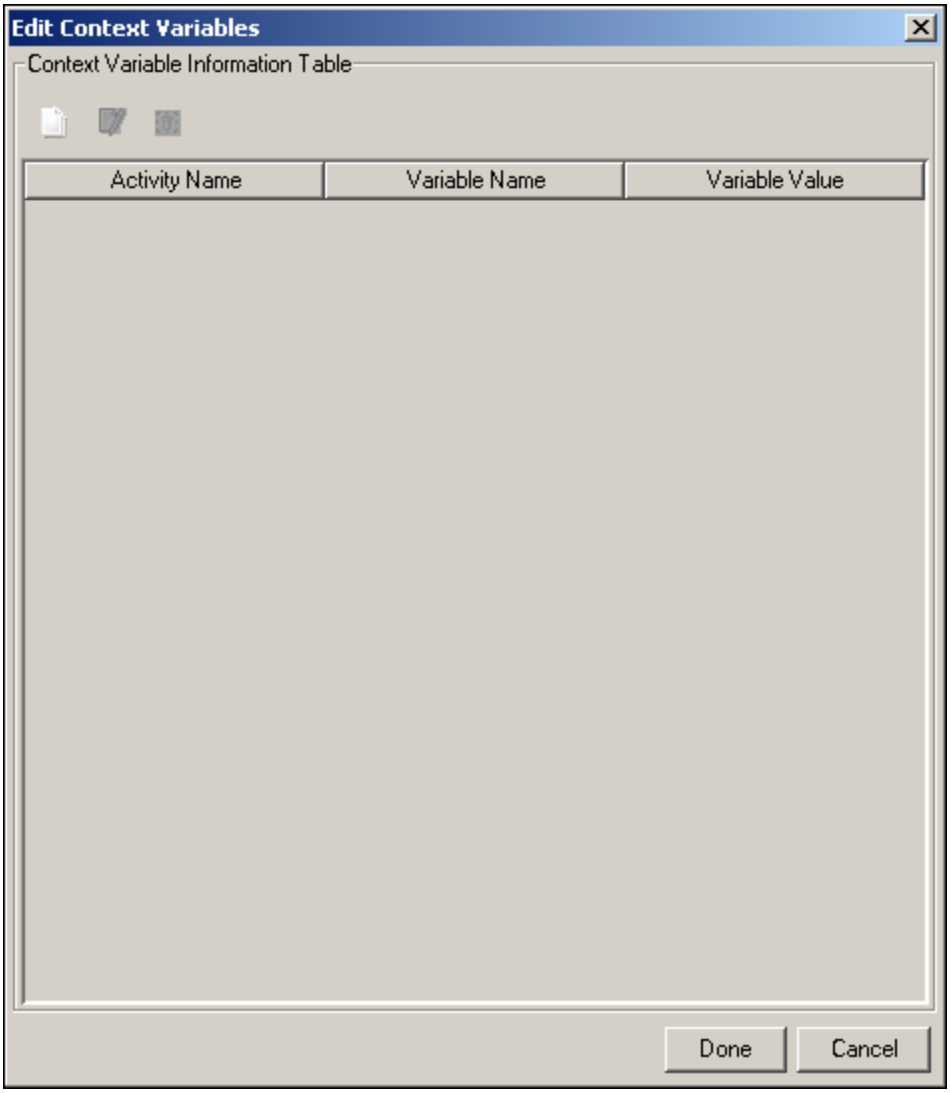

Figure 219: Edit Context variable

33. Click **New variable** ( ) button. The Context Variable Information dialog box is displayed (see Figure 220).

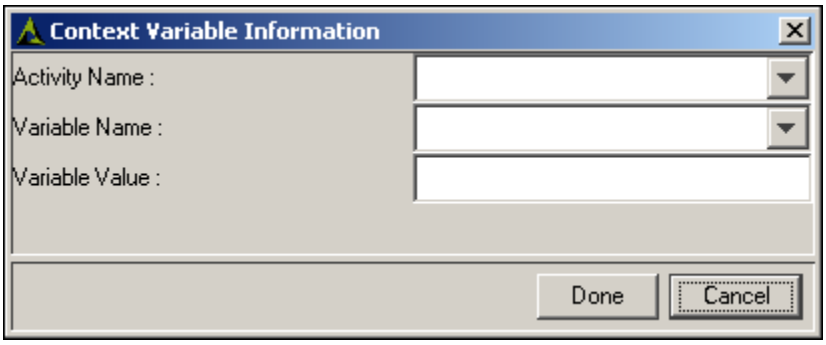

Figure 220: Add Context variable

34. Select *nextRecord* from the *Variable Name* dropdown 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 apiepropriate 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 221).

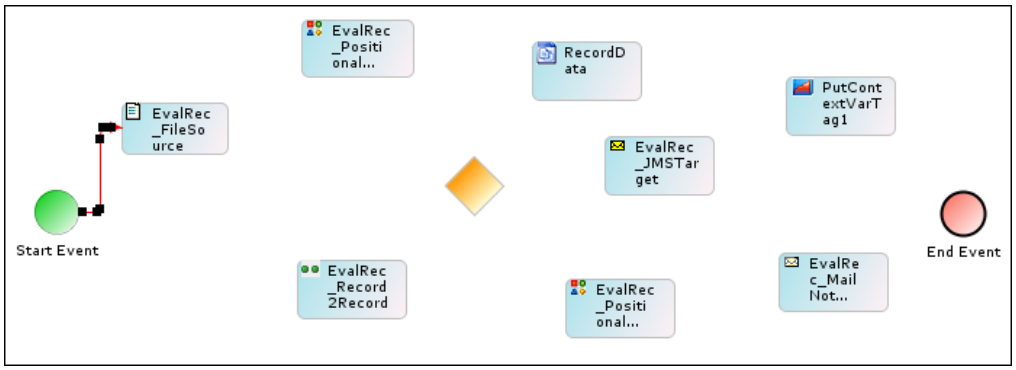

Figure 221: Connect Start Event to File Source Activity

40. Similarly, connect all other activities as shown in Figure 222.

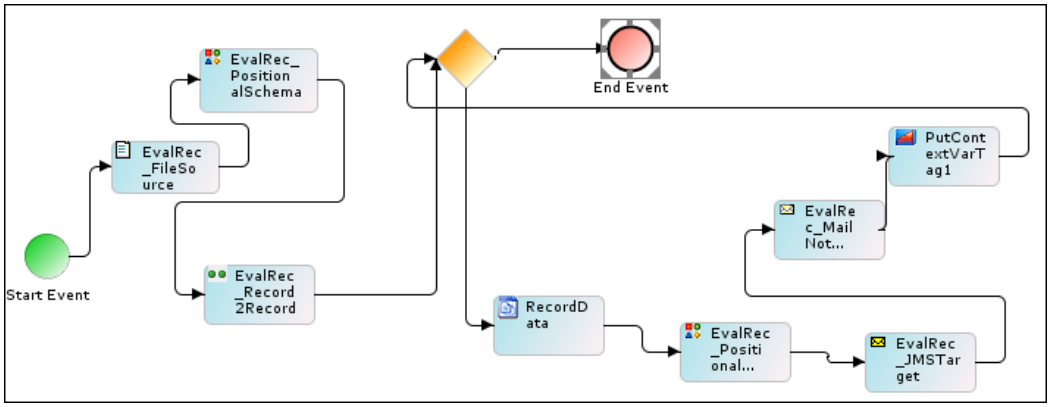

Figure 222: Connect all Activities

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 223).

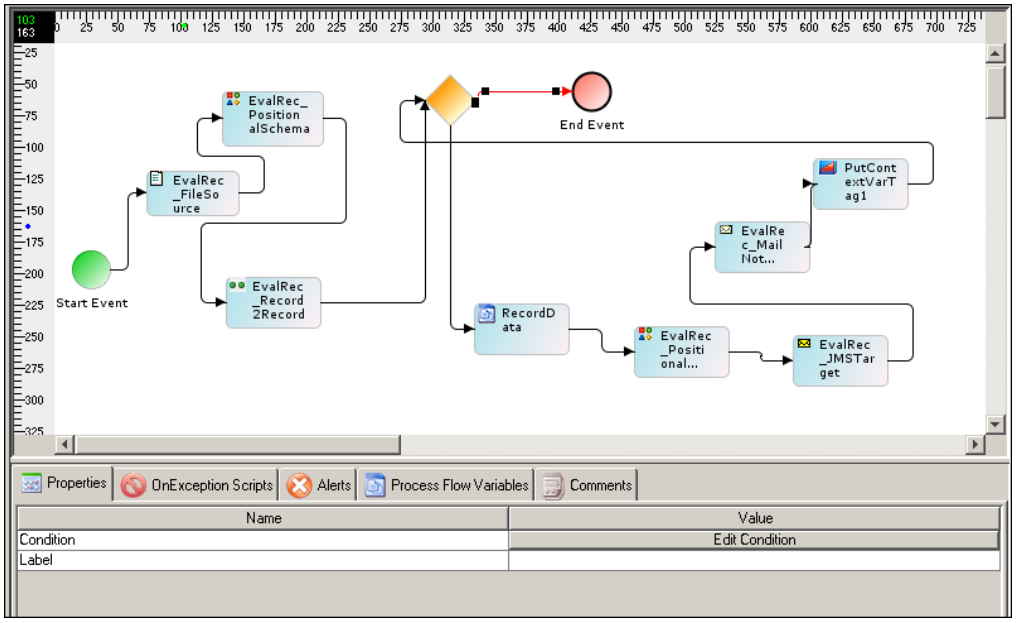

Figure 223: View Properties of Control Flow between Gateway Element and End Event

42. Click **Edit Condition**. The Condition Wizard is displayed (see Figure 224).

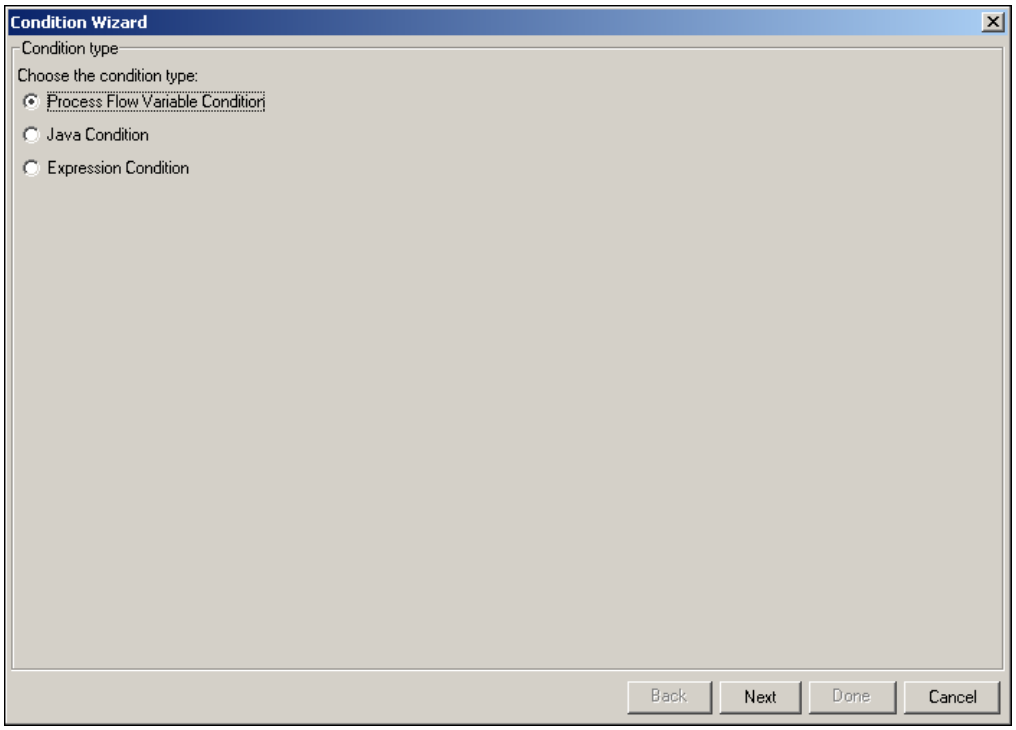

Figure 224: Condition Wizard

43. Select **Java Condition** and then click the **Next** button. The Java Condition box is displayed (see Figure 225).

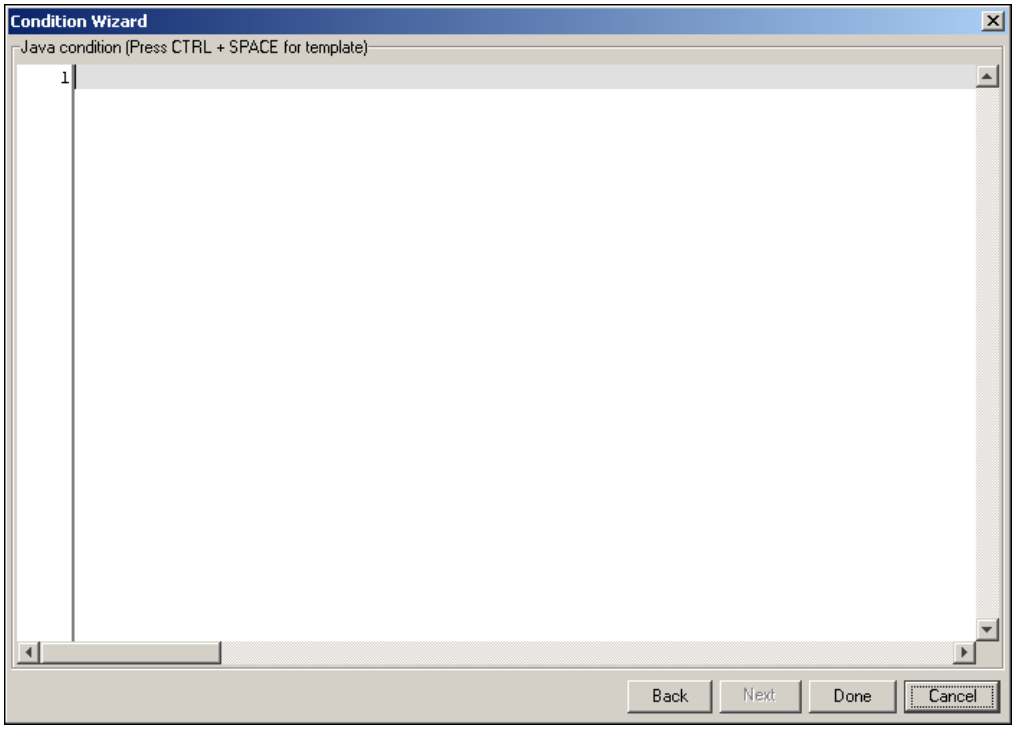

Figure 225: Enter JAVA Condition

44. Enter the sample Java Code in the Java Condition box (see Figure 226) 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 226: 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 227).

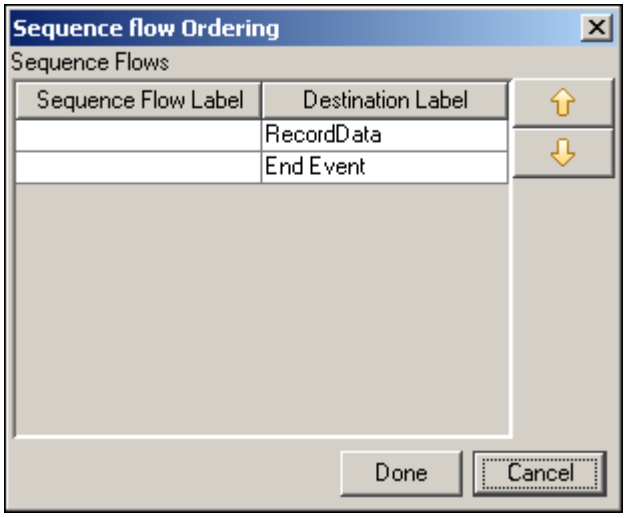

Figure 227: Sequence Flow Ordering

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 228).

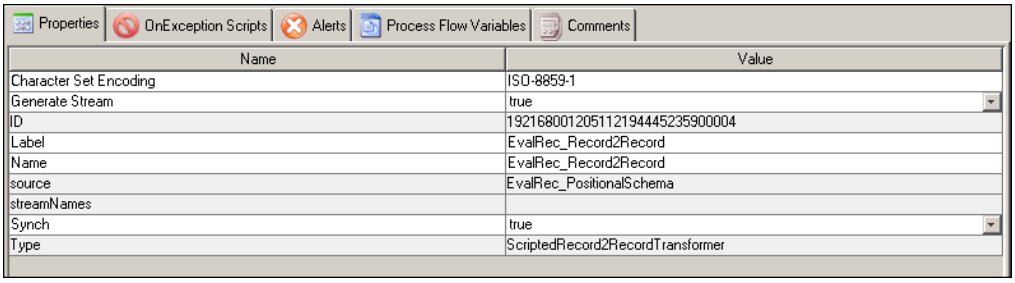

Figure 228: View Properties of *EvalRec_Record2Record* Activity

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 193).

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

The comment should be at least 1 character in length.

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

10

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 229) 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 229: 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 230).

```
<!-- 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 230: Content copied in *openjms.xml* file

6. Save the file and close it.
7. Set following parameter as System environment variable

   JAVA_HOME= "<Path of JRE home>"

   OPENJMS_HOME="Path where JMS Server is installed"

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`

   OpenJMS server is started (see Figure 231).

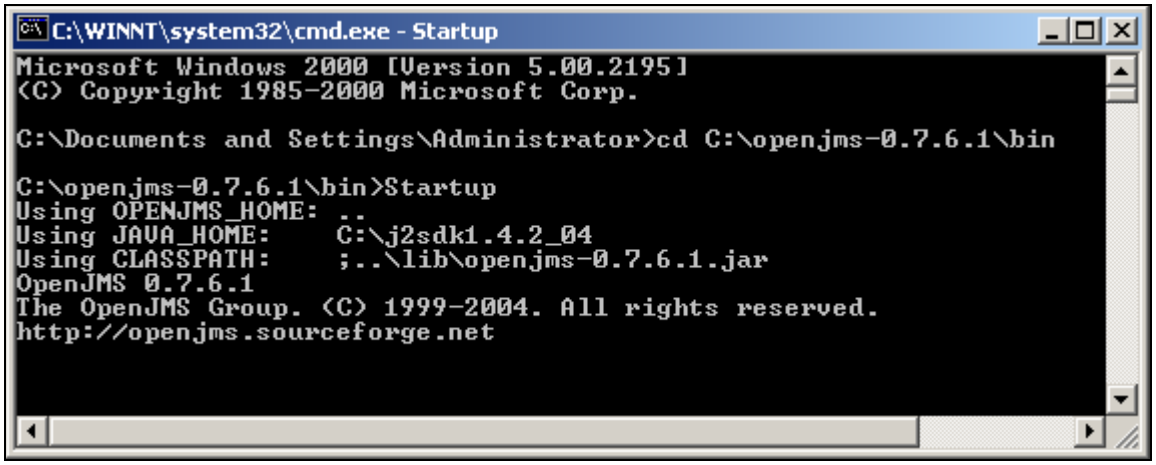

Figure 231: Start OpenJMS Server

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
   ```

   OpenJMS admin screen is displayed (see Figure 232).

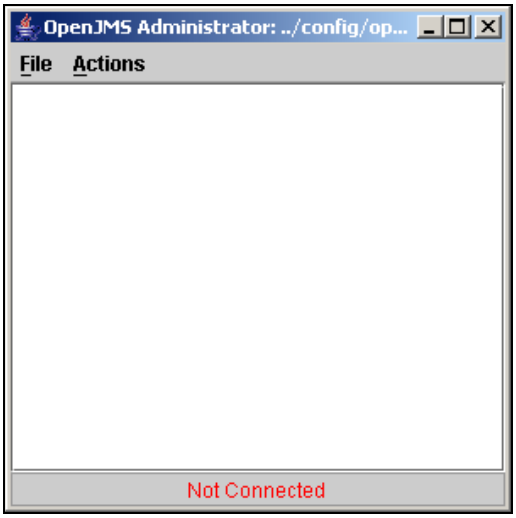

Figure 232: Start OpneJMS Admin GUI

3. Click **Actions** menu, go to **Connections** and then select **Online**. Default Queues and topics are displayed (see Figure 233).

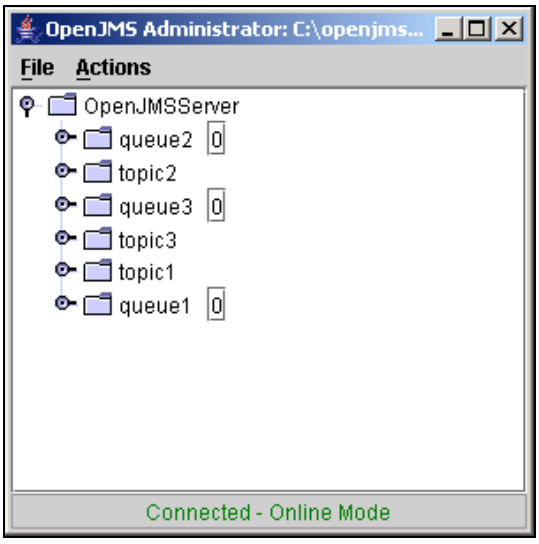

Figure 233: Connect to OpenJMS Server

4. To close the Admin screen, click **File** menu and select **Exit**

.

TABLE OF FIGURES

INDEX