ADEPTIA

Adeptia Suite 5.0

Developer Guide

Release Date September 15, 2009

Adeptia Inc.
443 North Clark Ave,
Suite 350
Chicago, IL 60654, USA

Copyright

Trademarks

The Adeptia™ is a trademark of Adeptia, Inc.

Statement of Conditions

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

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

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

Document Conventions

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

Abbreviations Used

Abbreviation	Description
WebDAV	Web-based Distributed Authoring and Versioning

Contact Information

In case of any queries, please contact us at:

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

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

Table of Contents

1 ABOUT THIS GUIDE

This document 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.

A developer also has access to the other features of the Adeptia Suite. For details, refer to the Business User Guide, Process Modeling and Simulation Guide and Administrator Guide.

Pre-Requisite:

- It is important to read the *Getting Started* guide before reading this guide.

This document is divided into the following sections:

- Forms
  - **Designing Web Forms**
- Workflow
  - **Designing Workflow Task**
- Services
  - **Creating Data Dictionary**
  - **Creating Source Activity**
  - **Creating Schema Activity**
  - **Creating Target Activity**
  - **Creating Web Services Activity**
  - **Creating Data Transformation Activities**
  - **Creating Extensions**
  - **Creating Native Service Activity**
  - **Creating Notification Activity**
  - **Creating Polling Activity**
  - **Creating Miscellaneous Activities**
- Process Flow
  - **Working with a Process Flow**
- Events and Triggers
  - **Creating Calendar Event**
  - **Creating Database Event**

- **Creating Event Registry**
- **Creating File Event**
- **Creating FTP Event**
- **Creating HTTP Trigger**
- **Creating JMS Event**
- **Creating Mail Event**
- **Creating Timer Event**
- **Creating Web Service Trigger**

- Reports and Dashboards
  - **Creating Custom Report**
  - **Creating Dashboard Component**
  - **Creating Dashboard Design**

2 TARGET AUDIENCE

Even though all these features are available to all users, they are primarily performed by a developer. Thus, this document is intended for developers. They can use these details to seamlessly perform all design features of Adeptia Suite.

FORMS

3 DESIGNING WEB FORMS

Web forms can be used for two purposes in Adeptia Suite. It can be used in the Workflow task or as a form, which is added as a link in the Workspace Menu.

You can create web forms and trigger process flows or customize using the web forms. Adeptia Suite also allows you to integrate a custom application into the Adeptia Suite using its graphical user interface. You can upload any custom JSP or HTML file, written to serve a specific purpose. This feature facilitates smooth management of multiple JSPs and HTML files, saves time and allows seamless integration into the Adeptia Suite.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√		

This chapter describes the following tasks:

- Creating a Web Form
- Activating a Web Form

CREATING A WEB FORM

A web form can be created in three ways:

- Using Template
- Entering HTML Code manually
- Using Rich Form

Steps to create web form

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Forms** to expand the tree and then click **Web Forms**. The Manage Web Forms screen is displayed (see Figure 3.1).

This feature is visible to only those users who have *Write* and *Execute* permissions.

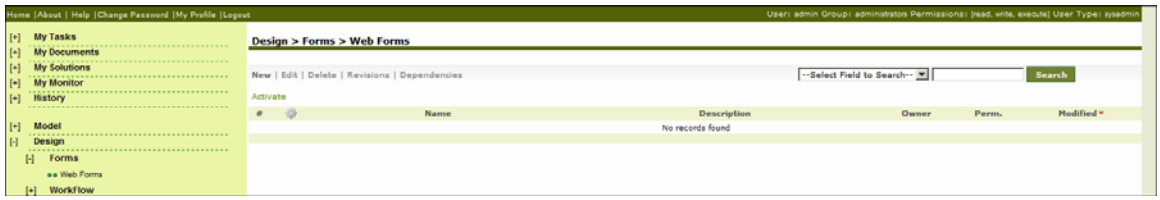


Figure 3.1: Manage Web Forms

3. Click the **New** link. The Create Web Forms screen is displayed (see Figure 3.2).

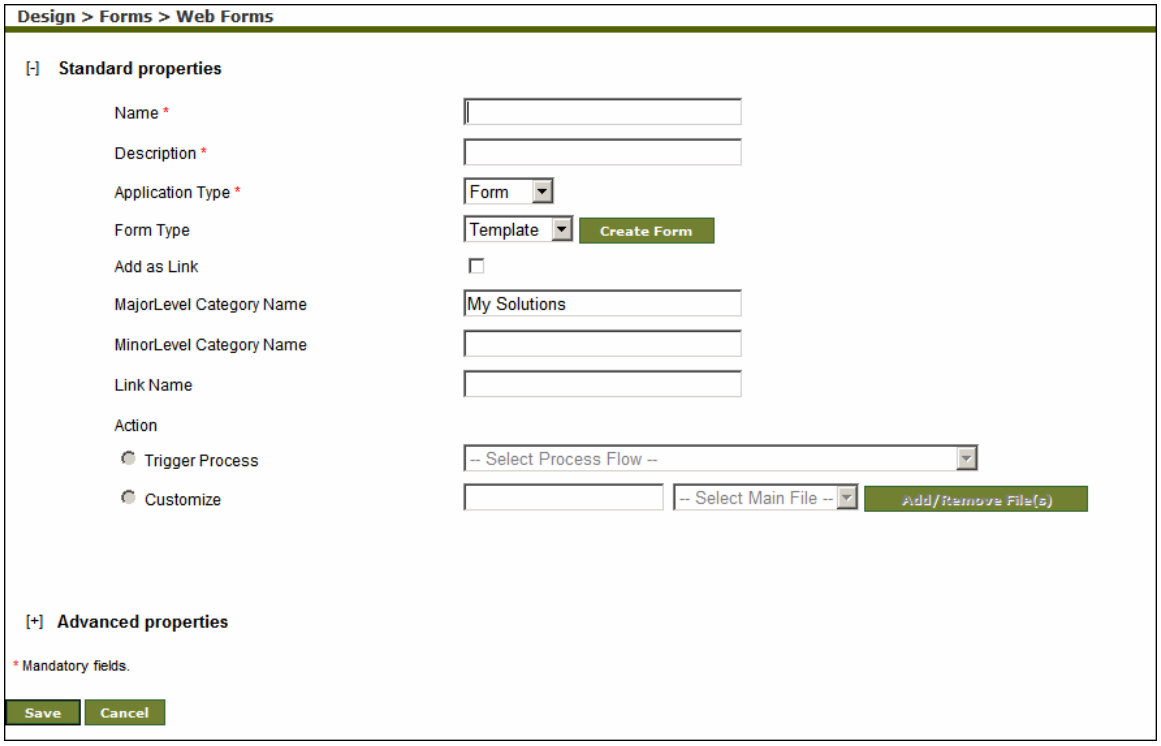


Figure 3.2: Create Web Form

4. Enter the name and description of the web form in the *Name* and *Description* fields respectively.
5. Select *Form* from the *Application Type* drop-down list. By default, *Form* is selected.
6. Select the method to create the form from the *Form Type* drop-down list. The methods used to create forms are described in the table below.

Table 3.1: Form Types

Form Type	Description
Template	When Template is selected, a *Form template* opens, where you can define the form.
Manual	When *Manual* is selected, you need to write complete HTML code for the form manually.

Rich Form	When *Rich Form* is selected, you can create and design a rich web form using the *Frevvo* tool.

7. To design the form using a template, select *Template* from the *Form Type* drop-down list. For details, refer to the Creating Web Form Using Template section.
8. To design the form manually, select *Manual* from the *Form Type* drop-down list. For details, refer to the Creating Web Form Manually section.
9. To design the form using a rich form, select *Rich Form* from the *Form Type* drop-down list. For details, refer to the Creating Web Form Using Rich Form section.
10. Click **Create Form** button. This displays the selected design form screen. Create the form as desired.
11. Select the *Add as Link* checkbox, if you want the custom form to appear as a link in the Workspace Menu.
12. Enter the major category name under which the link would appear in the Workspace Menu, in the *MajorLevel Category Name* field. By default, the link appears in **My Solutions** menu in the Workspace Menu. Thus, the major category name is already entered as My Solutions. You can edit it if required. However, if you edit this and enter another MajorLevel category Name (for example, *Customer*), then **Customer** appears as a link at the bottom of the Workspace Menu. The link is added in a hierarchical format. First the MajorLevel Category Name link is displayed. Under this, the MinorLevel Category Name is displayed, and under that is displayed the link name. The web form is uploaded under this link.
13. Enter the minor category name to in the *MinorLevel Category Name* field. For example, if you enter Insurance Policy in this field, it will be considered as a *MinorLevel Category Name*.
14. Enter the name of the link in the *Link Name* field. For example, you can enter the link name as *Claim Insurance Policy Form*. Under this link, you can upload the claim insurance form. Thus, the hierarchy for the above example will appear as *My Solutions -> Insurance Policy ->Claim Insurance Policy Form,* or *Customer ->Insurance Policy ->Claim Insurance Policy Form.*
15. Once you select the *Add as Link* option, the *Action* fields get activated. You can set an action once this web form is executed. You can either trigger a process flow or by uploading custom jsp files. To trigger a process flow, select the *Trigger Process* radio button and select the process flow that you want to trigger, from the drop-down list.
16. Alternately, to upload custom jsp files, select the *Customize* radio button.
17. Click **Add/Remove File(s)** button to upload the custom JSPs, HTMLs and support files. This displays the Add/Remove Files screen (see Figure 3.3).

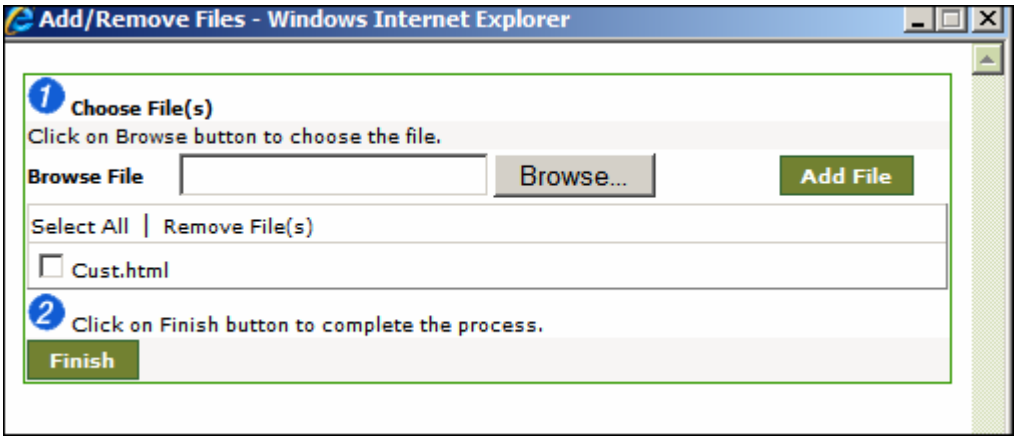


Figure 3.3: Add/Remove Files

18. Click **Browse** button to select the file to be uploaded. Once it is selected, it appears in the *Browse File* field.
19. Click **Add File** button to add the selected file. This displays the file with a checkbox. Similarly, you can add multiple files to be uploaded (see Figure 3.4).

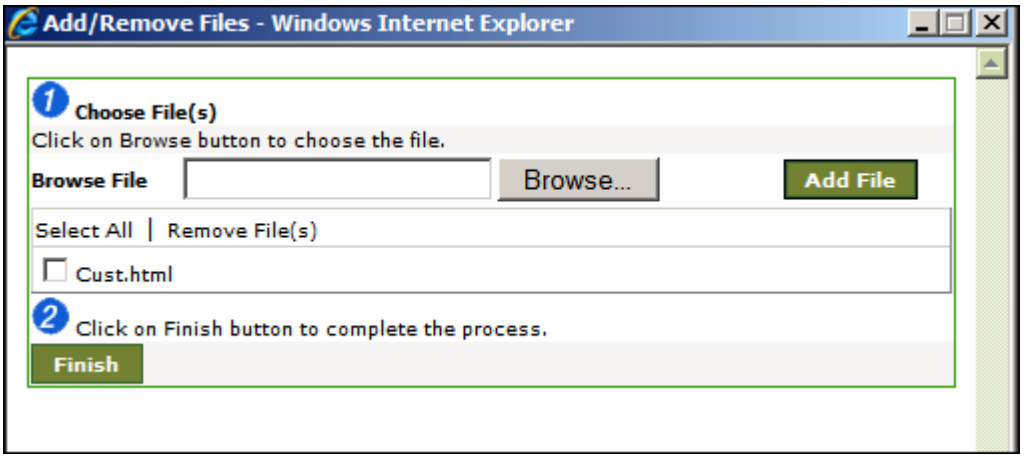


Figure 3.4: Added File(s)

There is no limit to the files that you can upload into the Adeptia Suite.

20. To remove the uploaded file, mark the checkboxes of the files that you want to remove and click **Remove Files** link. To select all files, click **Select All**.
21. Click **Finish** button. This closes the Add/Remove Files screen and displays the selected files in the *Customize* field (see Figure 3.5).

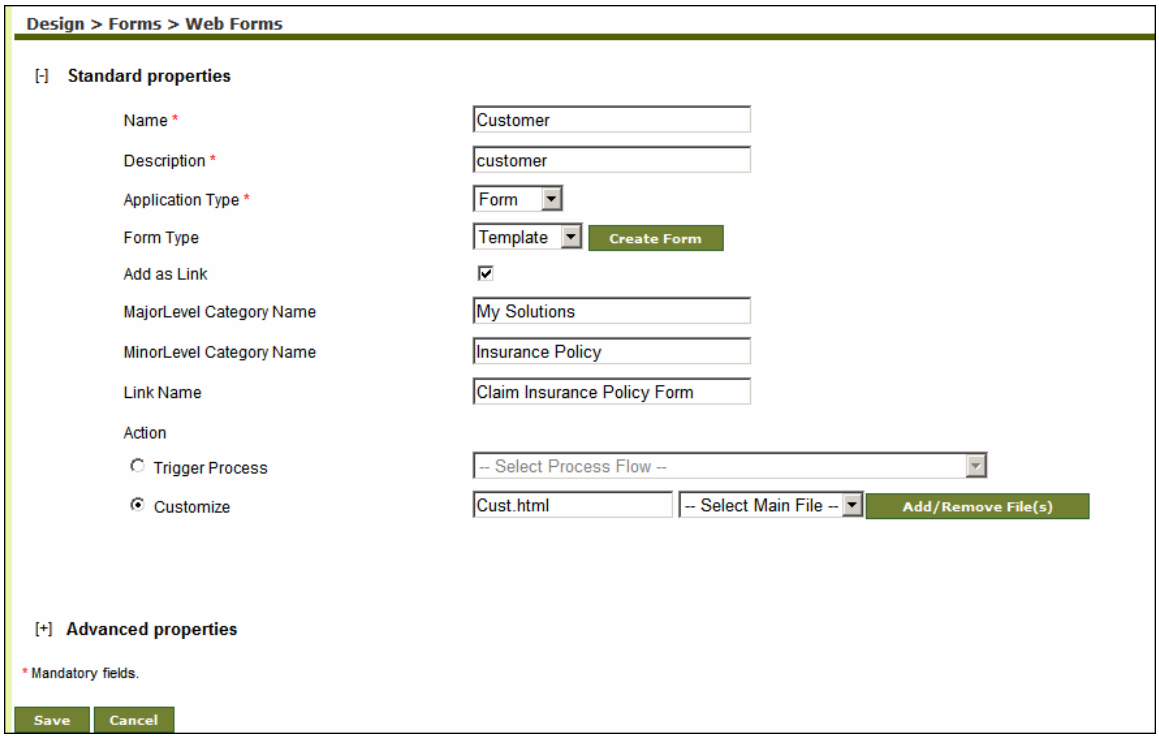


Figure 3.5: Uploaded File(s)

You can upload only JSP and HTML files and all their supported classes and jar files using this feature.

Once the files are uploaded, if an error occurs due to the connection being disconnected, then the system displays an error message.

When you upload the files, a folder is created with name as specified in *Link Name* field, in the `../../AdeptiaServer/AdeptiaServer-4.9/ServerKernel/web/custom` folder and the uploaded file are copied into this directory. If your application is referring to any JS, CSS or image file, you need to mention the path of these files relative to custom folder. Following is the sample HTML file.

```
<Html>
<TITLE> Adeptia BPM Server </TITLE>
<script                                    type="text/javascript"
src="Custom/ErrorReports/calendar.js"></script>
<Body>
<H1>
Hi!
</H1>
<img src='Custom/ErrorReports/nature2.jpg'>
</Body>
</Html>
```

	Here ErrorReports is the name of the link.

22. All the uploaded files are displayed in the *Main File* drop-down list. Select the file that you want to select as the main file from this list.
23. Click the **Save** button. A screen is displayed confirming that the web form has been created successfully. If the comments property is enabled, then clicking Save will display a screen where you need to enter comments related to creating the web form (see Figure 3.6).

Figure 3.6: Enter Comments

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

	The comment should be at least 1 character in length.

25. Click **OK** to save the comments. This displays a screen confirming that the web form has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

	You can create a Custom Application too from the Create Web Form screen. Select *Custom* from the *Application Type* drop-down list in the Create Web Form screen. This deactivates the *Form Type* field and the *Trigger Process* option. Enter the details as required and upload the custom jsp files.

Creating Web Form Using Template

Steps to create web form using Template

1. Select *Template* from the *Form Type* drop-down list in the Create Web Form screen.
2. Click **Create Form** button. This displays the Design Form using Template screen (see Figure 3.7).

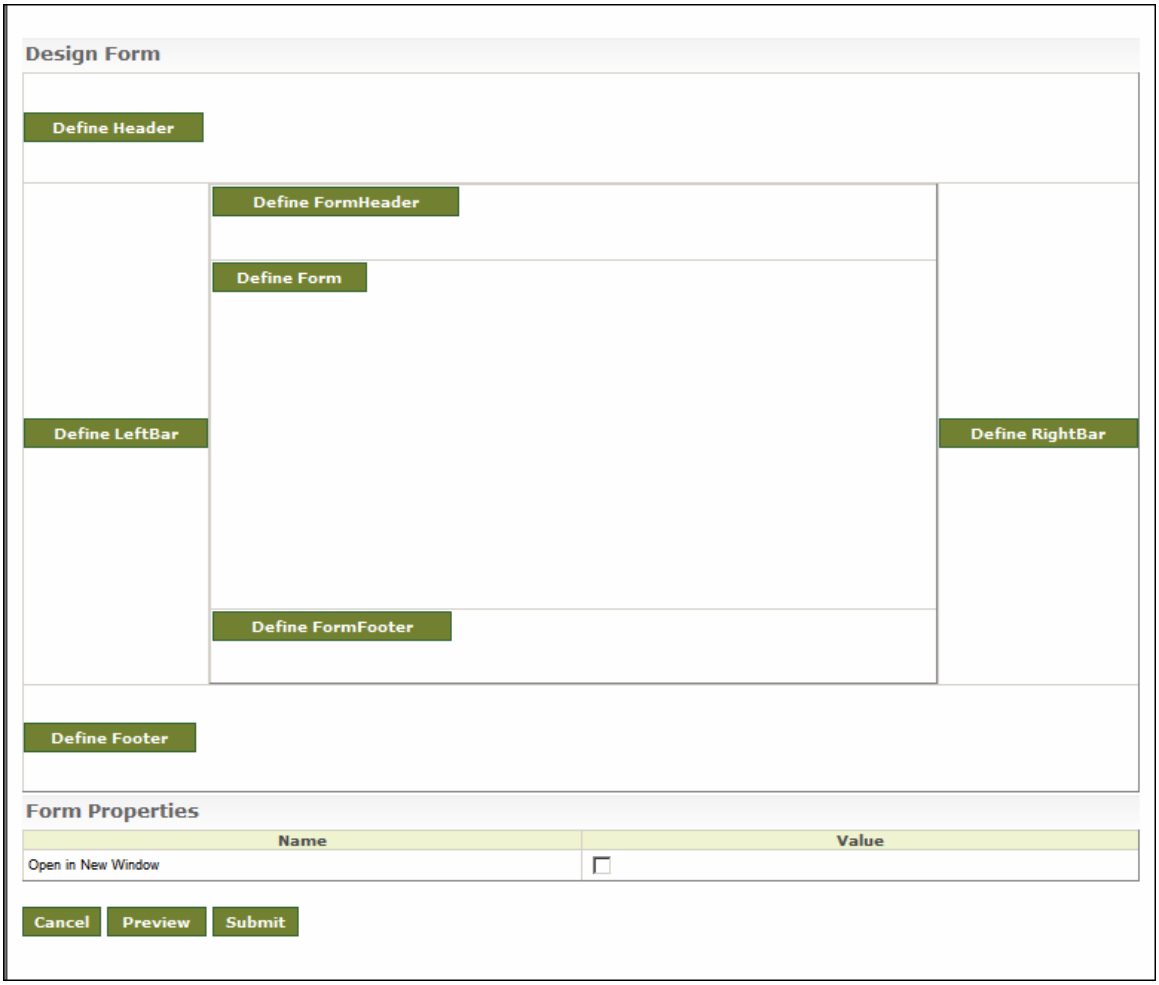


Figure 3.7: Design Form using Template

3. To define header, click **Define Header** button. The following screen is displayed (see Figure 3.8).

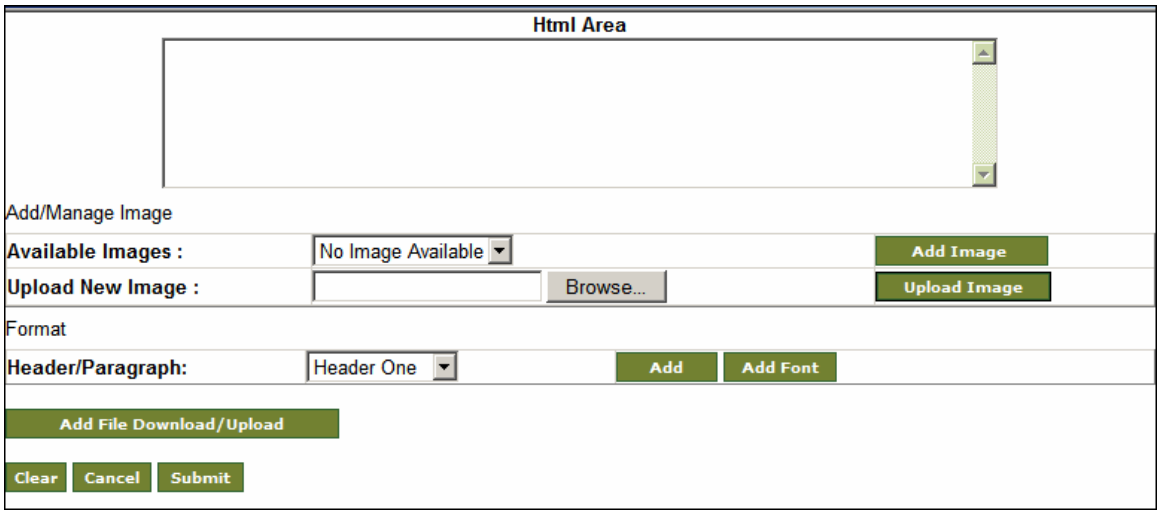


Figure 3.8: Define Header

4. To add an image into header, select the image from *Available Images* drop-down list and click **Add Image** button.
5. To upload any new image in the list of Available Images, browse the required image and click **Upload Image** button. The selected image is added to the *Available Images* list.
6. Select header type from *Header/Paragraph* drop-down list and click **Add** button. Tags for selected header are added into HTML area. Enter the required text between the header tags.
7. To change the fonts of the text, click **Add Font** button and select the required font and color.
8. To add link for File Download/Upload option, click **Add File Download/Upload** button.
9. Click **Submit** button to return to Design Form using Template screen.
10. Repeat step 3 to 8 to design other portion (except Form) of the web form.
11. To define the form, click **Define Form** button. The Define Form screen is displayed (see Figure 3.9).

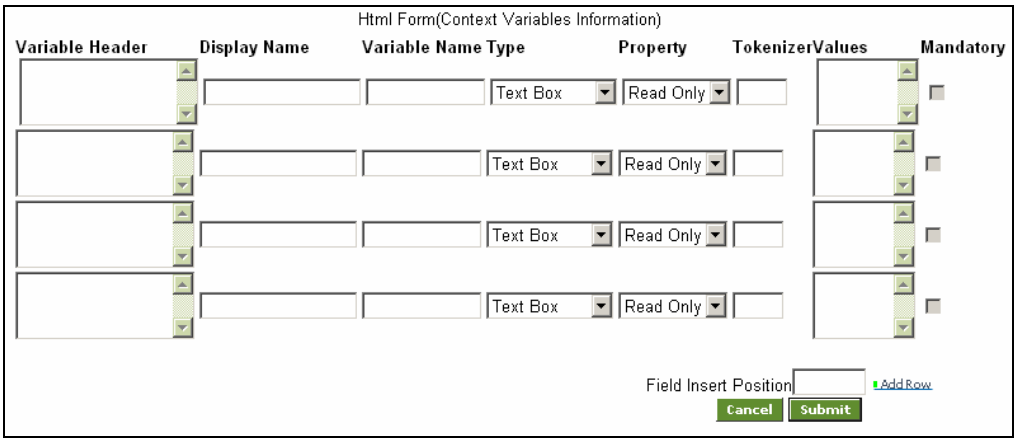


Figure 3.9: Define Form

12. Enter the name of the variable header in the *Variable Header* field.
13. Enter the name of the variable, which will be displayed in the HTML page, in the *Display Name* field.
14. Enter the name of the variable, corresponding to Display Name in the *Variable Name* field.
15. Select the type of the field from the *Type* drop-down list.
16. Select the property of the field whether Read Only or Editable from the *Property* drop-down list.
17. Enter the tokenizer (e.g. comma) in the *Tokenizer* field.
18. Enter the possible values of the field separated by the tokenizer character in the *Values* field.
19. Check the *Mandatory* checkbox against the field that you want to define as mandatory entry field.

	If the property of the variable is selected as *Editable*, variable is exposed in Process Designer, while creating process flow using this activity and you can further change its value. Incase the property is of the variable is selected as *Read Only*; you have to create the variable with same name in Process Designer. Otherwise Workflow task will give an error.

20. After defining variables for all required field, click **Submi**t button to return to the Design Form using Template screen.

	Click **Preview** button to view a preview of the new HTML template, before submitting it.

	By default, the new web form opens in the workspace area. To open it in a new screen, select the *Open in New Window* checkbox in the *Form Properties* section of the Design Form using Template screen.

21. Once you have created the template, click **Submit** button. This returns control to the Create Web Form screen.

Creating Web Form Manually

Steps to create web form manually

1. Select *Manual* from the *Form Type* drop-down list in the Create Web Form screen.
2. Click **Create Form** button. This displays the Design Form manually screen (see Figure 3.10).

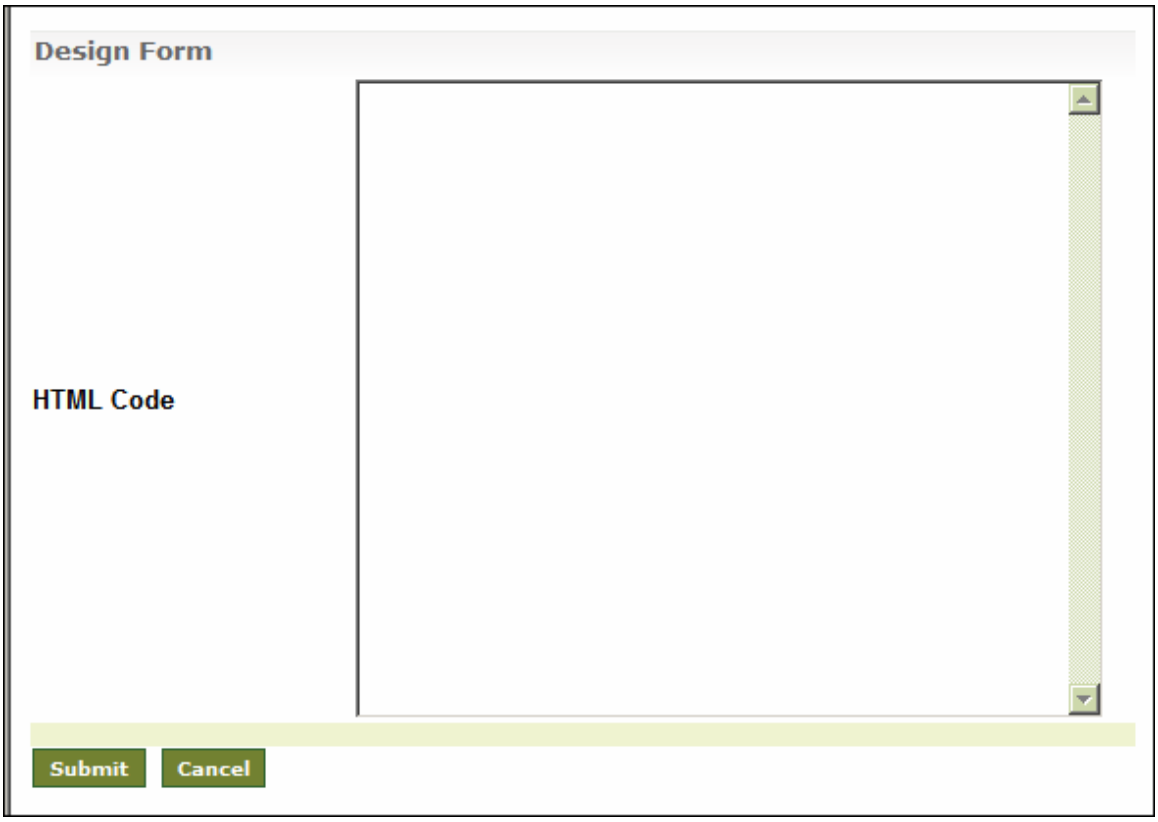


Figure 3.10: Design Form Manually

3. Enter the HTML code in *HTML Code* field to design the form. Sample HTML code which is used to design a form is displayed below (see Figure 3.11).

```
<TABLE height="90%" cellSpacing=0 cellPadding=0 width="100%" border=0>
 <TBODY>
 <TR height="15%">
```

```
   <TD colSpan=3>
    <H1 align=center><FONT face="Times New Roman" color=brown
size=8>Adeptia
    BPM Server</FONT></H1></TD></TR>
  <TR height="70%">
   <TD width="15%">
      </TD>
   <TD width="70%">
    <TABLE height="100%" width="100%" border=0>
     <TBODY>
     <TR height=10>
      <TD>
       <H1 align=center><FONT face="Times New Roman" color=brown
       size=6>Employee Management Form </FONT></H1></TD></TR>
     <TR height=80>
      <TD>
       <FORM name=HTMLForm>
       <TABLE>
        <TBODY>
        <TR>
         <TD>Employee Id</TD>
         <TD><BR><INPUT name=Employeeid type=text></TD></TR>
        <TR>
         <TD>Employee Address</TD>
         <TD><BR><INPUT name=Employeeaddress type=text
></TD></TR>
        <TR>
         <TD>Employee Work Experience</TD>
         <TD><BR><SELECT name=Workexperience><OPTION value=1
            selected>One</OPTION><OPTION
value=2>Two</OPTION><OPTION
            value=3>Three</OPTION><OPTION
          value=4>Four</OPTION></SELECT></TD></TR>
        <TR>
         <TD>Are these details are fine?</TD>
         <TD><BR><INPUT type=radio value=Yes
          name=Checkdetails>Yes<INPUT type=radio value=No
          name=Checkdetails>No</TD></TR>
        <TR>
         <TD colSpan=2><INPUT type=submit value="Save Task"
name=partialSubmit>
<INPUT type=submit value="Complete Task"
name=fullSubmit></TD></TR></TBODY></TABLE></FORM></TD></TR>
     <TR height=10>
```

```
<TD>
 <H1 align=center><FONT face="Times New Roman" color=brown
 size=4>Press Save Task button to save this form </FONT>
 <H1 align=center><FONT face="Times New Roman" color=brown
 size=4>Press Complete Task button to complete this form
 </FONT>
```

Figure 3.11: Sample HTML Code

Tips:

While creating a form, certain points need to be considered:

- User must specify HTML Form Name.
- User must specify only one **Complete Task** button. There should not be any other action i.e. Cancel or Back etc.
- User can also specify **Save Task** button. **Save** button is used when user wants to save the partial completed task.
- User must specify *fullsubmit* and *partialsubmit* variable for **Complete Task** and **Save Task** button respectively.
- There could be any number of radio buttons.
- Variable defined in the HTML Code is automatically gets created in the Process flow.

4. Once you have entered the HTML code, click **Submi**t button. This returns control to the Create Web Form screen.

When writing the HTML code, at times some business scenarios cannot be addressed by simple HTML code. To implement these scenarios, you need more dynamic behavior of Human Workflow web pages and interaction with server end to get the required information. The Adeptia Suite allows you to integrate the custom web application that you have created using JSP or AJAX.

Integrating Custom Web Application in Human WorkFlow

To integrate the custom web application, you need to take care of the following points:

- While writing HTML code for a Workflow activity, you need to define a hidden variable named PID using following code:

```
<input Type=hidden name=PID readOnly>
```

  This variable is used to pass the *Process Flow ID* (unique value generated for each execution of Process Flow) to the custom web application, which you have created.

- Create a hyperlink to call your custom web application and the corresponding Java script. The sample code (see Figure 3.12) creates a "Review" link in the Workflow page. Clicking the "Review" link calls the "review" function.

```
<a href='javascript:review();'>
<font color=blue>Review</font></a>
```

```
<script>
function review()
{
var pid=document.HTMLForm.PID.value;
var vWinTrans=window.open("custom/Review.jsp?PID="+pid
,"_blank","toolbar=yes,location=no,directories=no,status=no,menubar=
yes,scrollbars=yes,resizable=yes,copyhistory=no");
}
</script>
```

Figure 3.12: Sample Code for "review" Function

In the above mentioned code, *Review.jsp* is the custom jsp, which is called when you click the "Review" link in the Workflow page.

- In the custom web application, add the following code to access the value of Process Flow ID.

```
string pid=request.getParameter("PID");
```

- To create instance of class transactioninformation add the following code.

```
TransactionInformation ti=new TransactionInformation(request,session);
```

- A custom web application allows you to perform any kind of operation that is required. In addition, you can use Adeptia API to perform operations like logging information in process flow log, viewing variables, service objects etc. For Adeptia API documentation contact **support@adeptia.com**.

- Create a subfolder "custom" in *../../AdeptiaServer/AdeptiaServer-5.0/ServerKernel/web* and copy your custom JSP in the "custom" folder.

Creating Web Form Using Rich Form

You can create a rich internet form and use it as an HTML page in an activity. This form can be used for a Human Workflow activity or a custom application. Adeptia supports an external tool, the **Frevvo** tool for creating a rich form. For details on a rich form by Frevvo, refer to http://docs.frevvo.com/docs/index.php/V3.0_Main_Page .

Steps to create a web form using a Rich Form

1. Select *Rich Form* from the *Form Type* drop-down list in the Create Web Form screen.
2. Click **Create Form** button. This displays theAdeptia Rich Form screen (see Figure 3.13).

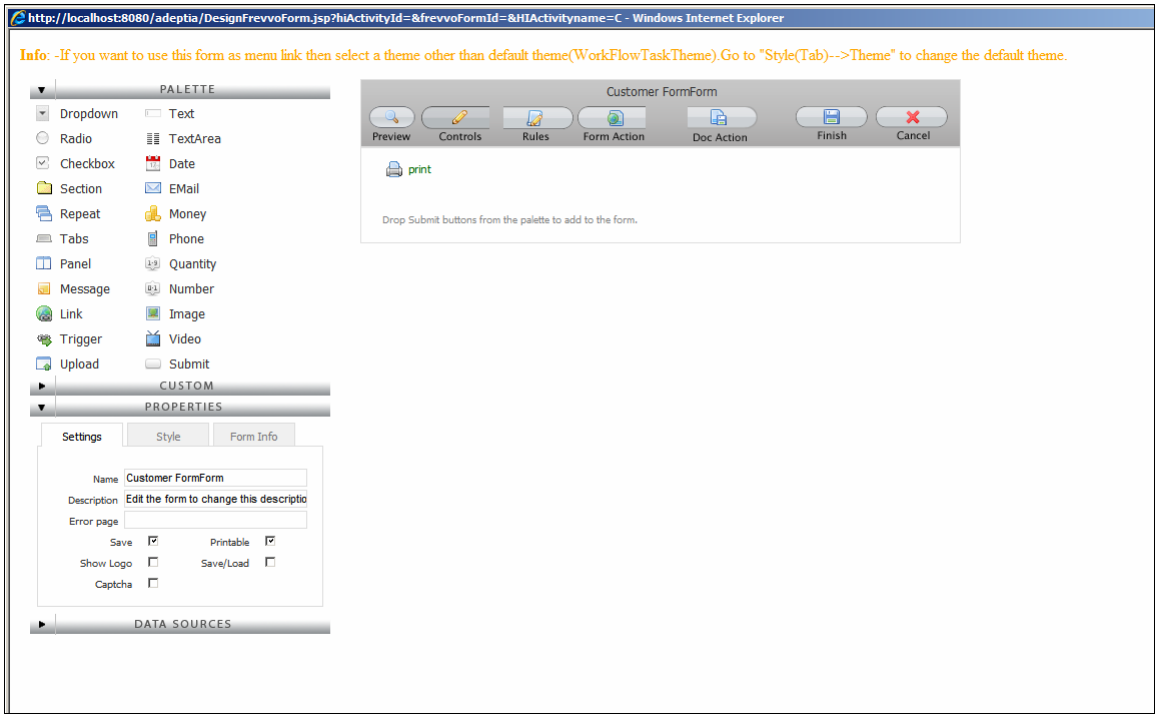


Figure 3.13: Adeptia Rich Form

3. This screen is divided into two parts. The left pane displays a palette of controls and their properties. The right pane displays a blank form. You can create a form by dragging the controls in the left pane and dropping them in the right pane. You can edit the control properties and also rearrange the controls.
4. When you drag and drop a control in the form canvas, it is displayed as a field with the control label. You can edit and customize the control by editing its

properties that are displayed in the Properties section in the left pane. You can edit properties such as the control name or control type.

5. Additionally, you can add a name for the control in the *Name* property, which is created as a variable in the process flow context and used as a process flow variable. For example, if you add a text control and add var1 in the *Name* property, var1 is created as a process flow variable and used in the process flow, when required.

6. Furthermore, you can customize the control by defining its maximum length and hints, error or help messages. You can also manipulate its visibility and validations such as mandatory and password entries.

	The properties can vary for each control. For details on each property, refer to **http://docs.frevvo.com/docs/index.php/V3.0_Designing_Forms#Editing_Controls**.

7. You can also customize the appearance and style of each control by editing the style properties displayed in the *Style* tab in the *Properties* section (see Figure 3.14).

Figure 3.14: Style Properties

8. Additionally, you can also customize the style of the entire form by selecting the form and then clicking the *Style* tab.

9. A sample created rich form is displayed in Figure 3.15.

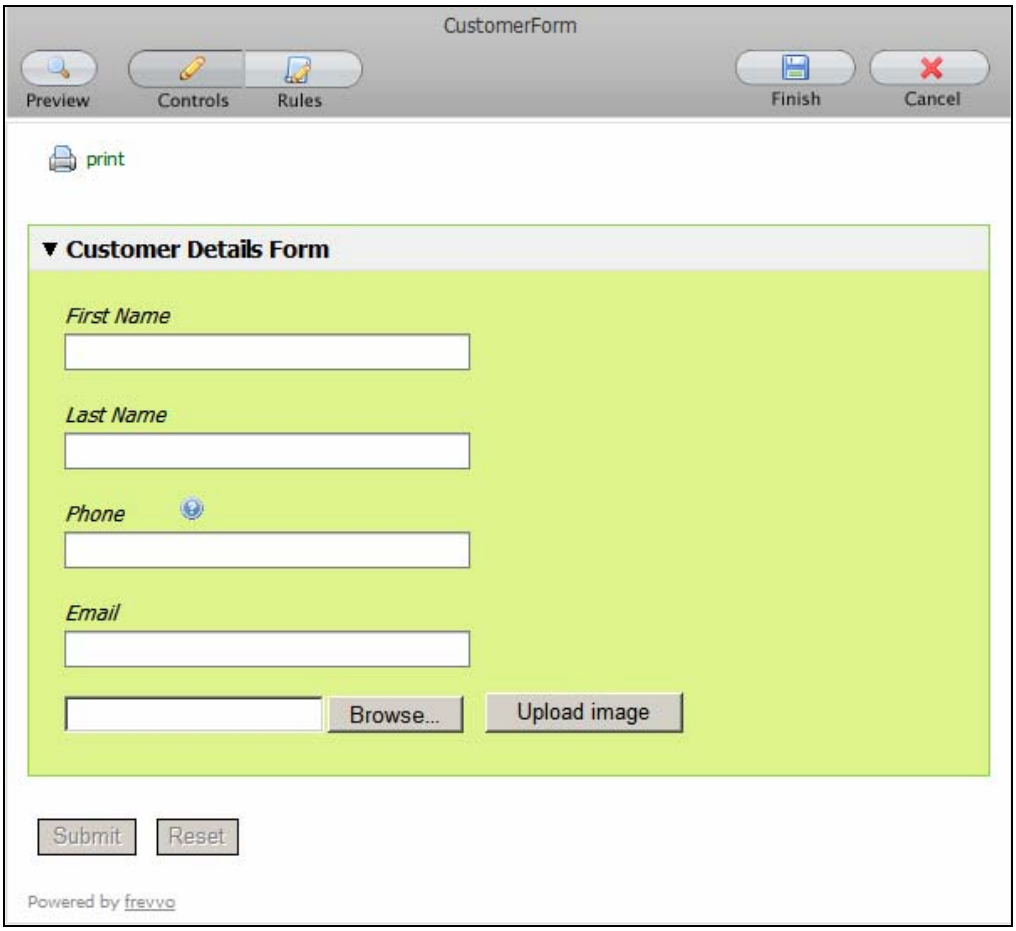


Figure 3.15: Sample Rich Form

10. The **Submit** and **Reset** buttons are displayed as disabled. You can remove these buttons from the form by changing its *theme* property. To do this, select the form and click the *Style* tab. Then, select *ClearThemeWithHiddenSubmitButtons* in the *adeptia's themes*, from the *Theme* drop-down list.

	If you want the form to trigger a process flow, then you need to select the *ClearThemeWithHiddenSubmitButtons* theme.

	You can also edit other form properties such as enable print settings for the form, display **Powered by frevvo** logo in the form, etc. by selecting the form and clicking *Settings* tab.

11. Before saving the form, you can preview it in HTML, by clicking the **Preview** button. You can also define rules to dynamically modify the behaviour of the form by clicking the Rules button. For more details on creating rules, refer to **http://docs.frevvo.com/docs/index.php/Form_builder#Rules**.

12. Once you have completed the form, save the form by clicking the **Finish** button. It is saved in *C:\Program Files\AdeptiaServer\AdeptiaServer-4.9\ServerKernel\frevvo\WEB-INF\users\adeptia*.
13. Once you save the form, the control is taken back to the Create Web Form screen.

ACTIVATING A WEB FORM

By default, a web form is created in deactivate state. To access it, you need to activate it.

Steps to activate a web form

1. Select the web form you want to activate, in the Manage Web Forms screen, and click the **Activate** link. This activates the web form (see Figure 3.16).

Figure 3.16: Activated Web Form

	Clicking **Deactivate** will deactivate the application and remove the link from the Workspace Menu.

WORKFLOW

4 DESIGNING WORKFLOW TASK

A Workflow activity allows user(s) to interact with running process flows. It can provide data and/or get data from the process flows at runtime. Users can decide the execution path of the process flow based on the data processed.

For example, if a purchase order is issued for $75,000, it may require approval from the Manager. The user can use a WorkFlow activity in such a case. The Workflow activity allows you to design an HTML page to perform this task. This HTML page can have all details of the purchase order and a button to **Approve** or **Reject**. Further, the activity allows you to assign this task to required person such as the manager, in this case. Once the Workflow activity is created, you can use this activity in the required process flow. When the process flow is executed, this activity gets listed as a task in the Manager's Task List. The Manager can open that task and click appropriate button to accept or reject the purchase order. If the manager accepts it, then the process flow will continue, else a rejection message will be sent to the user.

The Manager can enter input by clicking the **Open Task** link on the Task Manager screen. This displays the HTML form, where the Manager can enter input and click **Complete Task** or **Save Task** button.

Thus, users can use the Workflow service to perform various tasks such as:

- Design HTML page to show details of the task to be executed
- Assign tasks to other users and send them emails requesting for their necessary actions. These assigned tasks get listed in the Task Manager.
- Set Due date and Expiry date of the assigned task
- Defer the task to other users in not completed with due date

At times, the execution of tasks listed in the Task Manager, may get interrupted due to the kernel getting disconnected. To retain the executed data, the Human Workflow activity is equipped with the *Recovery Support* feature, which recovers all the executed data, at the next login.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√		

This chapter describes the following tasks:

- **Creating Workflow Task**
- **Executing Workflow Task**

CREATING WORKFLOW TASK

Pre-Requisites

- Web Form to be used in the Workflow Task must be created.

Steps to create a Workflow Task

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Workflow** to expand the tree and then click **Workflow Task**. The Manage Workflow Task screen is displayed (see Figure 4.1).

Figure 4.1: Manage Workflow Task

3. Click the **New** link. The Create Workflow Task screen is displayed (see Figure 4.2).

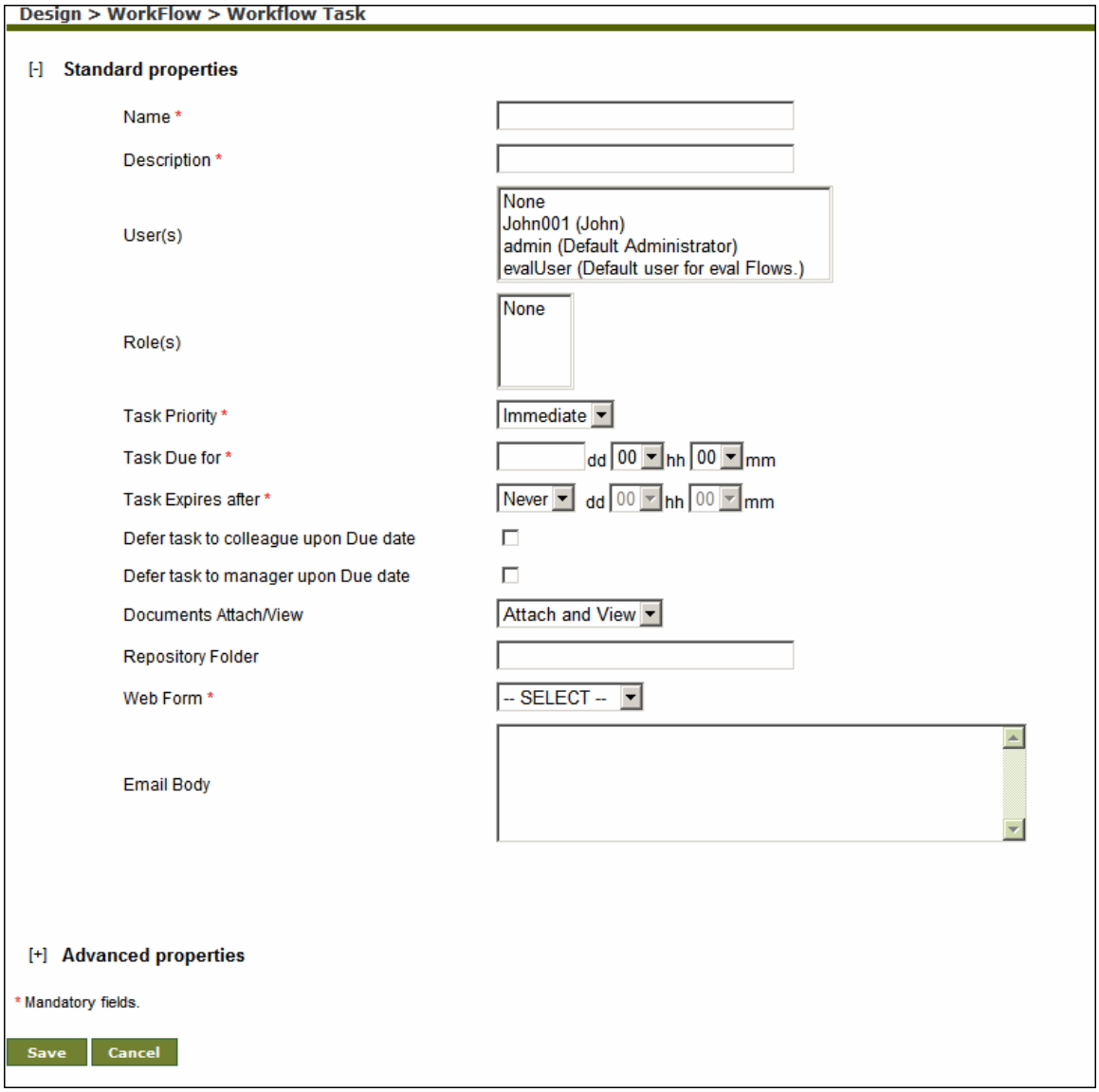


Figure 4.2: Create Workflow Task

4. Enter the name and description of the Workflow task in the *Name* and *Description* fields respectively.

5. Select the user to whom you want to assign this task, from the *User ID* list box. You can override this user with another user dynamically, at the time of process flow execution. For details on how to override assignee during process flow execution, refer to **Overriding Assignee User of a Human Workflow Task** section.

6. To assign this task to Business Role, select the Business Role from the *Role(s)* list box.

	To select more than one user or business role, press the **<Ctrl>** key and click on the user(s) or business role(s) with the mouse. To select no user or business role, select **None**. At least one user or business role should be

	selected. To learn about Business Role, refer to the *Creating Business Roles* section in the *Administrator Guide*.

7. Select the priority of this task from the *Task Priority* drop-down list. By default, *Immediate* is selected.

	Task priority does not have any direct impact on the task execution. It is just a flag so that user will be able to know that which task should be completed first.

8. Enter the due time for this task in days, hours and minutes, in the *Task Due for* drop-down lists.
9. Select the expiry time for this task in days, hours and minutes, from the *Task Expires after* drop-down lists.

	Time entered into *Task Due for* and *Task Expires after* fields are counted after the task is listed in the Task Manager not from the creation of the task. If a task is not completed within its due time, it will be listed in the *Over Due* list in the Task Manager of the user. If it is not completed with in its expiry time, then it will be deleted from the Task Manager.

10. Check the *Defer task to colleague upon Due Date* check box, to defer the task to a colleague, if it is not completed within its due date.
11. Check the *Defer task to manager upon Due Date* check box, to defer the task to the Manager, if it is not completed within its due date.

	Both the *Defer task to colleague* and *Defer task to manager* check boxes can be checked at the same time.

12. Select *Attach and View* to attach or view files from the *Documents Attach/View* drop-down list, if you want enable the option for attaching/viewing documents in the task manager.
13. Enter the name of folder, where the file attached with Workflow task is stored, in the *Repository Folder* field. This could be a WebDAV folder or process flow repository folder.

	▪ To specify, whether the defined folder is a WebDAV folder or a process flow repository folder, you need to change the **abpm.hi.repository.type** property. Refer to the *Administrator Guide* for details. ▪ If you set the *repository type* property as *WebDav* and leave the *Repository Folder* field blank, the files are saved into a default group folder created in "WebDAV folder" object. If files are stored in the default folder, its path would be *<group name folder\Process Flow name\Transaction Id>.* Here: *group name folder* is the folder of the group, the executor belongs to. *Process Flow name* is the name of the process flow, and

	Transaction Id is execution instance Id of the process flow. ▪ If you set the *repository type* property as *default* and leave the *Repository Folder* field blank, the files are saved into process flow repository. ▪ It is important to ensure that the folder path is correct and executor has write permissions to the folder; else the process flow will be created but will fail at runtime.

14. Select the web form that you want to attach to the task, from the *Web Form* drop-down list. All web forms that are created are listed in this drop-down field.
15. Enter your email message in the *Email Body* field. If you do not specify this message, then the default email with subject is sent. The subject of the email that is sent is defined in the code. You can override this subject and the first line of the email body by dynamically changing it. Refer to **Overriding Email Subject** for details.

	To learn about Advanced Properties refer to section **Changing Advanced Properties** section.

16. Click **[+] Advanced Properties** to expand Advanced Properties. The Advanced Properties of the WorkFlow task are shown (see Figure 4.3).

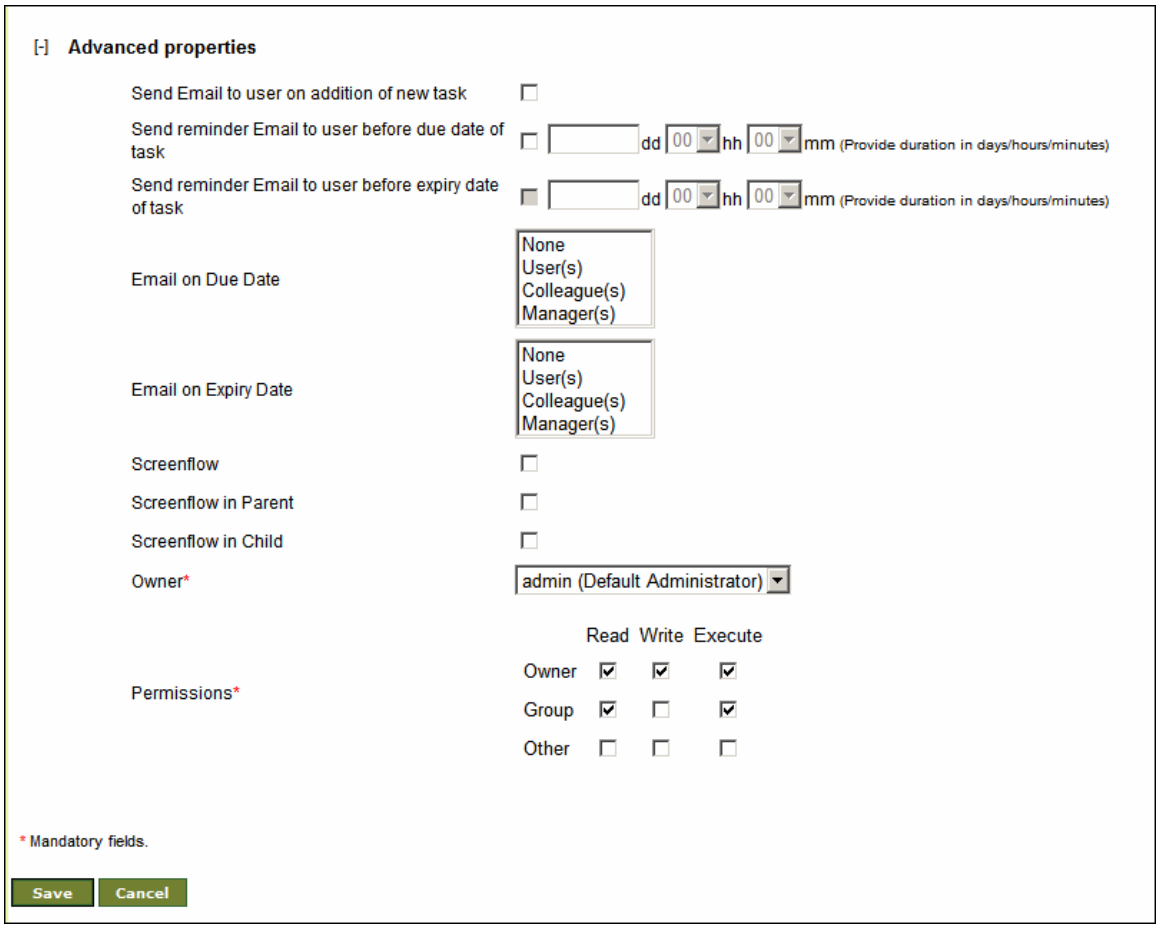


Figure 4.3: Advanced Properties of Workflow Task

17. If you want to notify the user by email when the task is listed in user's task list, check *Send Email to User on addition of new task* check box.

18. If you want to send a reminder email to user, before the task's due date, check the *Send reminder email to user before due date of task* check box and specify the time in days, hours and minutes. For example if you specify 1 day, the reminder email is send 1 day before the due date of the task.

19. If you want to send a reminder email to user, before the task's expiry date, check the *Send reminder email to user before expiry date of task* check box and specify the time in days, hours and minutes.

20. Select the User(s), Colleague(s) or Manager(s) from the *Email on Due Date* list box, to send an email to a user, colleague or manager if the task is not completed by due date and time.

21. Select the User(s), Colleague(s) or Manager(s) from the *Email on Expiry Date* list box, to send an email to a user, colleague or manager if the task is expired.

	Email sent to user(s), Colleague(s) or Manager(s), on Due Date or Expiry Date, contains URL of the Adeptia Suite Login Page (e.g. **http://localhost:8080/adeptia/control**).

	To login into Adeptia Suite and to access task manager, replace 'localhost' with name or IP address of the machine, where Adeptia Suite is running.

22. Select *Screenflow* checkbox, to open the subsequent task in the same screen, once the previous task is executed, in case of multiple tasks being assigned to you.

	If you are setting multiple tasks to be executed one after another in a process flow, and the *Screenflow* feature is enabled, then you need to attach the same type of web forms. For example, a process flow has 3 tasks to be executed one after another. If the first task has a Manual web form attached, then the remaining two tasks should also have Manual web forms attached. Adeptia does not support multiple web form types in a single process flow.

23. Select *Screenflow in Parent* checkbox, to open the subsequent parent task in the same screen, once the previous parent task is executed, in case of multiple tasks being assigned to you.
24. Select *Screenflow in Child* checkbox, to open the subsequent child task in the same screen, once the previous child task is executed, in case of multiple tasks being assigned to you.
25. Select the owner of the task in the *Owner* drop-down list.
26. Set the permissions by checking the appropriate checkbox(s) in the *Permissions* field.
27. Once you are done, click **Save** button. This displays a screen confirming that the workflow task has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the Workflow task (refer to Figure 3.6).
28. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

29. Click **OK** to save the comments. This displays a screen confirming that the workflow task has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

EXECUTING HUMAN WORKFLOW TASK

You can view a WorkFlow task assigned to you in your Task Manager and execute it by entering inputs.

In the Adeptia Suite, the Task Manager is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√		

Steps to execute a task assigned to a user

1. In the Adeptia Suite homepage menu, click **[+] My Tasks** to expand the tree. All the items in the **My Tasks** category are displayed.
2. Click **Task Manager**. The Task Manager screen is displayed with a list of tasks assigned to you (see Figure 4.4).

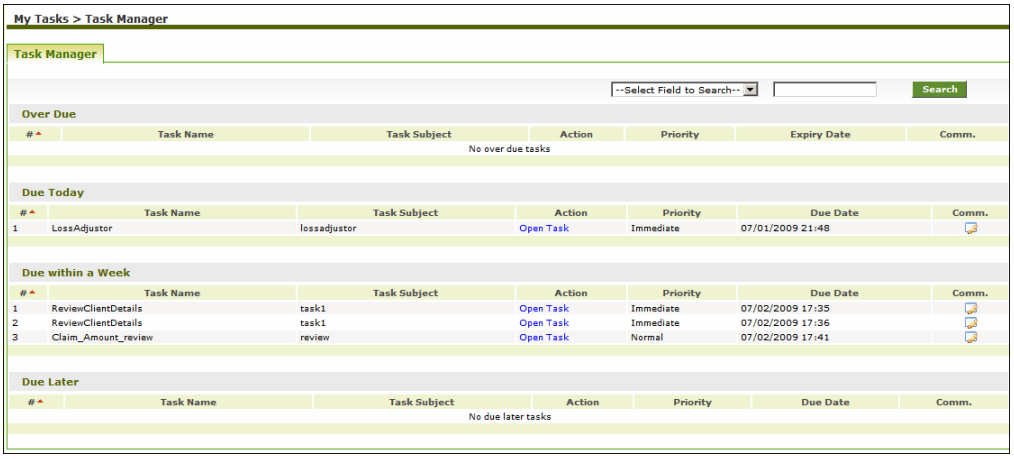


Figure 4.4: Task Manager

	You can dynamically set the task description during execution of the process flow. While creating the process flow, this description is specified by using the *put-context-var* action. The *put-context-var* is connected in the process designer, before the Workflow activity. It is attached to the activity, by right-clicking the *put-context-var* action and selecting **View Properties**, and then selecting the name of the Workflow activity from the *Activity* drop-down list. The description is specified by selecting "taskDescription" from *Key* drop-down list, and entering the required description in the *Value* field. This description is displayed in the *Description* field of the assigned task in Task Manager.

3. Click the **Comments** icon () to add comments to your task,. A dialog box is displayed to add the comments (see Figure 4.5).

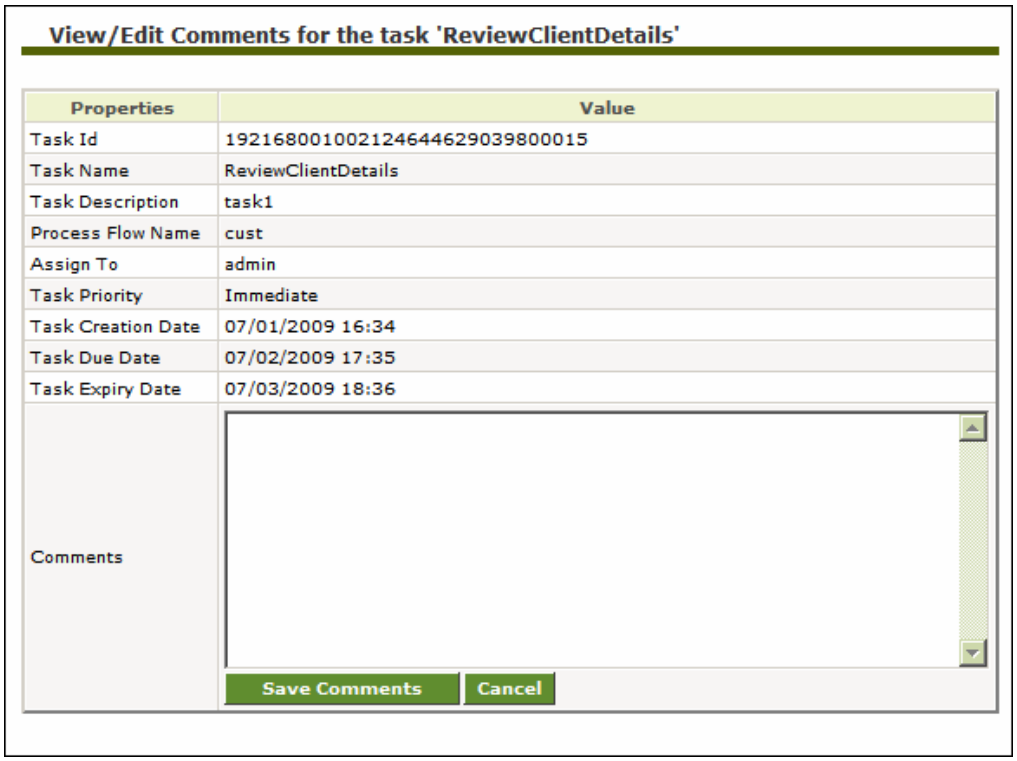


Figure 4.5: Comments

4. Enter your comments and click **Save Comments**.

5. Click **Open Task** link of the task you want to execute in the Task Manager screen. This opens the activity as displayed in the WorkFlow Task screen (see Figure 4.6).

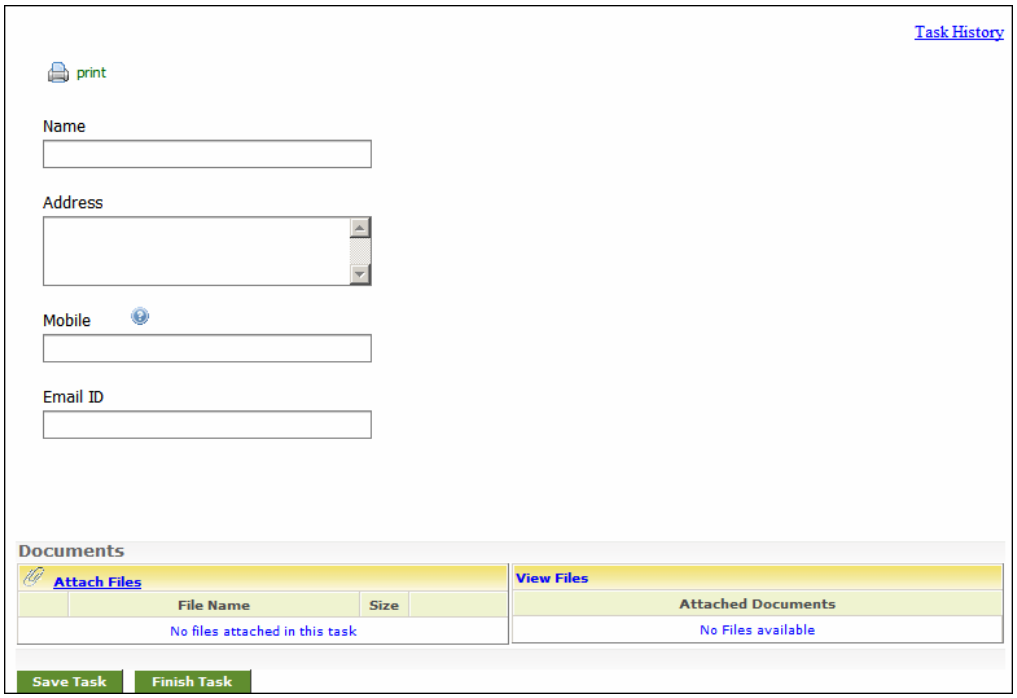


Figure 4.6: Workflow Task

6. Enter details as required.
7. Click the **Attach Files** link To attach the a file. This displays the File Upload screen (see Figure 4.7).

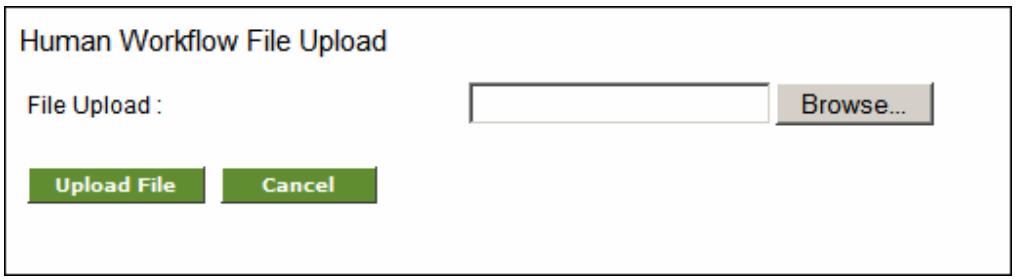


Figure 4.7: File Upload

8. Click **Browse** and select the file that you want to upload.
9. Click **Upload File**. This uploads the file and displays it in the *Attach Files* list in the Workflow Task screen (see Figure 4.8).

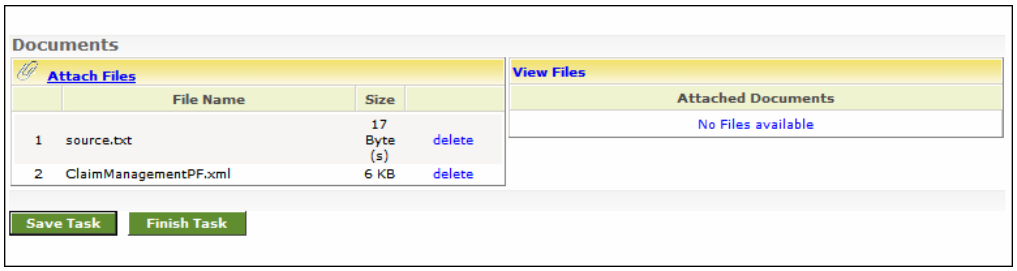


Figure 4.8: File Uploaded

10. Click **Save Task** button. Reopen the task by clicking **Open Task** in the Task Manager screen. This displays the Workflow task screen, with the attached file in the *View Files* list (see Figure 4.9). Click **View / Download** link to view or download the file.

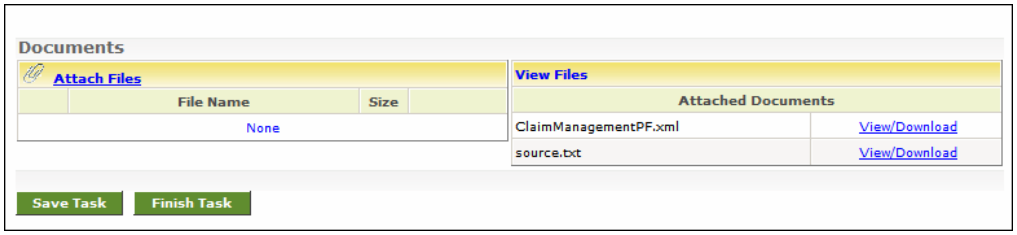


Figure 4.9: View Attached File

SERVICES

5 CREATING DATA DICTIONARY

A data dictionary is a collection of record definitions that are commonly used when a schema is created. You can just select the data dictionary and display the required records. This feature avoids redundancy of information and reduces the additional effort for creating a schema. Adeptia supports the Data Dictionary feature for Positional and EDI schemas only.

CREATING POSITIONAL DATA DICTIONARY

When creating Positional schemas, some record definitions may be common across schemas. You can create and define the record definitions commonly used in Positional schemas in a Positional Data Dictionary. Thus, when creating a Positional Schema, you can select the data dictionary and display the required records. Later, you can also view or delete a record from the Positional schema screen itself.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create Positional Data Dictionary

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Dictionary** to expand the tree and then click **Positional**. The Manage Positional Data Dictionary screen is displayed (see Figure 5.1).

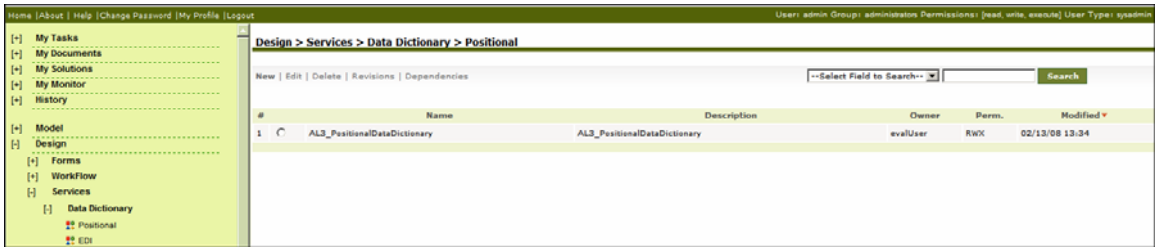


Figure 5.1: Manage Positional Data Dictionary

4. Click **New** link. The Create Positional Data Dictionary screen is displayed (see Figure 5.2).

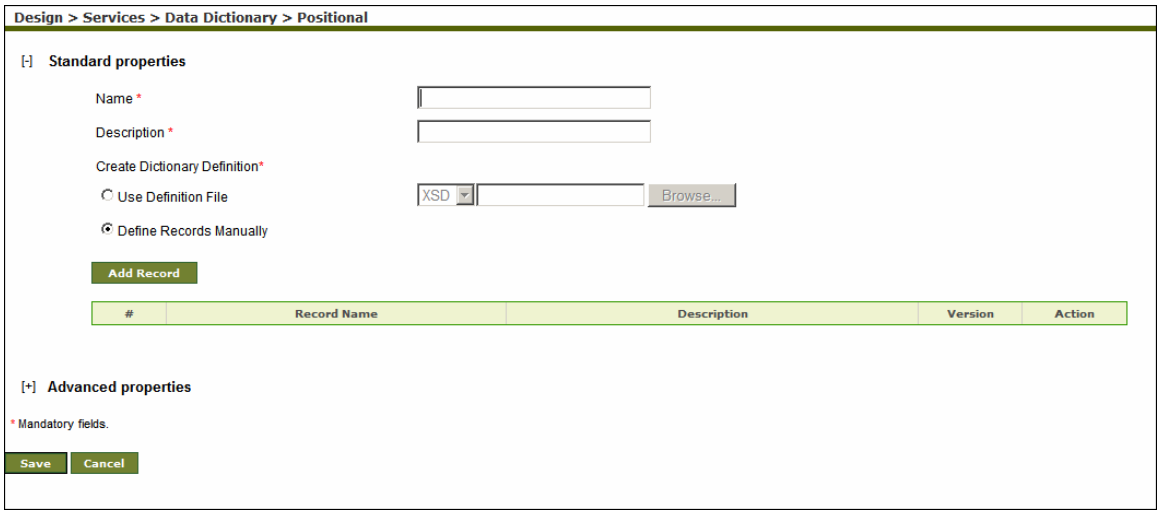


Figure 5.2: Create Positional Data Dictionary

5. Enter the name and description for Positional Data Dictionary in the *Name* and *Description* fields respectively.

You need to define records to be stored in the Data Dictionary. You can do this in two ways. These are outlined as:

- Use Definition file
- Define records manually

6. To define the records using **definition file** click the *Use Definition File* radio button, select the type of file from the drop-down list and click **Browse** button to select the required file. Alternately, to enter the records manually, click the *Define Records Manually* radio button and click **Add Record** button. This displays the Data Dictionary Record Builder screen (see Figure 5.3).

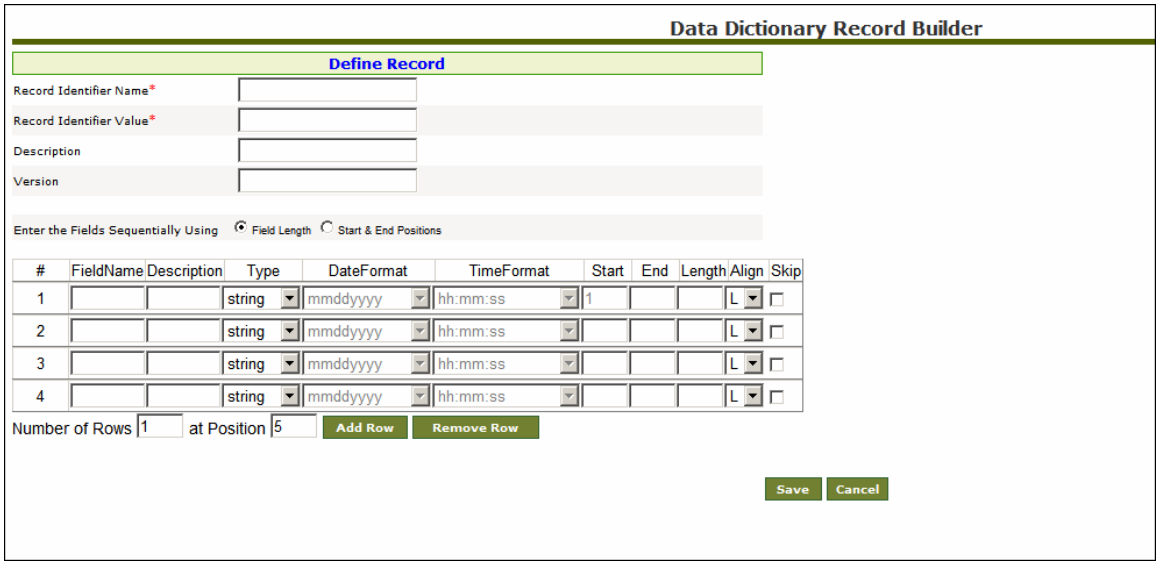


Figure 5.3: Data Dictionary Record Builder

7. Enter the name of the Record Identifier in the *Record Identifier Name* field.
8. Enter the value of the Record Identifier in the *Record Identifier Value* field.
9. Enter the description and version of the record identifier in the *Description* and *Version* fields respectively.
10. You can now enter fields for the record. Enter the name and description of the field in the *Field Name* and *Description* fields respectively.
11. Select the type of data from *Type* drop-down list.
12. If data type is *Date*, select the format of date and time from the *DateFormat* and *TimeFormat* drop-down lists respectively.
13. To define field position select one of the following options:
    - Field Length
    - Start & End Positions
14. To define the field position using field length, select *Field Length* radio button and enter the length of the field in the *Length* field. Alternately, to define the field position using start and end position, select the *Start and End Positions* radio button and enter the start and end positions of the field in the *Start* and *End* fields respectively.

The starting position of a row in a positional file is 1.

In a positional file, tab is counted as one position and not eight positions.

By default, field positions are created in sequence. You can also create a data dictionary with fields that are not in sequence. For details, refer to the **Defining Field Positions Non-Sequentially** section.

15. Select the alignment of the field from the *Align* drop-down list.

From *Align* field select

L if the field is left aligned.

R if the field is right aligned.

To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. A maximum of 99 rows can be added at a time.

16. Check the *Skip* checkbox if you want to skip this field while generating the XML. This selection skips the fields that are not required for the data dictionary.
17. Click **Save** to save the records. The saved records are displayed on the Create Positional Data Dictionary screen (see Figure 5.4). You can view, edit or delete a record from this screen by clicking the appropriate button for that record.

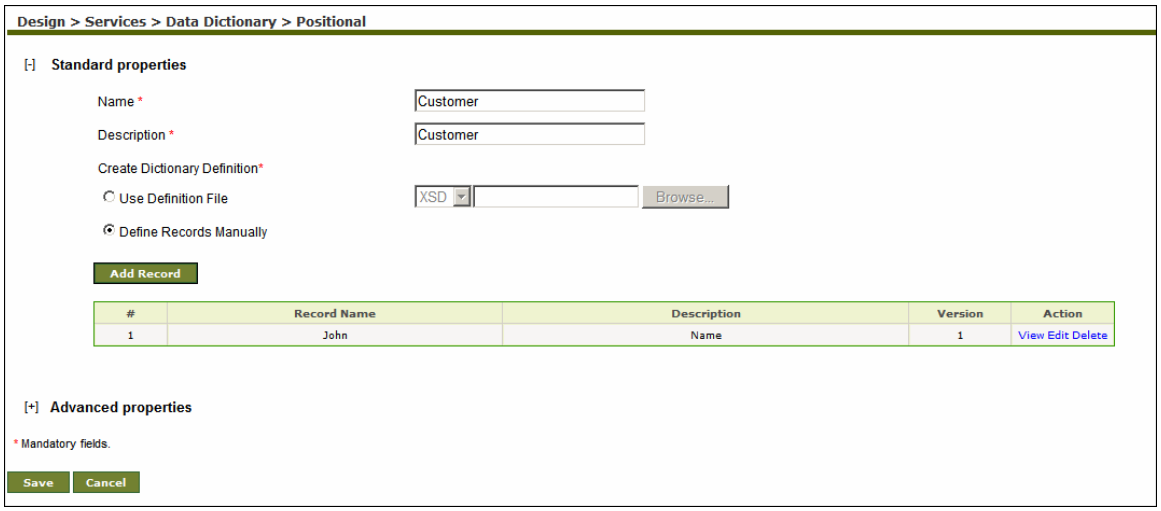


Figure 5.4: Records created for the Positional Data Dictionary

18. Click **Save** to save the Positional Data Dictionary. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the data dictionary (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

20. Click **OK** to save the comments. This displays a screen confirming that the data dictionary has been created successfully.

21. Once you save the Data Dictionary, you can view, edit or delete the data dictionary definition by clicking **View**, **Edit** or **Delete** links respectively. Additionally, you can also *print* or *download* the definition.

22. Select the data dictionary in the Manage Positional Data Dictionary screen and click **Edit** link. This displays the data dictionary in edit mode (see Figure 5.5).

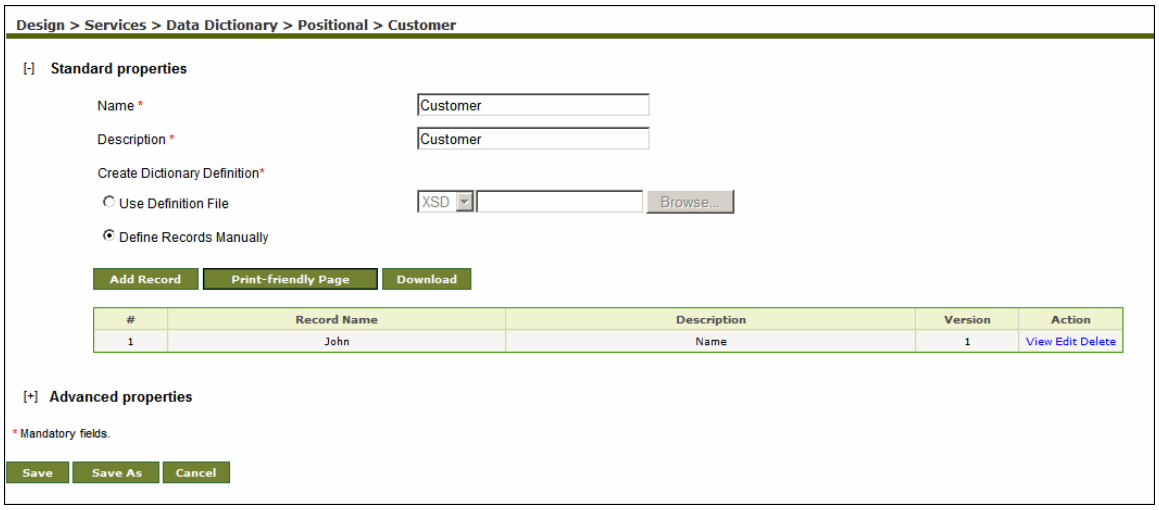
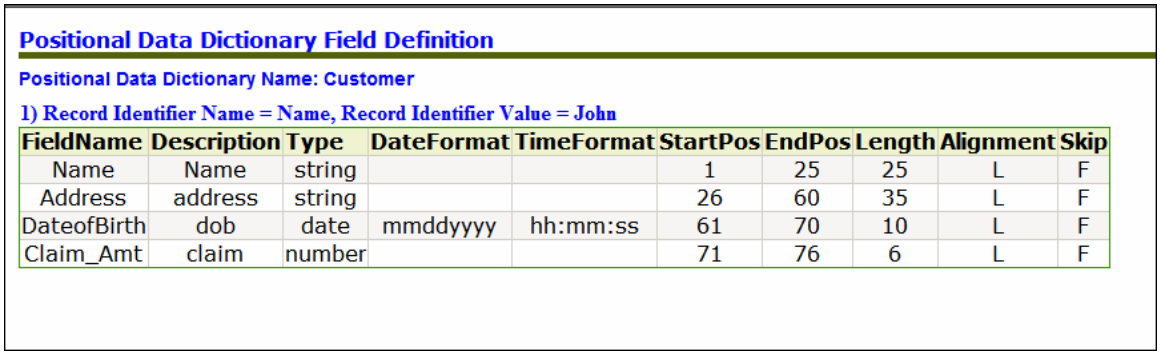


Figure 5.5: Edit Positional Data Dictionary

23. Click **Print-Friendly Page** button to view the dictionary definition in print mode in the web browser (see Figure 5.6). Alternately, click Download to download the XSD file of the selected dictionary definition (see Figure 5.7).



Positional Data Dictionary Field Definition

Positional Data Dictionary Name: Customer

1) Record Identifier Name = Name, Record Identifier Value = John

FieldName	Description	Type	DateFormat	TimeFormat	StartPos	EndPos	Length	Alignment	Skip
Name	Name	string			1	25	25	L	F
Address	address	string			26	60	35	L	F
DateofBirth	dob	date	mmddyyyy	hh:mm:ss	61	70	10	L	F
Claim_Amt	claim	number			71	76	6	L	F

Figure 5.6: Print Positional Data Dictionary Definition

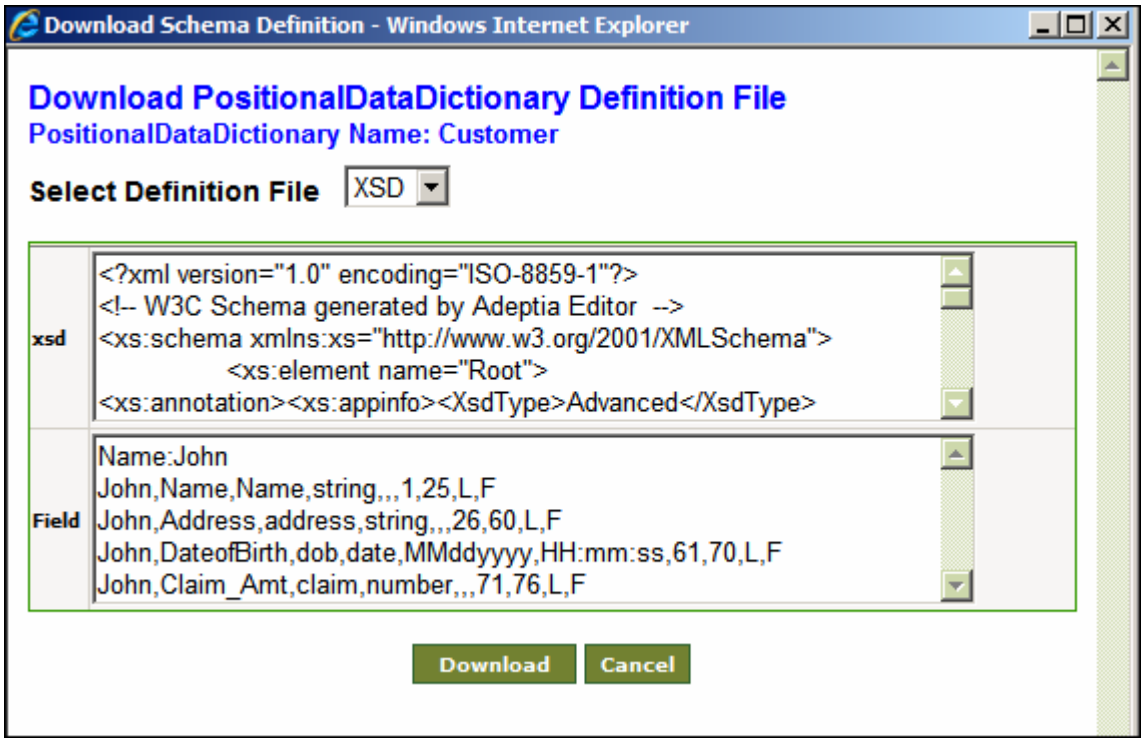


Figure 5.7:Download Positional Data Dictionary Definition

CREATING EDI DATA DICTIONARY

	This feature is a paid service and is thus not available in any of the Adeptia products by default.

When creating EDI Schemas, some record definitions may be common across schemas. You can create and define the record definitions that are commonly used in EDI schemas in an EDI Data Dictionary. Thus, when creating an EDI Schema, you can select the EDI Data Dictionary and display the required records.

Steps to create EDI Data Dictionary

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Dictionary** to expand the tree, and then click **EDI**. The Manage EDI Data Dictionary screen is displayed (see Figure 5.8).

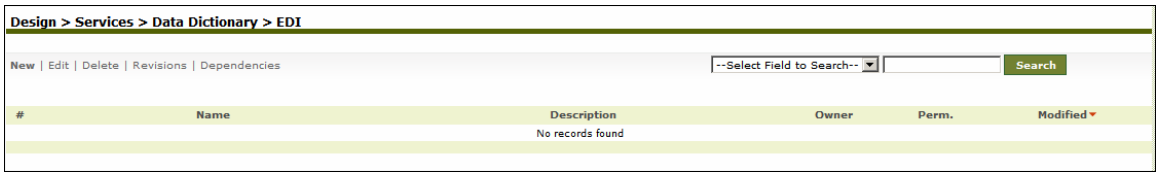


Figure 5.8: Manage EDI Data Dictionary

4. Click the **New** link. The Create EDI Data Dictionary screen is displayed (see Figure 5.9).

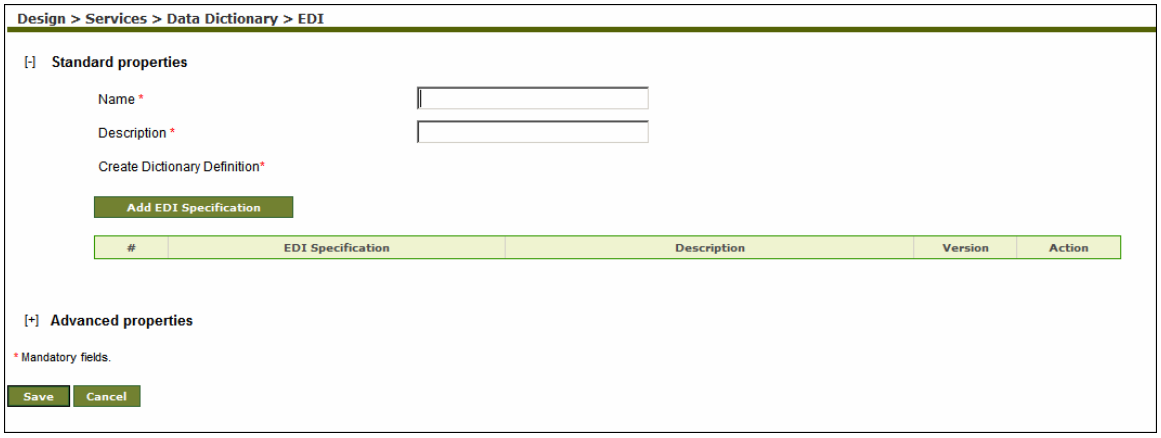


Figure 5.9: Create EDI Data Dictionary

5. Enter the name and description for EDI Data Dictionary in the *Name* and *Description* fields respectively.

6. Click **Add EDI Specification** button. The EDI Data Dictionary Builder screen is displayed (see Figure 5.10).

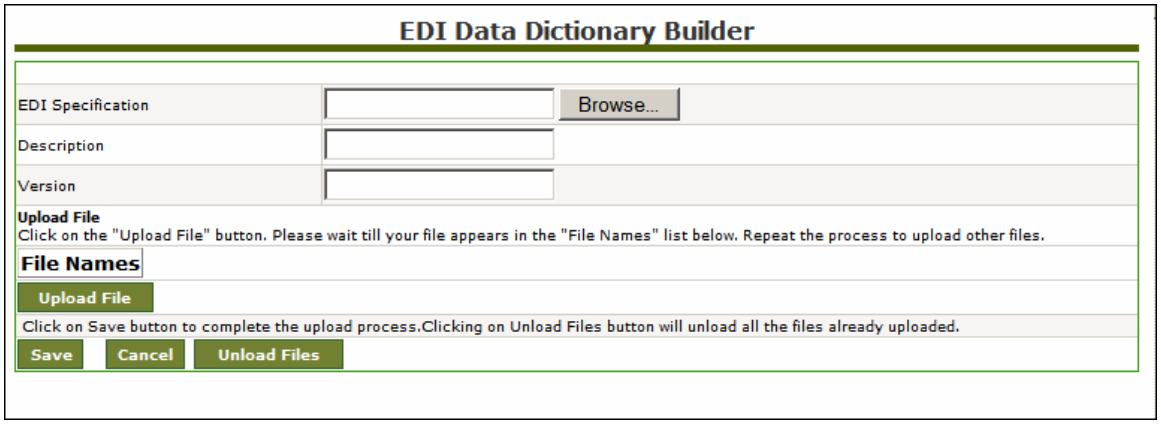


Figure 5.10: Select EDI Specification

7. Click **Browse** button and select the required EDI specification XSD.

8. Enter any description for the selected XSD in the *Description* field.

9. Enter the version of the selected EDI specification in the *Version* field.

10. Click **Upload File** button. The file name is displayed in the *File Names* list (see Figure 5.11).

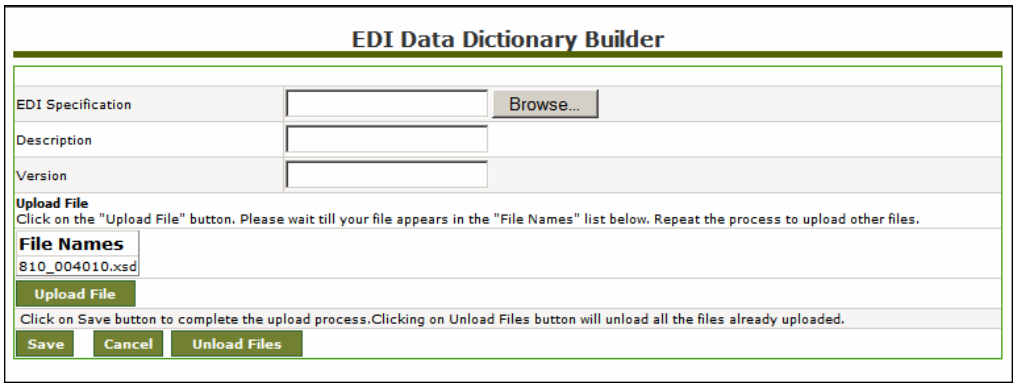


Figure 5.11: EDI Specification File Uploaded

11. Repeat steps 7 to 10 to upload additional XSD files.
12. To unload XSD file, click **Unload Files** button.
13. Click **Save** button to return to the Create EDI Data Dictionary screen. The uploaded XSD file(s) is displayed in the create EDI Data Dictionary screen (see Figure 5.12).

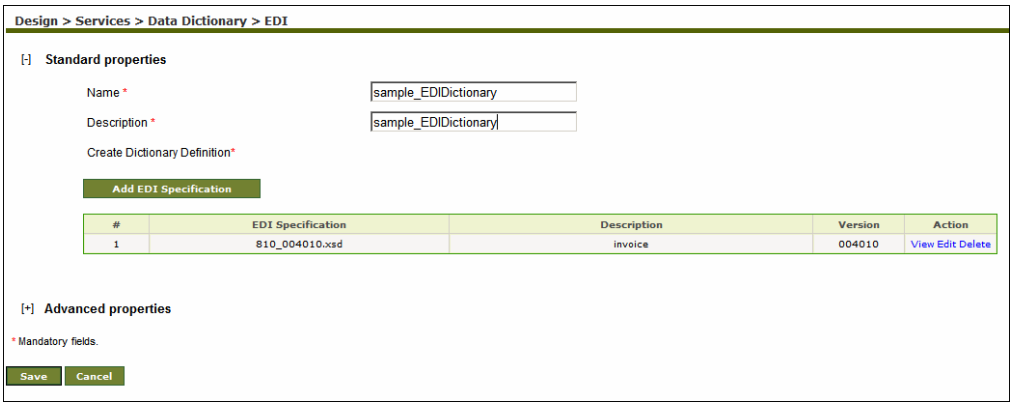


Figure 5.12: Create EDI Data Dictionary

	You can view, edit or delete the added EDI specification by clicking **View**, **Edit** or **Delete** link respectively.

14. Click **Save** button. This displays a screen confirming that the EDI Data Dictionary has been created successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the EDI Data Dictionary (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the EDI Data Dictionary has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

6 CREATING SOURCE ACTIVITY

The Source activity enables you to identify specific source data to be read, its location and the transport protocol that is used to retrieve data. The Adeptia Suite allows following types of Source activities:

- **Advance Database Source**
- **Database Source**
- **File Source**
- **FTP Source**
- **HTTP Source**
- **JMS Source**
- **LAN File Source**
- **Mail Source**
- **WebDAV Source**
- **Context Source**

Context Source is used to get the data from the process flow context. To know how to use Context Source refer to the section **Using Context Source and Context Target** activity.

When using a source activity in a process flow, the *eventContextEnabled* property is set to *True* by default. This implies that all variables are allowed to be passed from events to the source activity. For example, if a mail event has a file attached, its filename can be passed to the source activity easily.

CREATING ADVANCED DATABASE SOURCE ACTIVITY

The Advanced Database Source activity provides the ability to specify multiple tables of a database as source.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity and *Advance Database Schema* must be created before creating Advanced Database Source activity.

Steps to create an Advanced Database Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **Adv**. **Database**. The Manage Advanced Database Source screen is displayed (see Figure 6.1).

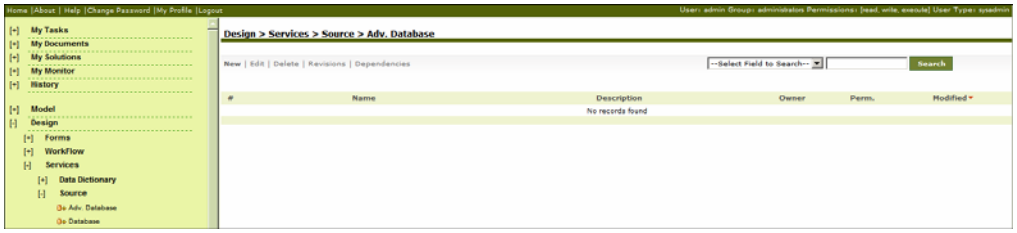


Figure 6.1: Manage Advanced Database Source

4. Click the **New** link. The Create Advanced Database Source screen is displayed (see Figure 6.2).

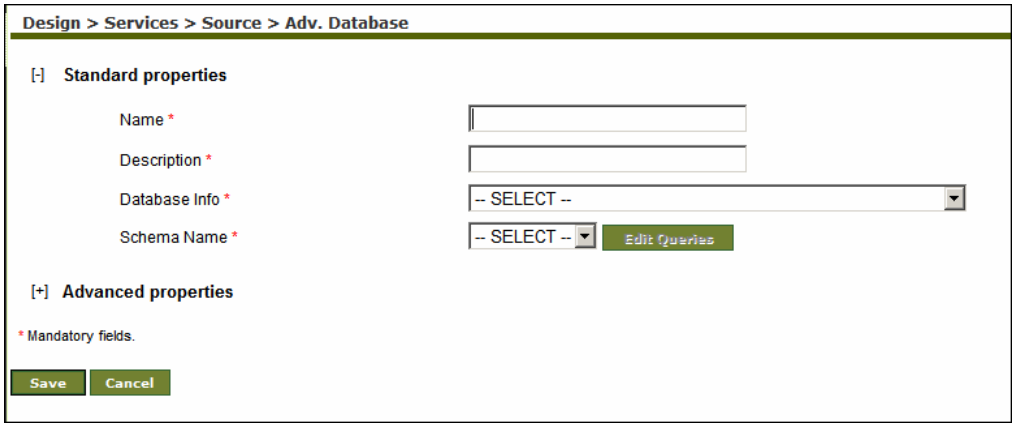


Figure 6.2: Create Advanced Database Source

5. Enter the name and the description of the new Advanced Database Source in the *Name* and *Description* fields respectively.
6. Select the database info activity and advanced database schema activity from the Database *Info* and the *Schema Name* drop-down lists respectively.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administrator Guide*. To learn how to create the Schema activity, refer to Creating Advance Database Schema Activity section.

7. To edit the database query of the selected schema, click **Edit Queries** button. The Edit Query(s) screen is displayed (see Figure 6.3).

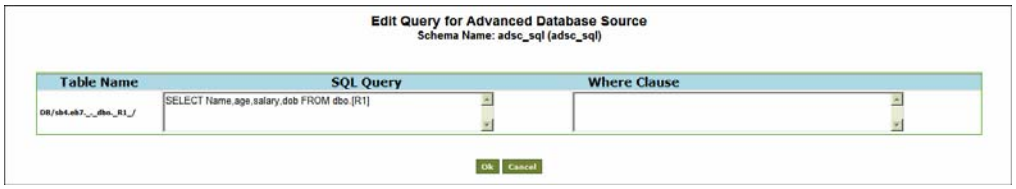


Figure 6.3: Edit Query

8. Edit the query and click **OK** button to submit the edited query and return to the Manage Advanced Database Source screen.

	▪ While creating Advanced Database Source, *Edit Query* button is disabled. You can edit the query only after saving the Advance Database Source activity. First you have to save the Advance Database Source and edit it to edit the query. ▪ You can edit the Query only for *Where* clause and not to change the table name or column names. To change the table name or select different columns, go to Advance Database Schema and edit the schema over there. ▪ If you specify where clause in Advance Database Source as well as in Advance Database Schema, the *Where* clause of Advance Database schema is used during execution. ▪ To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **[+]** to expand **Advanced Properties**.
10. In *Query Definition* field, Database Query is displayed in XML form. If you edit this query, database query defined using Edit Query button is overwritten.
11. You can check the *Use Optimize Algorithm* checkbox. If this option is enabled, data fetching from the source database table becomes faster. Enable this option only if the database source is an SQL database.
12. You can check the *With (No Lock) Option* checkbox. If this option is enabled, records are fetched from the source database without any lock constraint. This option is applicable, only when *Use Optimize Algorithm* option is enabled.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

13. Click **Save** button. This displays a screen confirming that the Advanced Database source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need

to enter comments related to creating the Advanced Database source (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

15. Click **OK** to save the comments. This displays a screen confirming that the Advanced Database source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING DATABASE SOURCE ACTIVITY

The Database Source activity provides the ability to specify a database as source.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity and *Database Schema* must be created before creating Database Source activity.

Steps to create a Database Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **Database**. The Manage Database Source screen is displayed (see Figure 6.4).

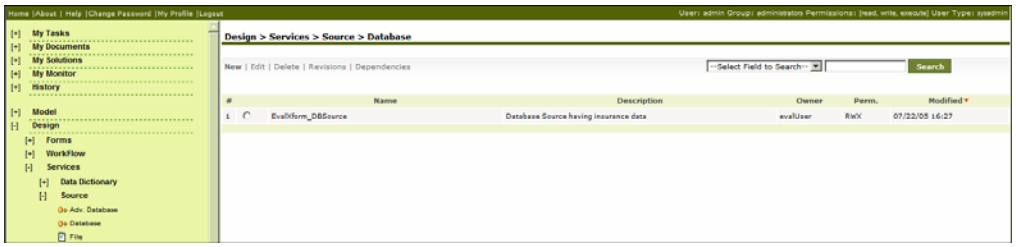


Figure 6.4: Manage Database Source

4. Click **the New** link. The Create Database Source screen is displayed (see Figure 6.5).

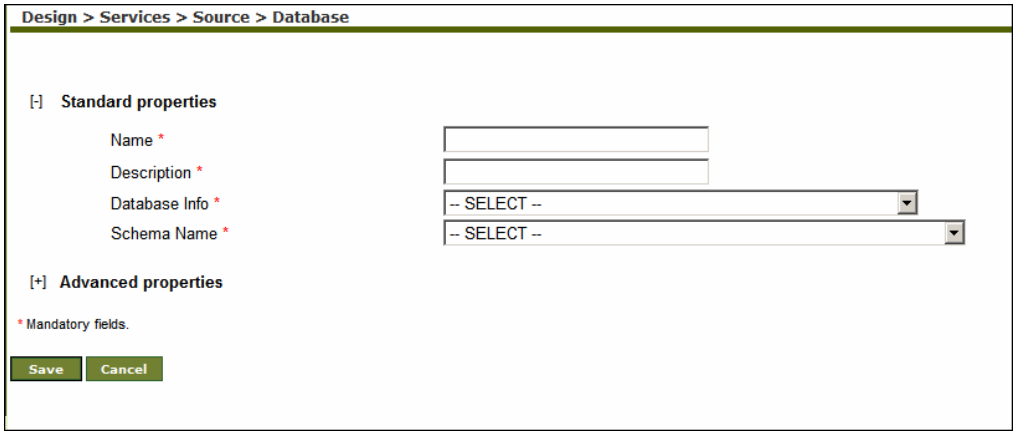


Figure 6.5: Create Database Source

5. Enter the name and the description of the new Database Source in the *Name* and *Description* fields respectively.

6. Select the database info activity and database schema activity from the *Database Info* and the *Schema Name* drop-down lists respectively.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administratoristrator Guide*. To learn how to create the Schema activity, refer to **Creating Database Schema** section.

	To learn about Advanced Properties refer to section **Changing Advanced Properties**.

7. Click **Save** button. This displays a screen confirming that the database source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the database source (refer to Figure 3.6).

8. Enter the 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 source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING FILE SOURCE ACTIVITY

The File Source activity provides the ability to specify any file that is located on the local hard disk, as a source.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create a File Source Activity

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

Figure 6.6: Manage File Source

4. Click the **New** link. The Create File Source screen is displayed (see Figure 6.7).

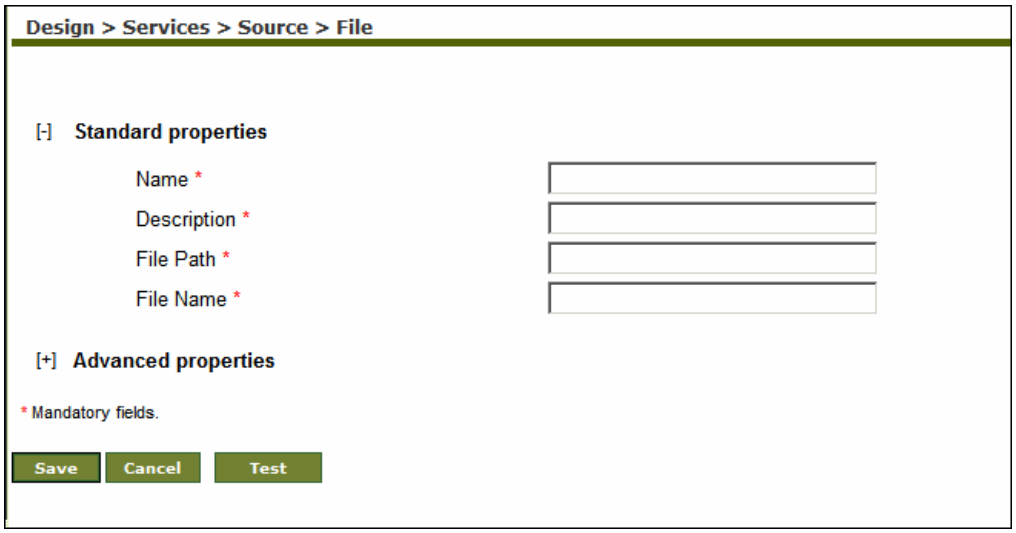


Figure 6.7: Create File Source

5. Enter the name and the description of the new File Source in the *Name* and *Description* fields respectively.
6. Specify the full path of the source file name in the *File Path* field. For example, if a file named sales.txt is stored in a directory Employee, on the host where Adeptia Suite is running, the File Path will be c:/Employee.
7. Specify the source file name in the *File Name* field. With respect to the above example, enter the file name sales.txt in this field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section. If you want to delete the source file after it is used in the process flow execution, you can check the *Delete File on Success* checkbox, while defining the Advanced Properties. This selection will delete the source file once the JTA block in which the source file exists, is executed successfully during the process flow execution.

8. Click **Save** button. This displays a screen confirming that the file source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the file source (refer to Figure 3.6).
9. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the file source activity at design time. For this, click **Test**. This verifies the values in the *File Path* and *Filename* fields and checks

	whether the file actually exists in the specified location.

CREATING FTP SOURCE ACTIVITY

The FTP Source provides the ability to specify a file as a source that is accessible via FTP.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a FTP Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **FTP**. The Manage FTP Source screen is displayed (see Figure 6.8).

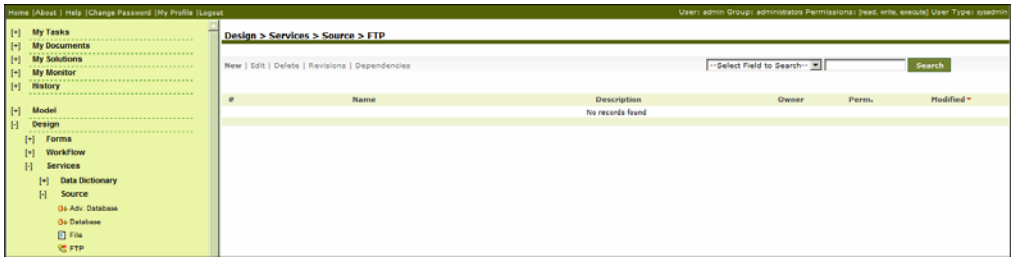


Figure 6.8: Manage FTP Source

4. Click the **New** link. The Create FTP Source screen is displayed (see Figure 6.9).

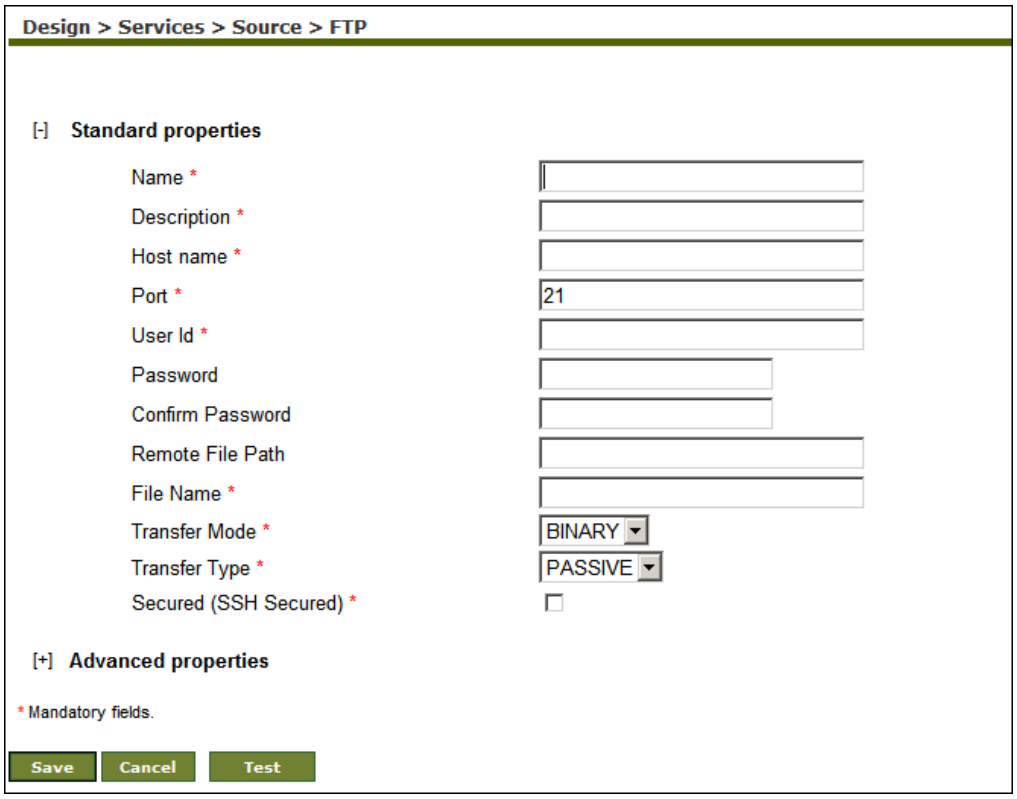


Figure 6.9: Create FTP Source

5. Enter the name and the description for FTP Source in the *Name* and *Description* fields respectively.
6. Enter the host name/IP address and port of the FTP Server in the *Host Name* and *Port* fields respectively.
7. Enter the username in the *User ID* field.
8. Enter the password, if required, in the *Password* field. Then re-enter the password in the *Confirm Password* field.
9. Enter the path of source file in the in the *Remote File Path* field.
10. Enter the name of the source file in the *File Name* field.
11. Select the transfer mode as either BINARY or ASCII from *Transfer Mode* drop-down list respectively. In BINARY mode a file is copied bit for bit from one machine to the other. Both files (the original and the transferred file) will contain exactly the same sequence of bytes. In ASCII mode a file may be changed slightly to maintain the meaning of EOL (End Of Line) characters.
12. Select the transfer type as either Active or Passive from the *Transfer Type* drop-down list. Active transfer is more secure since the client only initiates communication to the Server on one port whereas in case of Passive transfer the client initiates communication to the Server over two ports. Passive mode is useful when you are behind a firewall or a proxy.

13. Check the *SSH (Secure Shell)* checkbox if the FTP source accesses a secure FTP Server. When SSH is used to protect FTP Server, the control connection between the FTP client and Server is encrypted.

	▪ To learn about Advanced Properties refer to **Changing Advanced Properties** section. ▪ If you want to delete the FTP source file after it is used in the process flow execution, you can check the *Delete File on Success* checkbox, while defining the Advanced Properties. This selection will delete the FTP source file once the JTA block in which the FTP source file exists, is executed successfully during the process flow execution. ▪ At times, when the Adeptia Suite is connected with the FTP server, there could be connectivity issues. This could result in an infinite loop with no output. This issue can be resolved by setting a time limit if no data is transferred between Adeptia and FTP servers. You can set this limit (in seconds) in the *Data Timeout* field. By default, this is set to 60 seconds. This implies that if there is no data transfer for 60 seconds between both servers, the connection is broken and an error message is displayed.

14. Click **Save** button. This displays a screen confirming that the FTP source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the FTP source (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the FTP source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the FTP source activity at design time. For this, click **Test**. This verifies the values in the *Host Name, Port, User ID, Password* and *Secured* fields and checks whether the specified FTP Server exists or not.

CREATING HTTP SOURCE ACTIVITY

The HTTP Source provides the ability to specify a file as a source that is accessible via HTTP.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a HTTP Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **HTTP**. The Manage HTTP Source screen is displayed (see Figure 6.10).

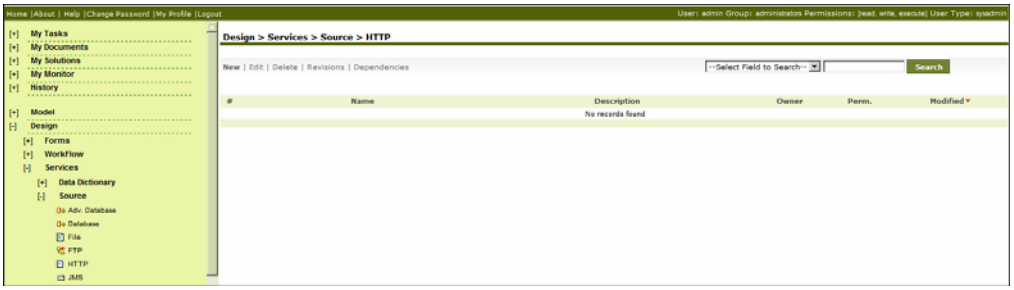


Figure 6.10: Manage HTTP Source

4. Click the **New** link. The Create HTTP Source screen is displayed (see Figure 6.11).

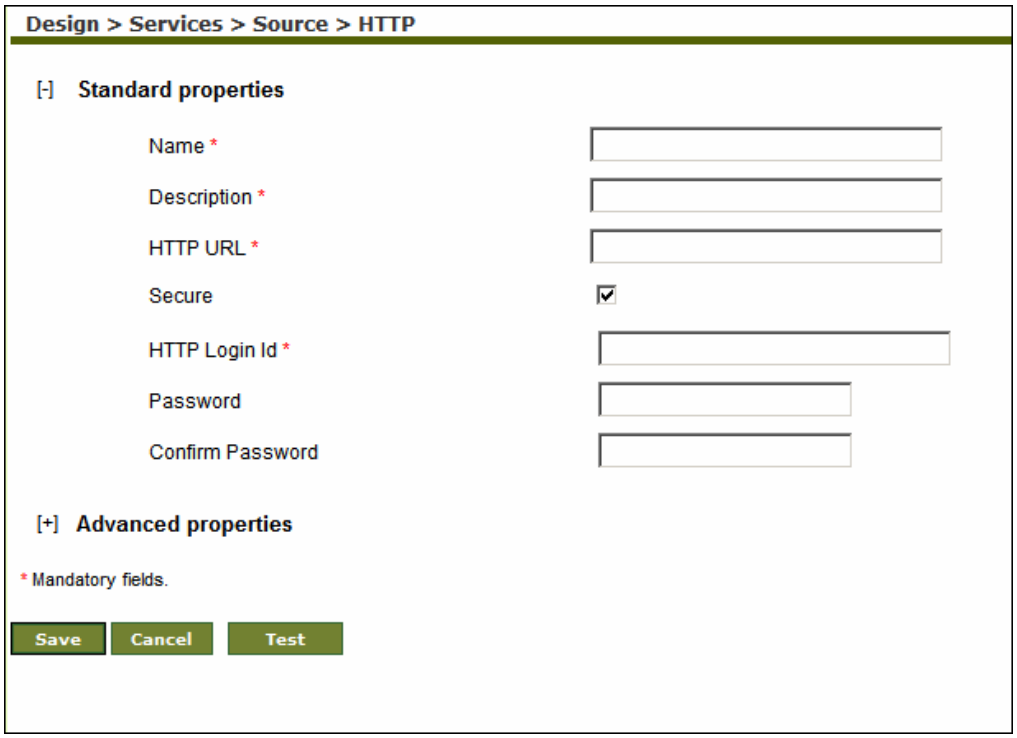


Figure 6.11: Create HTTP Source

5. Enter the name and description of the new HTTP Source in the *Name* and *Description* fields respectively.
6. Enter the URL of the HTTP Server in the HTTP URL field. For example, *http://www.adeptia.com/Employee/sales.txt*
7. If authentication is required to access the specified file then check the *Secure* checkbox.
8. Enter the username and password in the *HTTP Login Id* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. This displays a screen confirming that the HTTP source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the HTTP source (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the HTTP source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the HTTP source activity at design time. For this, click **Test**. This verifies the values in the *HTTP URL* and *HTTP Login Id* fields and checks whether the source activity actually exists in the specified location.

CREATING JMS SOURCE ACTIVITY

The JMS Source activity provides the ability to specify a message of a queue or topic of a JMS Server as a source.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Prerequisites:

- *JMS Provider* activity must be created before creating *JMS Source* Activity.

Steps to create a JMS Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **JMS**. The Manage JMS Source screen is displayed (see Figure 6.12).

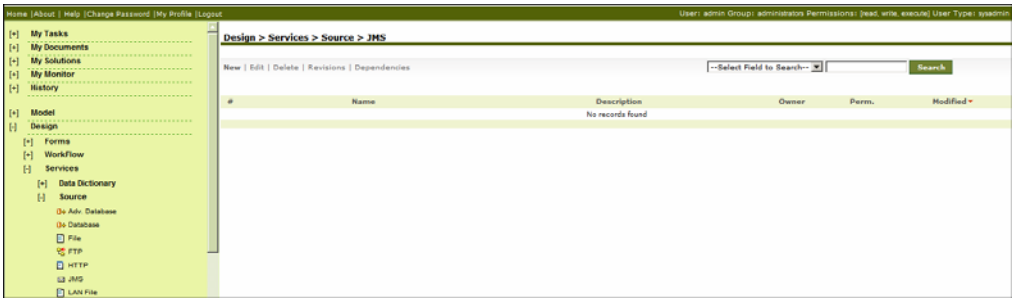


Figure 6.12: Manage JMS Source

4. Click the **New** link. The Create JMS Source screen is displayed (see Figure 6.13).

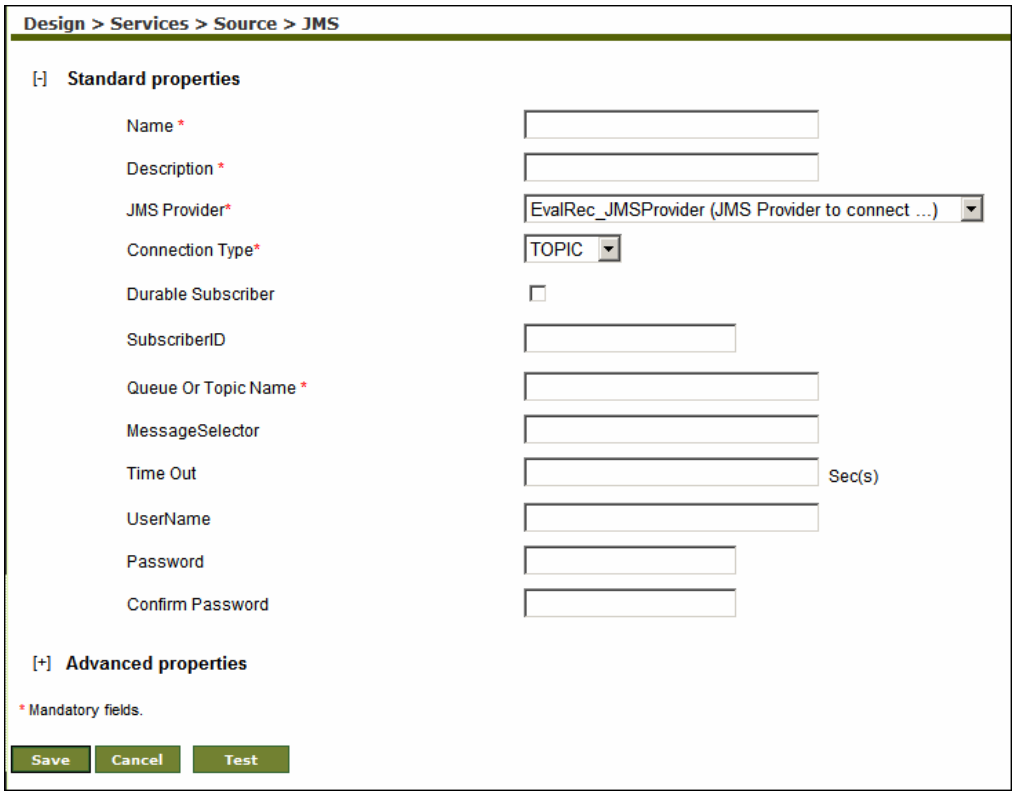


Figure 6.13: Create JMS Source

5. Enter the name and the description of the new JMS Source in the *Name* and *Description* fields respectively.
6. Select the JMS Provider activity from the *JMS Provider* drop-down list.

	To learn how to create JMS Provider activity, refer to the section *Creating JMS Provider* in *Administrator Guide*.

7. Select the Connection Type as either Topic or Queue from the *Connection Type* drop-down list.
8. Check the *Durable Subscriber* check box if the JMS Subscriber is durable. 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.
9. Enter the subscriber ID in the *Subscriber ID* field.
10. Enter the name of queue or topic as configured in the JMS Server in the *Queue Or Topic Name* field.
11. If you want to select a specific message from the JMS Server, enter the message selector in the *Message Selector* field.

	The message selector is used to specify the filter criterion to receive a message that the user is interested in. The messages can be filtered based on only header references and properties references of the message. The message selector uses SQL92 query syntax to define the filter criteria. SQL92 is widely used to query the entire standard databases i.e. Oracle, SQL Server. The only difference between the database query and the message selector query is that the message selector uses, only a part of the query which is after the where clause.
	
	The following message selector selects messages with a message type of car and color of blue and weight greater than 2500 pounds:
	
	`JMSType = 'car' AND color = 'blue' AND weight > 2500`
	
	The following message selector selects message with the property Sport has value either as Basketball or Football.
	`Sport in ('Basketball','Football')`

12. Enter the time in seconds in the *Time Out* field. If any message is not received in this interval, process flow execution will be stopped. If Time Out field is left blank, JMS Source activity checks for the specified message and if message is not available, process flow is aborted, without waiting for message.
13. Enter the username and password (if required) to connect to JMS Server in the *User*N*ame* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

14. Click **Save** button. This displays a screen confirming that the JMS source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the JMS source (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the JMS source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the JMS source activity at design time. For this, click **Test**. This verifies the values in the *JMS Provider, Connection Type* and *Queue/Topic Name* fields and checks whether the source activity actually exists in the specified location.

CREATING LAN FILE SOURCE ACTIVITY

The LAN File Source provides the ability to specify a file location that is accessible on the network.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create a LAN File Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **LAN File**. The Manage LAN File Source screen is displayed (see Figure 6.14).

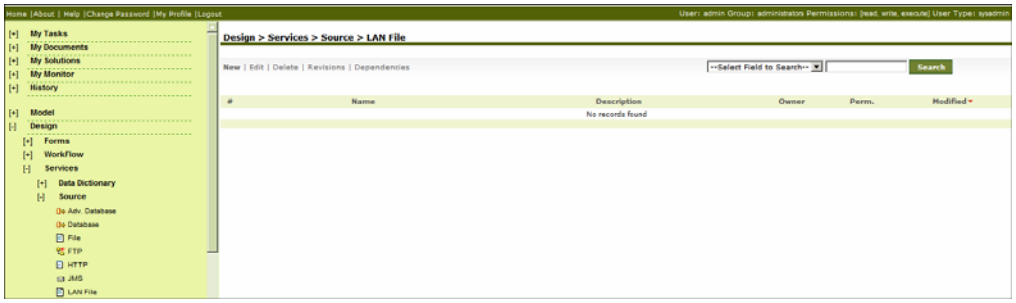


Figure 6.14: Manage LAN File Source

4. Click the **New** link. The Create LAN File Source screen is displayed (see Figure 6.15).

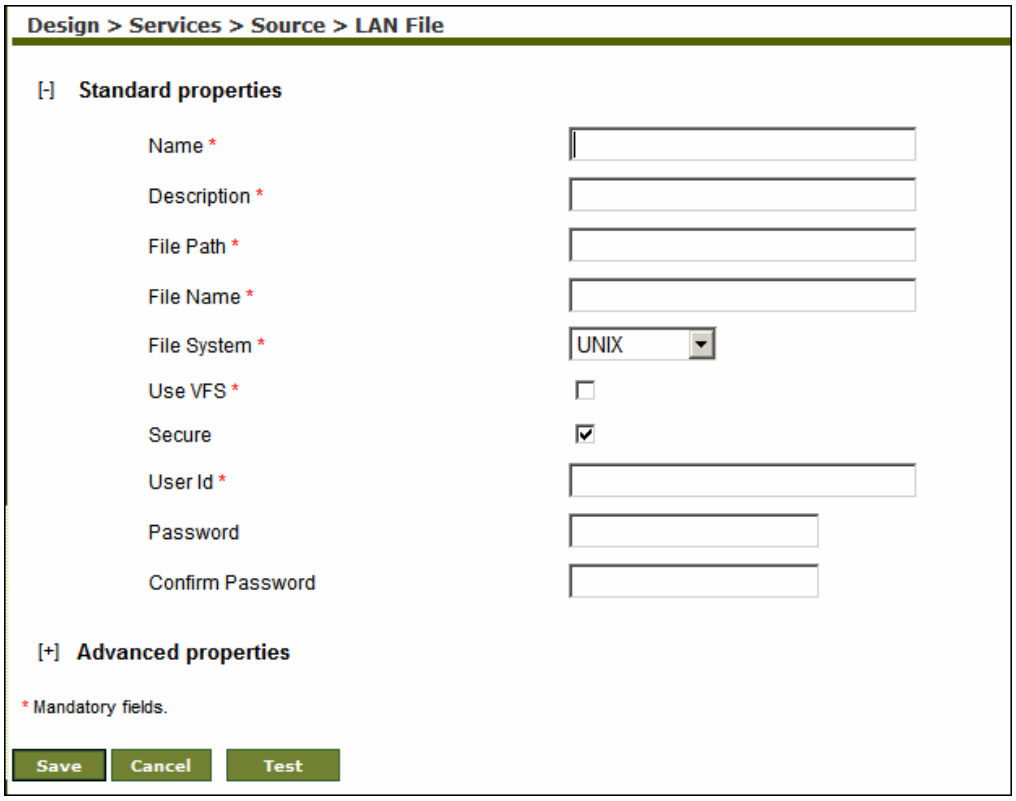


Figure 6.15: Create LAN File Source

5. Enter the name and the description of the new LAN File Source in the *Name* and *Description* fields respectively.
6. Enter the network path of the source file in the *File Path* field in the following format:

*hostname**folder name*

For example, if a file named *sales.txt* is stored in a shared folder employee of a host Server. The File Path will be *Server**employee*\

7. Enter the name of the source file in *File Name* field. For Example Sales.txt.
8. Select the file system whether Windows or Unix from the *File System* drop-down list.
9. When Adeptia Suite is installed on Windows, the LAN File Source uses windows service to connect to remote machine to access any file. It just connects once and uses the same connection with the same User ID and Password (which is stored in the cache) every time. If you want to enforce the validation of User ID and Password every time while accessing the file on a remote machine, check *Use VFS* checkbox.
10. If authentication is required to access the source file, check the *Secure* checkbox.
11. Enter the username in the *User ID* field.

12. Enter the password, if required, in the *Password* field. Then re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section. If you want to delete the LAN source file after it is used in the process flow execution, you can check the *Delete File on Success* checkbox, while defining the Advanced Properties. This selection will delete the LAN source file once the JTA block in which the LAN source file exists, is executed successfully during the process flow execution.

13. Click **Save** button. This displays a screen confirming that the LAN File Source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the LAN file source (refer to Figure 3.6).
14. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

15. Click **OK** to save the comments. This displays a screen confirming that the LAN File Source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the LAN file source activity at design time. For this, click **Test**. This verifies the values in the *File Path, User ID* and *Password* fields. and checks whether the file actually exists in the specified location.

CREATING MAIL SOURCE ACTIVITY

Mail Source provides the ability to specify a file as a source that is accessible via Mail.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a Mail Source Activity

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

3. Click **[+] Source** to expand the tree, and then click **Mail**. The Manage Mail Source screen is displayed (see Figure 6.16).

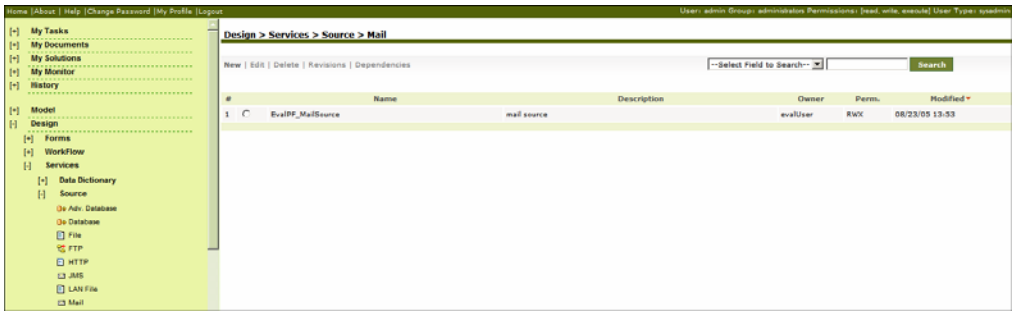


Figure 6.16: Manage Mail Source

4. Click the **New** link. The Create Mail Source screen is displayed (see Figure 6.17).

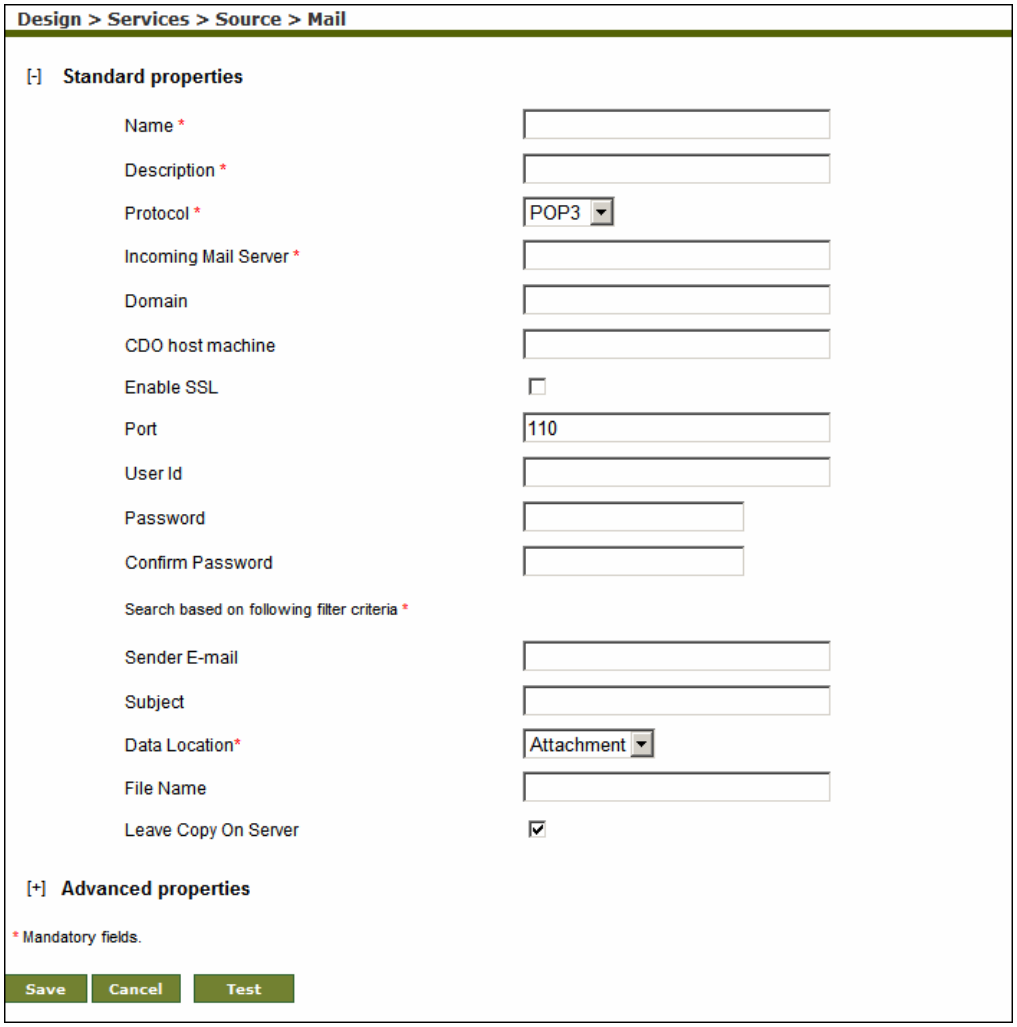


Figure 6.17: Create Mail Source

5. Enter the name and the description of the new Mail Source in the *Name* and *Description* fields respectively.
6. Select the Internet standard protocol to be used for retrieving incoming mails, from the *Protocol* drop-down list. You can select either POP3, IMAP4 or MAPI protocol. Based on the selected protocol, the default port number for that protocol is displayed in the *Port* field.
7. Enter the incoming mail server address in *Incoming Mail Server* field.

To access mails from Microsoft Exchange Server use MAPI in the *Protocol* drop-down list. To connect Adeptia Suite with Microsoft Exchange Server, you need to buy a third party tool called *J-Integra for Exchange*. J-Integra for Exchange is a high performance middleware bridge that enables Java Exchange interoperability. If you want to retrieve mails from an Exchange Server using J-Integra, select *MAPI* in the *Protocol* drop-down list.

If *MAPI* is selected in the *Protocol* drop-down list:

- Enter name of the exchange server in the Incoming Mail Server field
- Enter the domain name in the Domain field
- Enter the name of the CDO host machine in the CDO host machine field. CDOConfig.exe is a tool that comes with the J-Integra for Exchange SDK and is used for configuring CDO. Host where CDO is configured is called CDO host machine.

For detailed information about Jintegra for Exchange, refer to **http://j-integra.intrinsyc.com/products/exchange/**.

8. Check the *Enable SSL* checkbox, if the specified incoming mail server requires a secure connection.
9. The default port number for the selected protocol is displayed in the *Port* field. If you want to change this port number, enter the new port number in the *Port* field.
10. Enter the username and password of Mail Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.
11. Select any of the following filter criteria:
    - Sender E-mail
    - Mail Subject

    You may select more than one filter criteria.
12. Enter the sender's email address and subject of email in the *Sender Email* and *Subject* fields respectively.

13. Select the location of data in the mail whether it is in attachment or in email body from the *Data Location* drop-down list. This is mandatory.
14. If you select an Attachment in the Data Location field, enter the name of the file in the File *Name* field.

	Mail Source activity does not support more than one file attachment.

15. If you want to leave a copy of the mail on the Server, check the *Leave Copy On Server* checkbox.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

16. Click **Save** button. This displays a screen confirming that the Mail source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the Mail file source (refer to Figure 3.6).
17. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

18. Click **OK** to save the comments. This displays a screen confirming that the Mail source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the mail source activity at design time. For this, click **Test**. This verifies the values in the In*coming Mail (POP3) Server, Port, User ID* and *Password* fields and checks whether the source activity actually exists in the specified location.

	When a mail event triggers process flows, each process flow uses a mail source. At times, when multiple process flows use a mail source, errors can occur. In such a case, you can retry the action before exiting the mail source. You can set the number of retries and the sleep time between each retry, in case an error occurs while using a mail box. To change the settings, refer to the *abpm.mailEvent.retry* property in the *Administrator Guide*.

CREATING WEBDAV SOURCE ACTIVITY

The WebDAV Source activity provides the ability to specify files that is stored in a WebDAV Server, as a source.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create WebDAV Source Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Source** to expand the tree, and then click **WebDAV**. The Manage WebDAV Source screen is displayed (see Figure 6.18).

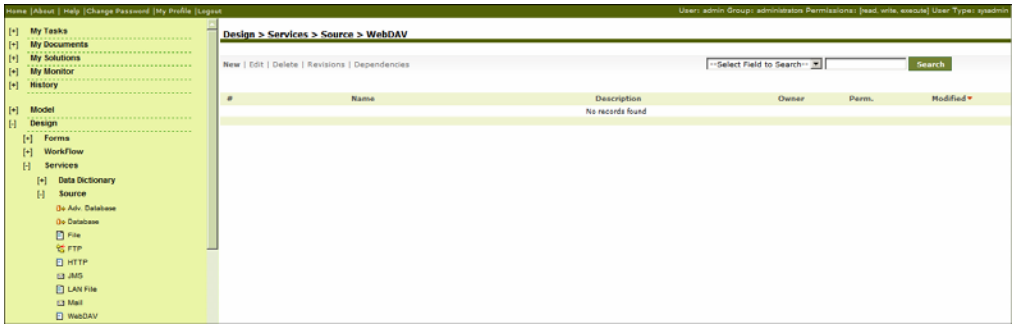


Figure 6.18: Manage WebDAV Source

4. Click the **New** link. The Create WebDAV Source screen is displayed (see Figure 6.19).

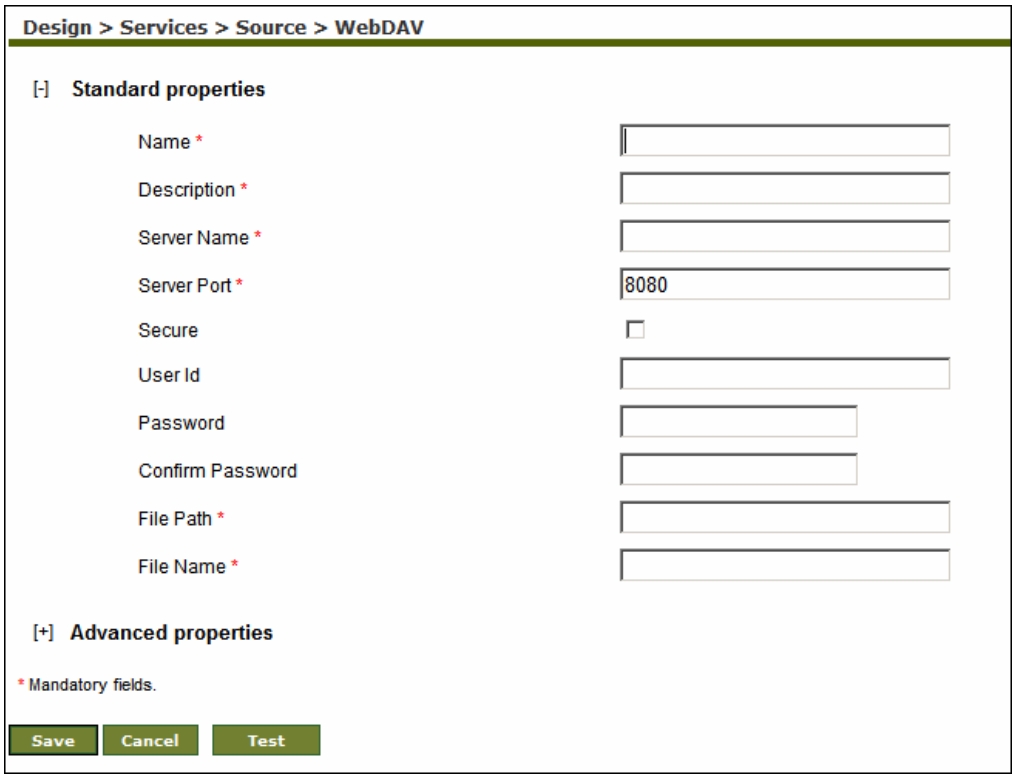


Figure 6.19: Create WebDAV Source

5. Enter the name and description of the new WebDAV Source in the *Name* and *Description* fields respectively.
6. Enter the name of the WebDAV Server and Server port on which WebDAV Server is running, in the *Server Name* and *Server Port* fields respectively.
7. If the WebDAV is secured i.e. username and password are required to access it, then check the *Secure* checkbox and enter the username and password of the WebDAV Server in the *User ID* and *Password* fields respectively. Re-enter the password in the *Confirm Password* field.

	If you are using WebDAV Server, which is built in with Adeptia Suite, the default Username is “Administrator” and the password is “indigo”.

8. Enter the source file with full path in the *File Name* (full path) field. For example, */slide/files/Documents/File.txt,* where Documents is name of the folder and *File.txt* is the name of the file.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. This displays a screen confirming that the WebDAV source activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the WebDAV source (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the WebDAV source activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the WebDAV source activity at design time. For this, click **Test**. This verifies the values in the *Server Name* and *Server Port* fields and checks whether the source activity actually exists in the specified location.

7 CREATING SCHEMA ACTIVITY

Schema is the structure of a file format and it specifies information about different data fields and record types that a message or a data file may contain. Designing a schema is the process of providing metadata information. Schema can be used both at the source end and the target end. At the source end, it is used to read data from the source file and at the target end, it is used to write data to the target file. The Adeptia Suite provides following types of Schema activities:

- **Advance Database Schema**
- **Advance Positional Schema**
- **Advance Text Schema**
- **Database Schema**
- **Excel Schema**
- **Positional Schema**
- **Text Schema**
- **Word Schema**
- **XML Schema**
- **XML Validator**
- **EDI Schema**

Adeptia Suite allows you to create schema in two ways:

- **Using Definition File**
- **Entering the Fields Sequentially**

Using Definition File

You can create a schema using a Definition File in three ways:

- **Using Data File**
- **Using Field File**
- **Using XSD File**

These methods may vary across different schemas. Their compatibility with the schemas are outlined in the table below.

Table 7.1: Definition File Methods used for Creating Schemas

Schema	Data File	Field File	XSD File
Advance Database Schema	√		√

Advance Positional Schema		√	√
Advance Text Schema		√	√
Database Schema			√
Excel Schema	√	√	√
Positional Schema		√	√
Text Schema	√	√	√

Using Data File

Data file is the file, which contains the actual data used as source or target during execution of a process flow. It could be same file, which is used in the process flow or another sample file of same format.

Using Field File

Field file is Comma Separated Values (CSV) file, which contains names of the fields and their definition, separated by comma. This option is helpful in case the number of fields in source or target data is very large. All the field names are picked up from this CSV file. Each line of the CSV file should contain one field name and its definition separated by comma. If the data type is *Date*, format of the date must be specified after data type, separated by comma.

Incase a field is defined as *Date* type and date format and time format is not defined, the default date format will be mm/dd/yy and time format will be blank.

When copying a field file, you need to verify that the field format is correct and there are no extra lines in the document. Else, an error is generated when converting to HTML.

Field File format for Advance Positional Schema

The Field file format for Advance Positional Schema can be of two types:

- **Based on Start Position and End Position**
- **Based on Field Length**

Field File format for Advance Positional Schema based on Start and End Position

<Record Identifier1>:<Value>,<Record Identifier2>:<Value>
<RecordIdentifier Value>,<FieldName>,<Description>,<DataType>,[DateFormat],[TimeFormat],<Start Position>,<EndPosition>,<Alignment>,<Skip>

In case of Advance Positional Schema, Record Identifier and Value should be specified at the beginning of the CSV file as displayed below:

```
a:first,b:second
first,a,first_field,string,,,1,11,L,F
first,name,name_of_employee,string,,,12,21,L,T
first,empid,employee_ID,int,,,22,36,L,F
second,b,second_field,string,,,1,11,L,F
second,DOB,date
birth,date,yyyy/dd/MM,HH:mm:ss,12,24,L,F
second,Address,Address
employee,string,,,25,44,L,T
```

where: **L** means left alignment

R means right alignment

T means True

F means False

Field File format for Advance Positional Schema based on Field Length

<RecordIdentifier Value>,<FieldName>,<Description>,<DataType>,[DateFormat],[TimeFormat],<Length>,<Alignment>,<Skip>

In case of Advance Positional Schema, Record Identifier and Value should be specified at the beginning of the CSV file as displayed below:

```
a:first,b:second
first,a,first_field,string,,,11,L,F
first,name,name_of_employee,string,,,10,L,T
first,empid,employee_ID,int,,,15,L,F
second,b,second_field,string,,,11,L,F
second,DOB,date
birth,date,yyyy/dd/MM,HH:mm:ss,13,L,F
second,Address,Address
employee,string,,,20,L,T
```

where: **L** means left alignment

R means right alignment

T means True

F means False

Field File format for EDI Schema

<Field Separator>
<Segment Identifier>,<Field Name>,<Description>,<ID>,<Minimum Size>, <Maximum Size>

Field Separator should be specified at the beginning of the CSV file. Following is the content of sample CSV file used to create EDI schema:

```
~
ISA,Name,Employee Name,1,2,2
ISA,Age, Employee Age,2,10,10
ISA,DOB,Date of Birth,3,2,2
ISA,MaritalStatus,Employee
Marital Status,4,10,10
```

Field File format for Excel and Text Schema

<Field Name>,<Data Type>,[Date Format],[Time Format]

Following is the content of sample CSV file used to create Text and Excel schema:

```
NAME,string,,
PHONE_NO,int,,
DOB,date,MM/dd/yy,hh:mm:ss
DOJ,date,MM/dd/yy,
```

Field File format for Positional Schema

Field file format for Positional Schema can be of two types:

- **Based on Start Position and End Position**
- **Based on Field Length**

Field File format for Positional Schema based on Start and End Position

<Field Name>,<Description>,<Data Type>,[Date Format],[Time Format],
<Start Position>,<End Position>,<Alignment>,<Skip>

Following is the content of sample CSV file used to create Positional schema:

```
name,name of employee,string,,,1,10,L,F
empid,employee ID,int,,,11,30,L,T
DOB,Dat of birth,date,yyyy-dd-MM,HH:mm,31,60,L,F
```

where: **L** means left alignment
R means right alignment

T means True
F means False

Field File format for Positional Schema based on Field Length

<Field Name>,<Description>,<Data Type>,[Date Format],[Time Format], <Length>,<Alignment>,<Skip>

Following is the content of sample CSV file used to create Positional schema:

```
name,name of employee,string,,,10,L,F
empid,employee ID,int,,,20,L,T
DOB,Dat of birth,date,yyyy-dd-MM,HH:mm,30,L,F
```

where: **L** means left alignment
R means right alignment

T means True
F means False

Using XSD File

XML Schema Definition (XSD) file describes the elements in an XML document. XSD file used to create the schema must be compliant to the Adeptia Suite format. To get the Adeptia Suite compliant XSD, you can edit any existing schema and download the XSD file. You can further edit the field in XSD file and use it to create the schema. For example you already created a schema with 100 fields and you want to create another schema with same 90 fields out of 100. You can download the XSD file from existing schema, delete 10 additional field by editing the XSD file and use that XSD file to create another schema.

Entering Fields Sequentially

This is manual way of creating the schema. If you select this option, you have to enter the field's name, their data type manually in correct sequence.

When creating a schema (except XML schema), a *Record Number* attribute is automatically created at the record level. It is available for each record. If the schema is used at the source level, then this attribute will be populated in the intermediate XML file at the record level. It always starts at1. If an error is detected, then this attribute will be generated in the Error File. For example, if error is found at record number 5 in the source file, then Record Number 5 is displayed in the Error File.

CREATING ADVANCE DATABASE SCHEMA ACTIVITY

The Advance Database Schema activity is created to define how the data from multiple tables can be obtained or inserted into selected tables. The Advance Database schema uses the predefined *Database Info* to connect to the database.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity must be created before creating A*dvance Database Schema* activity.

Steps to create the Advance Database Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Adv. Database**. The Manage Adv. Database Schema screen is displayed (see Figure 7.1).

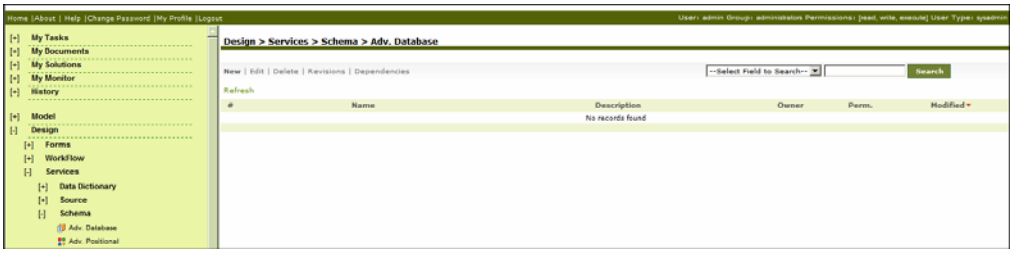


Figure 7.1: Manage Advance Database Schema

4. Click the **New** link. The Create Advance Database Schema screen is displayed (see Figure 7.2).

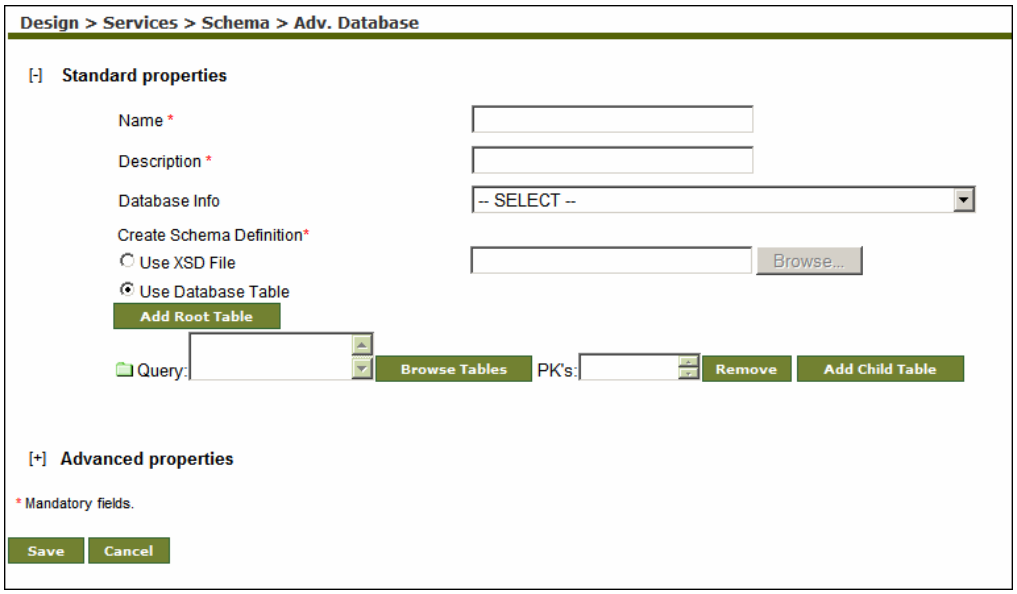


Figure 7.2: Create Advance Database Schema

5. Enter the name and description for new Advance Database Schema in *Name* and *Description* fields respectively.

6. Select the database info activity from the *Database Info Id* drop-down list.

To learn how to create Database Info activity, refer to the *Creating Database Info* section in *Administrator Guide*.

7. To define schema definition, select one of the following options:
   - Use XSD File
   - Use Database Table

8. To select the XSD file, which contains schema information, select the *Use XSD File* radio button and click Browse.

9. To define schema using database table, select the Use *Database Table* radio button and click **Browse Tables**. The Select Schema screen is displayed with the list of RDBMS Schema in case of *SQL Server* and *DB2* Database Info (see Figure 7.3).

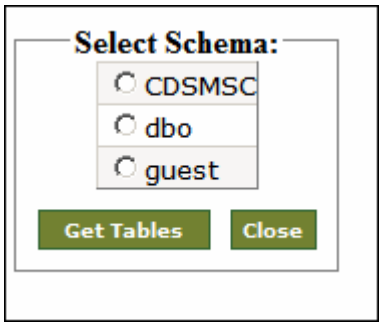


Figure 7.3: Select Schema

10. Select the required RDBMS schema and click **Get Tables**. The Select Table screen is displayed with the list of tables (see Figure 7.4).

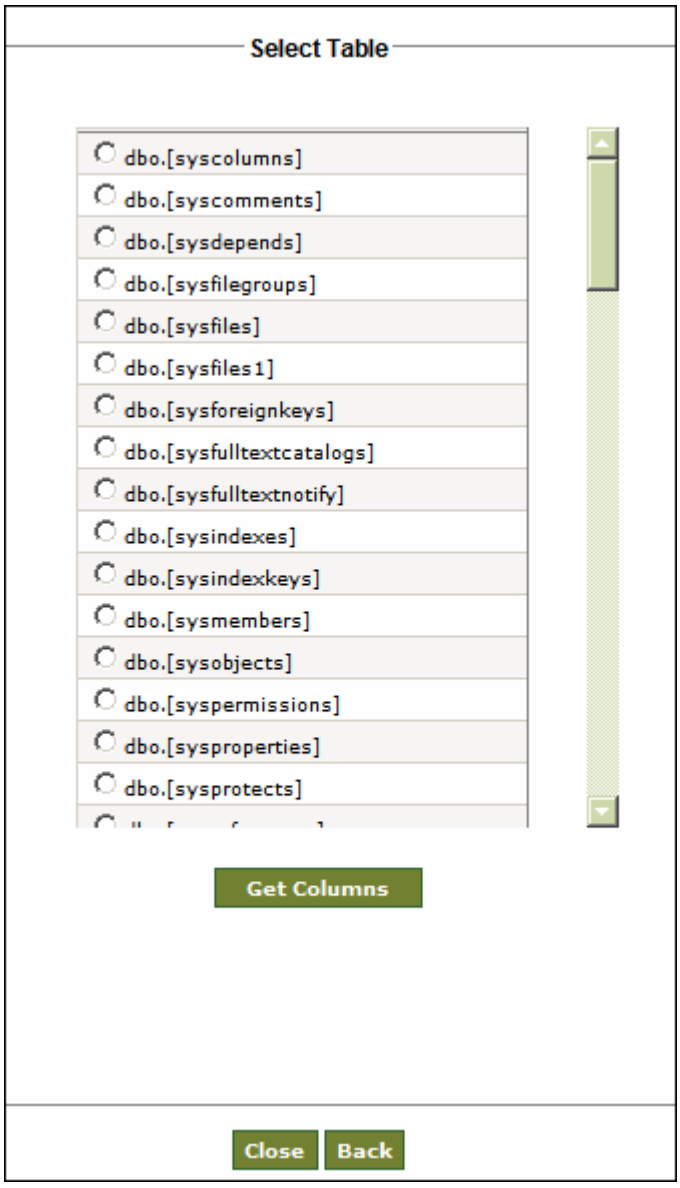


Figure 7.4: Select Tables

	A **Close** button appears on the Select Table screen, in case of SQL Server or DBO Database Info. Clicking this button takes the control to the Select Schema screen. If the schema and its tables are created on DB2, then you need to remember that the schema name in DB2 is case-sensitive. Else, it will display the same table more than once.

11. Select the required table and click **Get Columns** button. The following screen is displayed with the list of columns and their data types (see Figure 7.5).

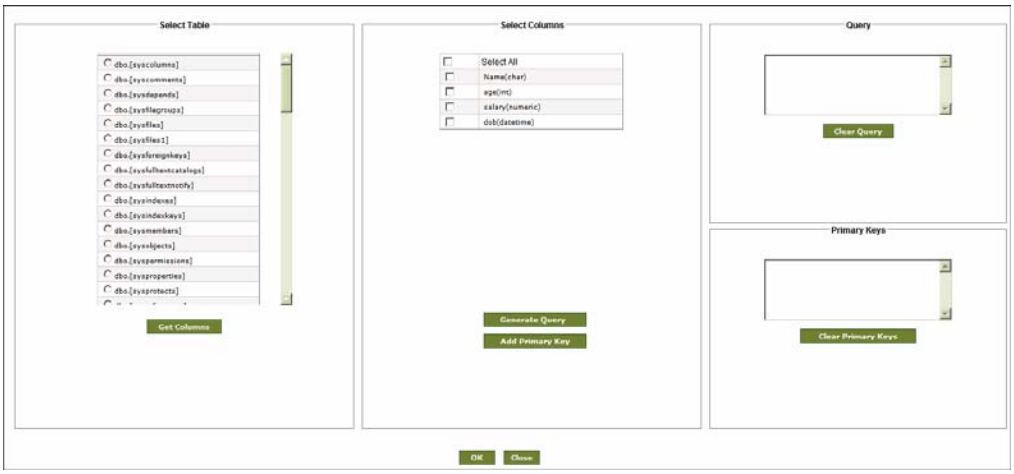


Figure 7.5: Select Columns and Primary Key

12. In *Select Columns*, select the required column(s) and click **Generate Query** button to generate the database query. The generated query is displayed in the *Query* field. You can also specify the where clause within the generated select query (see Figure 7.6).

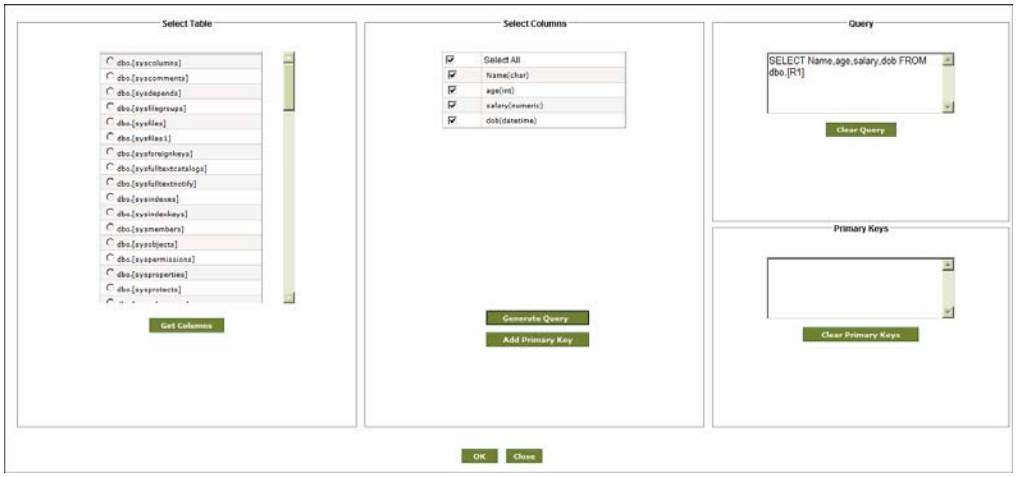


Figure 7.6: Generate Query

	There should not be any special character (except $ and SID#) in column name.

	▪ This query is validated, once you click the **Save** button. ▪ In case some SQL function is used or some calculation is done over the Column Name(s), use the alias name for that Column Name(s). For example, for a query like: *SELECT SID,Name+Dept,Salary FROM dbo.Employee* You can use: *SELECT SID, (Name+Dept) as **AliasName,Salary** FROM dbo.Employee*

	Reason: While getting the result set, you have the assign the output of (Name+Dept) to some new field.

13. To designate a column as Primary Key, select the required column(s) and click **Add Primary Key**. The selected column is displayed in the *Primary Key* field.

	At times, you may not use the Primary Key in the query, but append it internally in the query. You need to ensure that the Primary Key name should be the same at both the places. Else, the schema will fetch the data twice for the Primary Key column and result in errors.

14. Click **OK** to return to the Create Advance Database Schema screen. The selected query and primary key are displayed in their respective fields.

15. To add the child table of the selected table, click **Add Child Table** in the Create Advance Database Schema screen. A child table is displayed under the root table (see Figure 7.7).

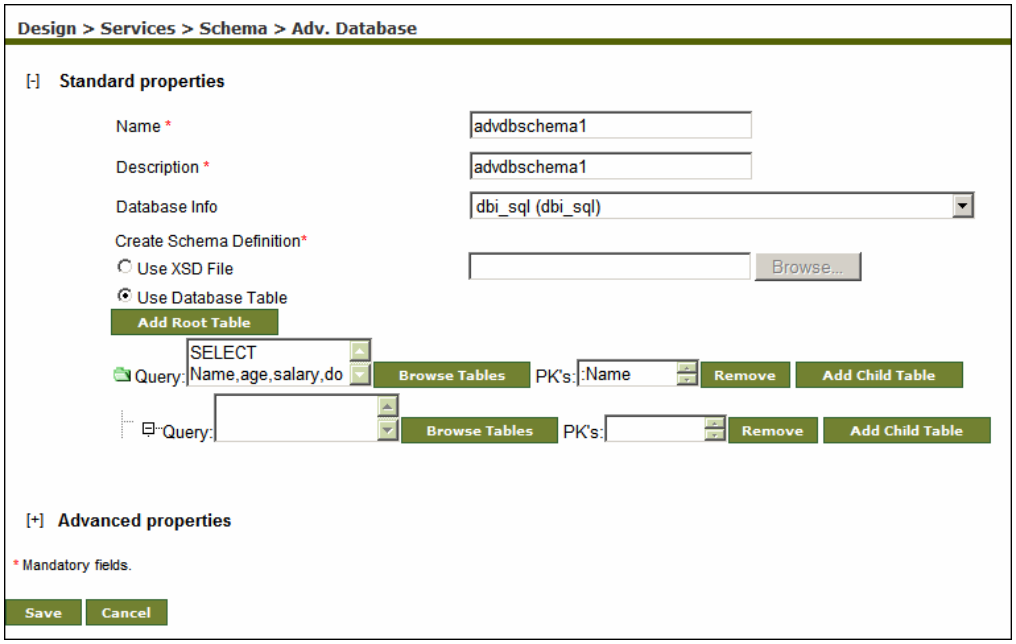


Figure 7.7: Create Child Table

16. Click **Browse Tables** in the child table. The Select Schema screen is displayed with a list of RDBMS Schema in case of SQL Server and DB2 Database Info (refer to Figure 7.3). If HSQLDB Database Info is selected, then the Select Table screen is displayed (refer to Figure 7.4).

17. In the Select Schema screen, select the required RDBMS schema and click **Get Tables**. The Select Table screen is displayed with the list of tables (refer to Figure 7.4).

	A **Close** button appears on the Select Table screen, in case of *SQL Server* or *DBO* Database Info. Clicking this button takes the control to the SSelect Schema screen.

18. Select the required table and click **Get Columns**. The following screen is displayed (see Figure 7.8).

Figure 7.8: Select Column, Primary and Related Keys

19. In Select Columns, select the required column(s) and click **Generate Query** button to generate the database query. The generated query is displayed in the *Query* field. You can also specify the where clause within the generated select query.

	A column name should not include any standard SQL keyword (for example, *Identity*), as it generates an error, when the Advance Database schema is used as a source or target database.

20. To designate a column as a Related Key, select the required column(s) and click **Add Related Key**. The selected column is displayed in the *Related Key* field (see Figure 7.9).

Figure 7.9: Add Related Key

21. To change the Primary Key, select the required column(s) and click **Add Primary Key**. The selected column is displayed in the Primary Key field.

22. Click **OK** to return to the Advance Database Schema screen.
23. To add another independent database table, click **Add Root Table** and repeat the steps from 8 to 13. Alternately, you can create a root table from the Create Advance Database Schema screen by clicking **Add Root Table**.

	▪ At the Root level, the same table should not be used more than once. ▪ To learn about Advanced Properties refer to **Changing Advanced Properties** section. ▪ There are some Unicode Characters, which are reserved characters in XML. You can filter these characters by enabling Filter Unicode Character option.To enable Filter Unicode Character option, mark the *Filter Unicode Character* checkbox as checked in Advanced Properties.

24. Click **Save** button. This displays a screen confirming that the Advance Database Schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the Advance Database schema (refer to Figure 3.6).
25. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

26. Click **OK** to save the comments. This displays a screen confirming that the Advance Database Schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Special Usage Scenario

Delete Records from target table

In Advance Database schema, you can set action attribute to delete the records from the target database table, if they are matching with the records of source database table. Source and target records are matched based on primary key. This attribute is enabled in Data Mapper, while mapping source and target schemas.

Steps to set action attribute to delete matching records

Load required source and target schema in Data Mapper. In target schema, there will be an *Action* attribute.

1. Create a constant '*delete*' and map it to *Action* attribute of the target schema.

	In order to delete records where the target end database is ORACLE, IBM DB2, MS SQL server or HSQLDB, the keyword "*delete*" should be used. However for MS SQL server DBMS, the "*cancel*" keyword can also be used. "*cance*l" keyword cannot be used for other RDBMS except Ms SQL server.

2. Save the Mapping activity.

	To learn how to use Data Mapper, refer to section **Using Data Mapper**.

Enable RollBackOn Error Attribute

In Advance Database schema, you can set the *RollBackOnError* attribute. To roll back data updation or insertion, in case any error is encountered in the target database, you need to enable *RollBackOnError* attribute. This attribute is enabled in Data Mapper, while mapping source and target schemas.

Steps to Enable RollBackOnError attribute

1. Load required source and target schema in Data Mapper. In target schema, there will be a *RollBackOnError* attribute.
2. Create a constant 'true' and map it to RollBackOnError attribute of the target schema.
3. Save the Mapping activity.

	Suppose there is an hierarchy as: R1 - R1C1 - R1C1C1 If *RollbackOnError* is applied on *R1C1*, then in case of error condition in *R1C1* or *R1C1C1*, rollback will be up to *R1C1* only. *R1* will be unaffected. Error count will also be shown according to this implementation. To learn how to use Data Mapper, refer to section **Using Data Mapper**.

CREATING ADVANCE POSITIONAL SCHEMA ACTIVITY

The Advance Positional schema activity defines the procedure to read data from a multiple record format positional file and write data in a multiple record format. To do so, user needs to specify the names and positions of the required fields in order to enable identification of those fields.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create Advance Positional schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Adv. Positional**. The Manage Adv. Positional Schema screen is displayed (see Figure 7.10).

Figure 7.10: Manage Advance Positional Schema

4. Click the **New** link. The Create Advance Positional Schema screen is displayed (see Figure 7.11).

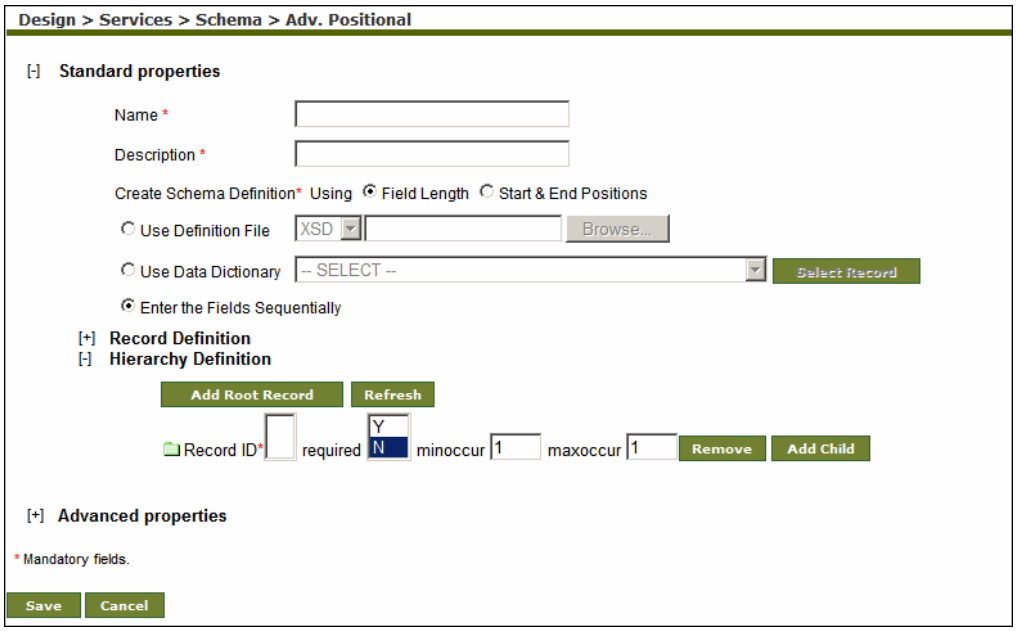


Figure 7.11: Create Advance Positional Schema

5. Enter the name and description for Advance Positional Schema in the *Name* and *Description* fields respectively.

6. You need to create the schema definition for the Advance Positional Schema. You can define the schema in three ways. These are outlined as:

   - Use Definition file
   - Use Data Dictionary
   - Enter the Fields Sequentially

7. To define the schema using definition file, select the *Use Definition File* radio button, select the type of file from the drop-down list and click **Browse** button to select the required file.

8. To define the schema using a data dictionary, click the *Use Data Definition* radio button, select the name of the data dictionary from the drop-down list and click **Select Record** button. This displays the Select Record screen (see Figure 7.12).

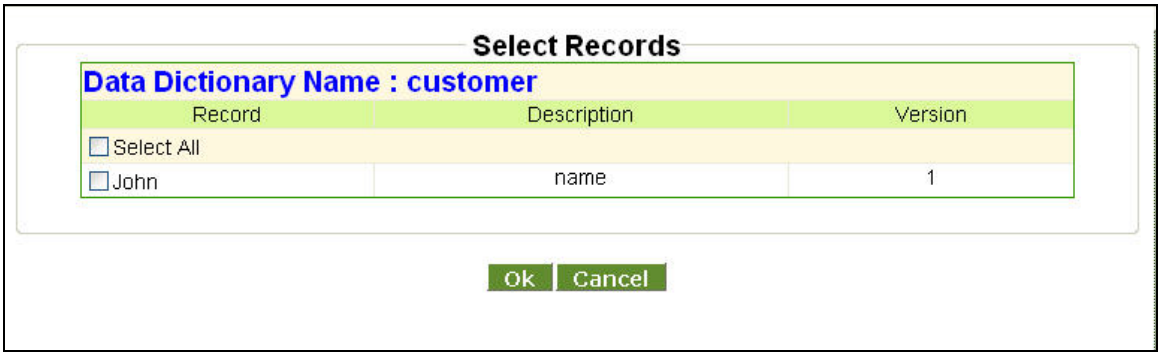


Figure 7.12: Select Data Dictionary

	To know, how to create Data Dictionary, refer to the Creating Data Dictionary section.

9. This screen displays the name of the selected data dictionary and a list of all the records defined in the dictionary. Check the checkbox against the record that you want to display in the Advance Positional schema.

	To display all records, check the *Select All* checkbox.

10. Click **OK**. This closes the Select Record screen and displays the selected records under Record Definition on the Manage Advance Positional Schema screen (see Figure 7.13). You can view or delete a record from this screen by clicking the appropriate button for that record.

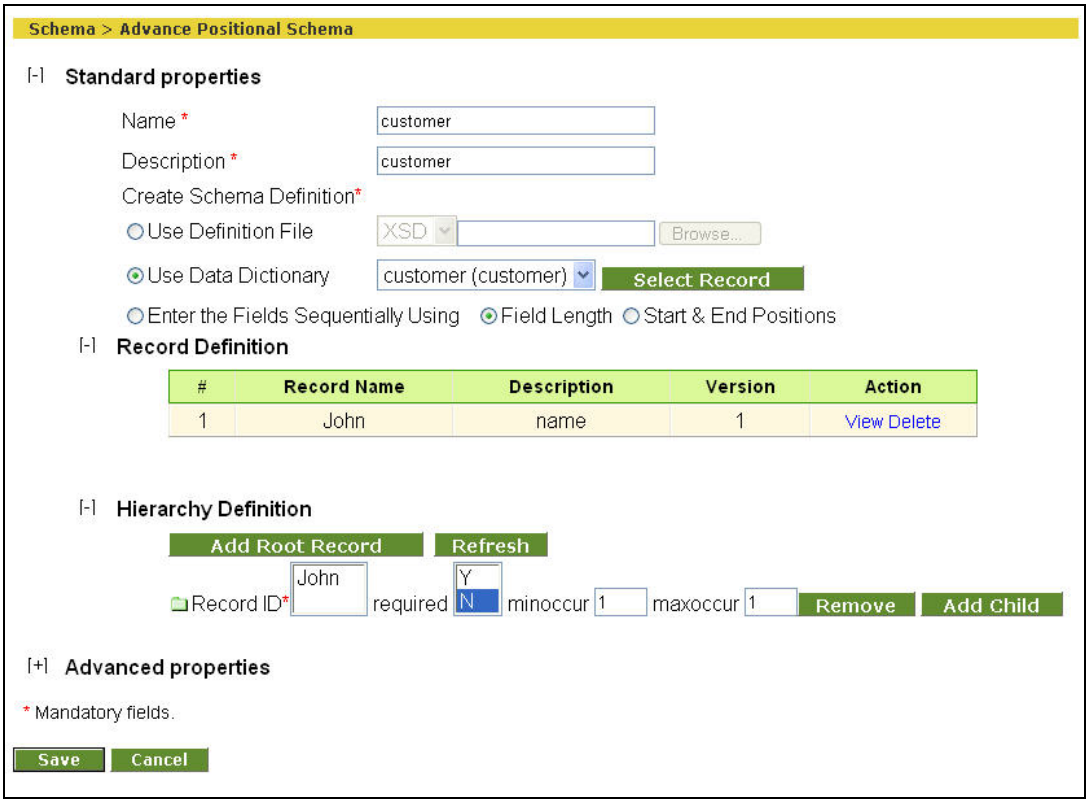


Figure 7.13: Create Record Definition using Data Dictionary

11. To enter the fields manually, select the *Enter the Fields Sequentially Using* radio button and click **(+) Record Definition**. This expands to display the Record Definition fields (see Figure 7.14).

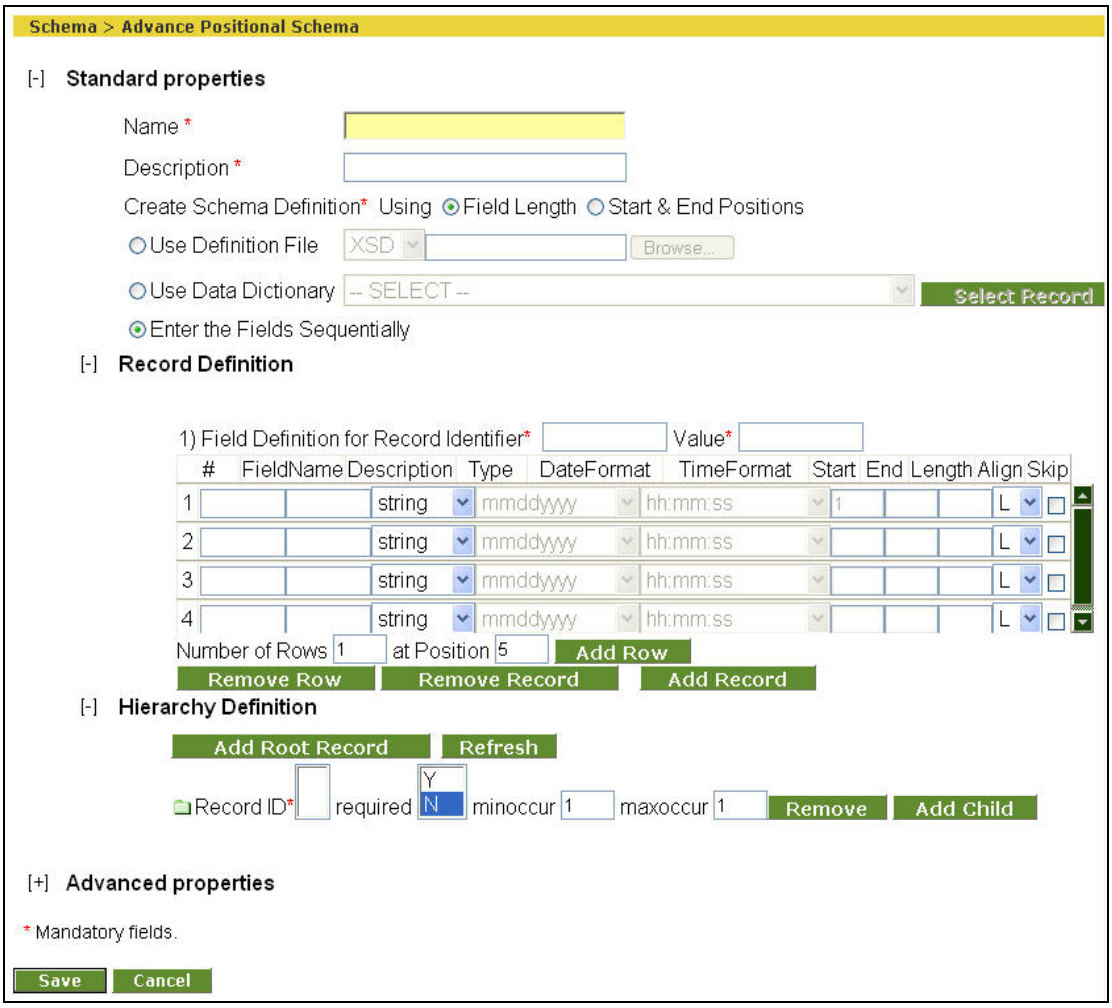


Figure 7.14: Create Record Definition

12. Enter name and description of the field in *FieldName* and *Description* fields respectively.
13. Select the type of data from *Type* drop-down list. For data type selection, refer to Table 7.2.
14. If data type is *Date*, select the format of date and time from the *DateFormat* and *TimeFormat* drop-down lists respectively.
15. To define field position select one of the following options:
    - Field Length
    - Start & End Positions
16. To define the field position using field length, select *Field Length* radio button and enter the length of the field in the *Length* field.
17. To define the field position using start and end position, select the *Start & End Positions* radio button.
18. Enter the start position of the field in the *Start* field.

19. Enter the end position of the field in the *End* field.

	The starting position of a row in a positional file is 1. In a positional file, tab is counted as one position and not eight positions. By default, field positions are created in sequence. You can also create a schema with fields that are not in sequence. For details, refer to the Defining Field Positions Non-Sequentially section.

20. Select the alignment of the field from the *Align* drop-down list.

	From **Align** select **L** if the field is left aligned. **R** if the field is right aligned. To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. Maximum 99 rows can be added at a time.

21. Check the *Skip* checkbox if you want to skip this field while generating the XML. This selection skips the fields that are not required for the schema. For example, the source file has over 1500 fields, but you just need to use 1000 fields. This selection skips the 500 unrequired fields, and does not read them, when the data is parsed to the XML. When the data file is created, the skipped fields are displayed in the file but are not read. If the schema is created using an existing XSD, the skipped fields will appear as a blank value. However, when the schema is used in other activities such as Mapping, all its fields are displayed.

	The skipping of unrequired fields is useful in case of a standard XSD with a large number of fields, as it reduces the size of the generated XML which now contains only the required fields. While editing the schema, when the data file is downloaded, the skipped fields are represented by 'T' and the unskipped fields by 'F' (see Figure 7.15). Similarly, while viewing the Print-Friendly Page, the skipped fields are represented by 'T' and the unskipped fields by 'F'.

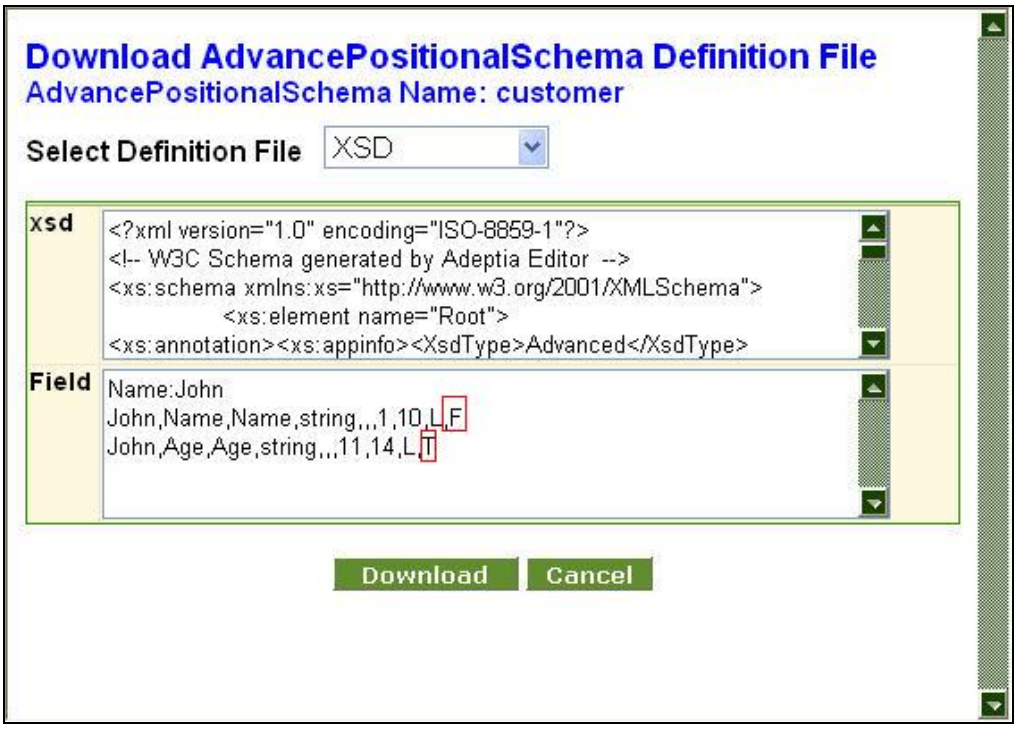


Figure 7.15: Skipped and Unskipped Fields in the Data File

22. Enter the name of any field in *Record Identifier* field and the value of the field in the *Value* field.
23. Click **Add Record** to define another set of data and repeat the steps from 6 to 20.
24. After defining records you need to define their hierarchy. This is mandatory for creating an Advance Positional schema.
25. Once you have added the records and defined the Record Identifiers and their values, click the **Refresh** button under **Hierarchy Definition**. This populates the Record Identifiers of the defined records, in the Record ID field under Hierarchy Definition (see Figure 7.16).

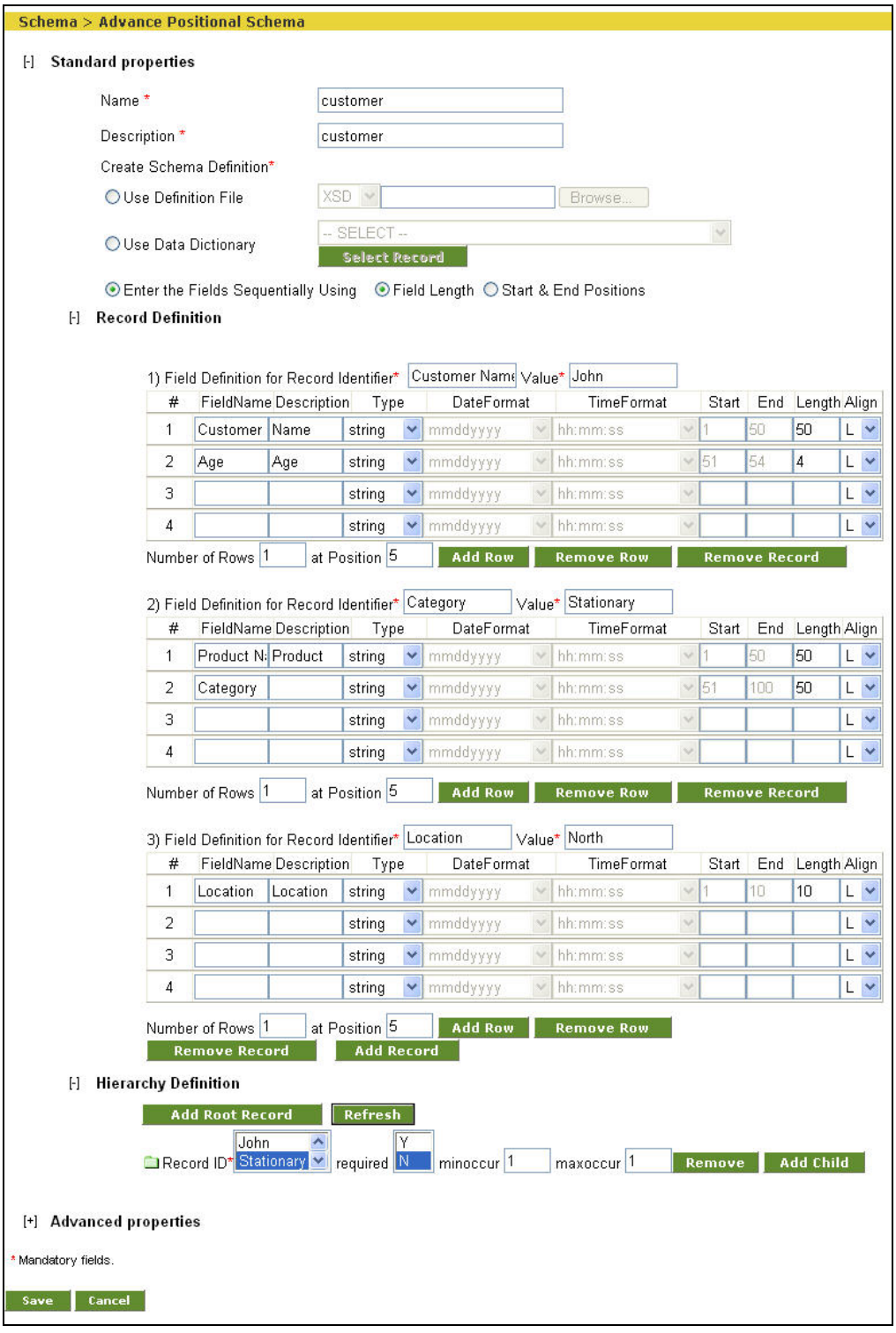


Figure 7.16: Create Record Hierarchy Definition

26. Select a record from the *Record ID* listbox.

27. Select *Y* or *N* from the required listbox, to indicate whether the selected record needs to present in the source file.
28. Enter the minimum number of occurrences of the selected record required in the source file, in the *minoccur* field. This value is based on the selection in the required field. If it has the value as 'Y', then the minoccur field should have a value as a minimum of 1. If the value is 'N', then the minoccur field can be 0.
29. Enter the maximum number of occurrences of the selected record required in the source file, in the *maxoccur* field. It should be a maximum of 2147483647.
30. You can now create a hierarchy either at the root level or at the child level.

Using Hierarchy at Root Level

To use a hierarchy at the Root Level

1. Click **Add Root Record** button. This creates a record at the same level as that of the displayed record (see Figure 7.17).

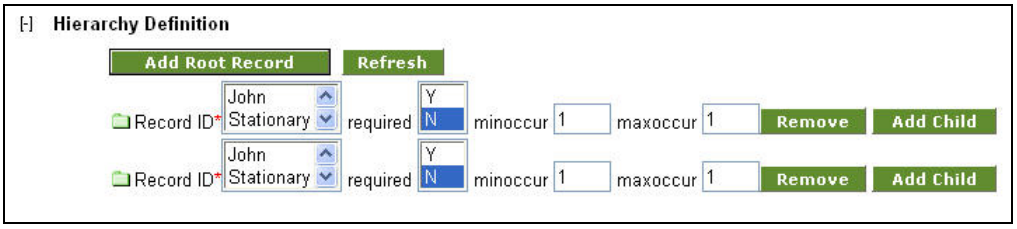


Figure 7.17: Create Root Record

2. Select a record from the Record ID field of the displayed record and enter all its related information.
3. Similarly, select a record from the *Record ID* listbox of the root record and repeat step 25-26 to create another root record.

	Every record must have a unique Record ID. For example, if *John* is selected as the first root record, then you need to select *Stationary* in the next root record.

Using Record at Child Level

To use a record at the Child Level

1. Click **Add Child** button. This creates a record at a level below that of the displayed record (see Figure 7.18).

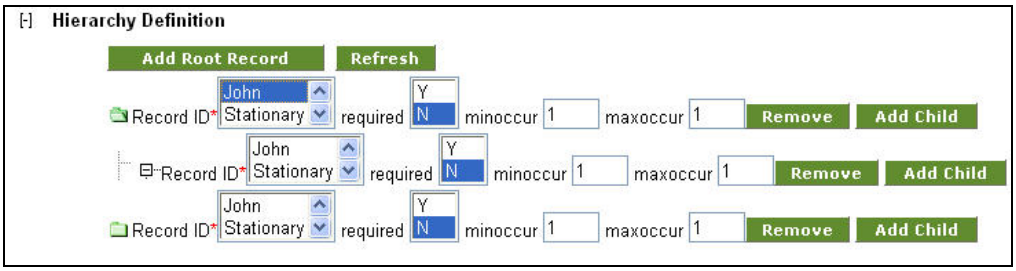


Figure 7.18: Create Child Record

2. Select a record in the *Record ID* field of the displayed record and enter all its related information.

3. Select a record from the *Record ID* listbox of the root record and repeat step28-29 to create a child record.

	A parent and child record must have a unique Record ID in one hierarchy. For example, if *John* is selected as the parent record, then you need to select *Stationary* as the child record. Similarly, if you create another child record under *Stationary*, then you need to select *North* as its Record ID (see Figure 7.19).

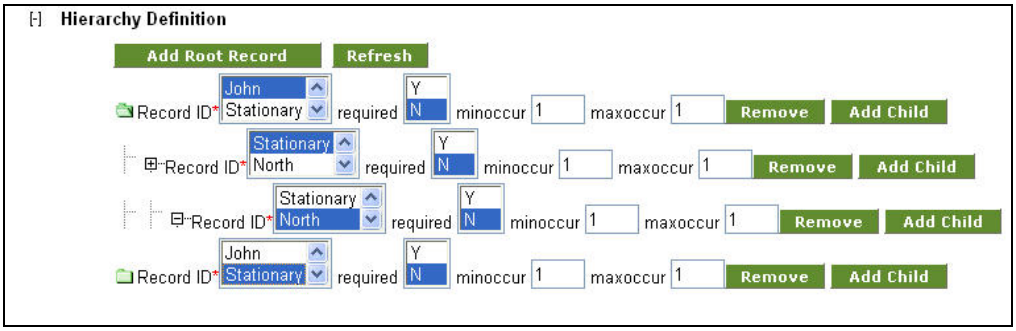


Figure 7.19: Create another Child Record

	You can remove a record by selecting the record and clicking **Remove**. This will display a confirmation delete message, where on clicking **OK**, will delete the record. Alternately, if you delete the last fieldname of a record, and then save the schema, the entire field gets deleted. For example, a record has three fields – *Name*, *Description* and *Age*. If you delete *Age*, the entire field will get deleted. If a record has one or more child records, deleting the parent record will delete all its child records too.

4. Click **[+]** to expand **Advanced Properties**. Advanced properties of advance positional schema are displayed (see Figure 7.20).

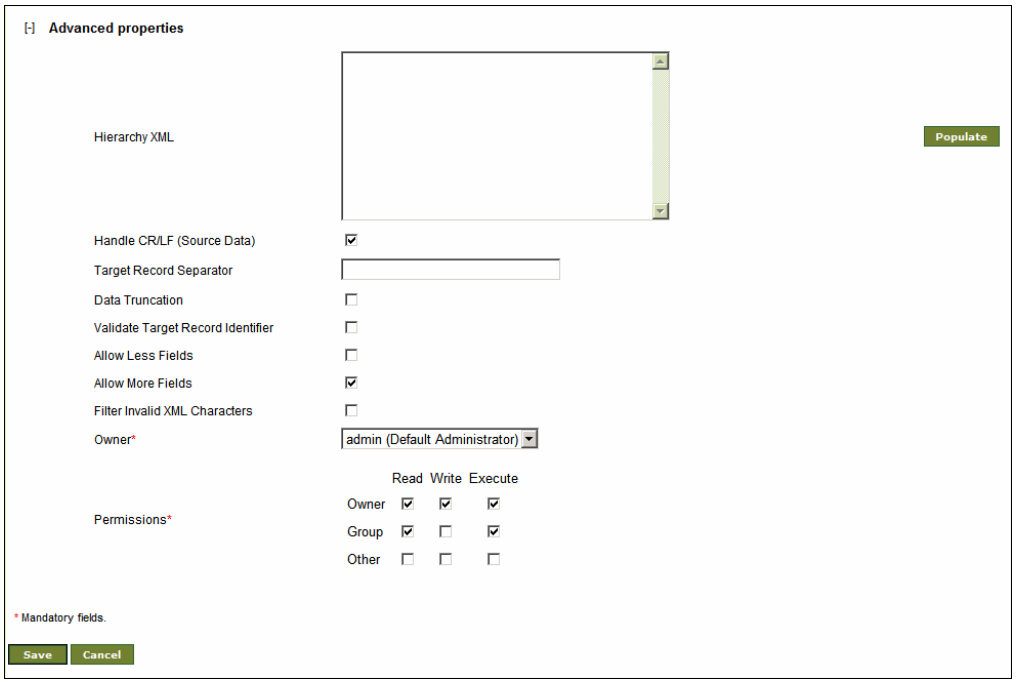


Figure 7.20: Change Advanced Properties

5. Click **Populate** to display the XML code for the defined hierarchy. This field is editable. In case there are numerous records, you can create a flat hierarchy and then edit this xml code to change to the desired hierarchy. If you edit or enter new XML code for the hierarchy, then it will override the existing hierarchy.

It is recommended that you edit this xml code in another editor by copying it to the editor and then making the changes. Once you are done with the changes, you can paste it into the *Hierarchy xml* field. Once you have pasted the xml code in this field, you should not click the **Populate** button again or make changes to the Hierarchy, as it will replace the edited xml with the original xml code.

6. Uncheck the *Handle CR/LF (Source Data)* checkbox, if the source file does not have any carriage return. By default, this option is checked, and schema expects file with carriage return. Handle CR/LF (Source Data) option is applicable only for the schema used at the source end.
7. Enter the record separator for target records, in the *Target Record Separator* field.
8. Check the *Data Truncation* checkbox, in case the data length is more than specified in the schema and you want to pass the specified length of data and ignore the rest of the data.
9. Check the *Validate Target Record Identifier* checkbox, if the value of the target record identifier in XML must match with the value specified in the schema. By default, this checkbox is unchecked. It means that if the Record Identifier Value is not exactly same, then it does not generate error records.

10. Check the *Allow Less Fields* checkbox, if you want to parse the data even if the number of fields in the data file is less than the number of field specified in the schema. If Allow Less Fields checkbox is checked and the schema is used at source end, schema will parse the input data and insert the empty tag of missing fields. If the schema is used at target end, it will write all the tags coming in input XML.
11. Incase number of fields in source data is more than the number of fields specified in the schema, only those fields are parsed, which are specified in schema. Other fields are ignored. If you want to generate error records, when number of fields in source data is more than the number of fields specified in schema, uncheck the *Allow More Fields* checkbox.
12. In case the input data contains some characters that are invalid in XML, then this may result in the mapping getting aborted. You can filter these invalid XML characters by checking the *Filter Invalid XML Characters* checkbox.

To learn about other Advanced Properties refer to **Changing Advanced Properties** section.

13. Click **Save** button. This displays a screen confirming that the Advance Positional schema activity has been created successfully. If the *Comments* property is enabled, then clicking Save will display a screen where you need to enter comments related to creating the Advance Positional schema (refer to Figure 3.6).
14. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

15. Click **OK** to save the comments. This displays a screen confirming that the Advance Positional schema activity has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Defining Field Positions Non-Sequentially

By default, the fields of a schema are created in a sequence. At times, you may need to create a schema with field positions that are not in sequence. For example, there are 500 fields in the source file, but you need to create a schema with only 200 fields. But these 200 fields are not in sequence. In such a case, you can create a schema by defining the start and end position of the fields in a non-sequential manner.

Steps to define field positions non-sequentially

1. Click the *Start & End Positions* radio button.
2. Enter the end position of the field in the End field, after which you want to create a field position that is not in sequence.
3. The *Start* field of the next field is activated, where you can define a new start position for the field. For example, you can define a start position (other than

11) such as 15 and the end position such as 25 for the Address field (see Figure 7.21).

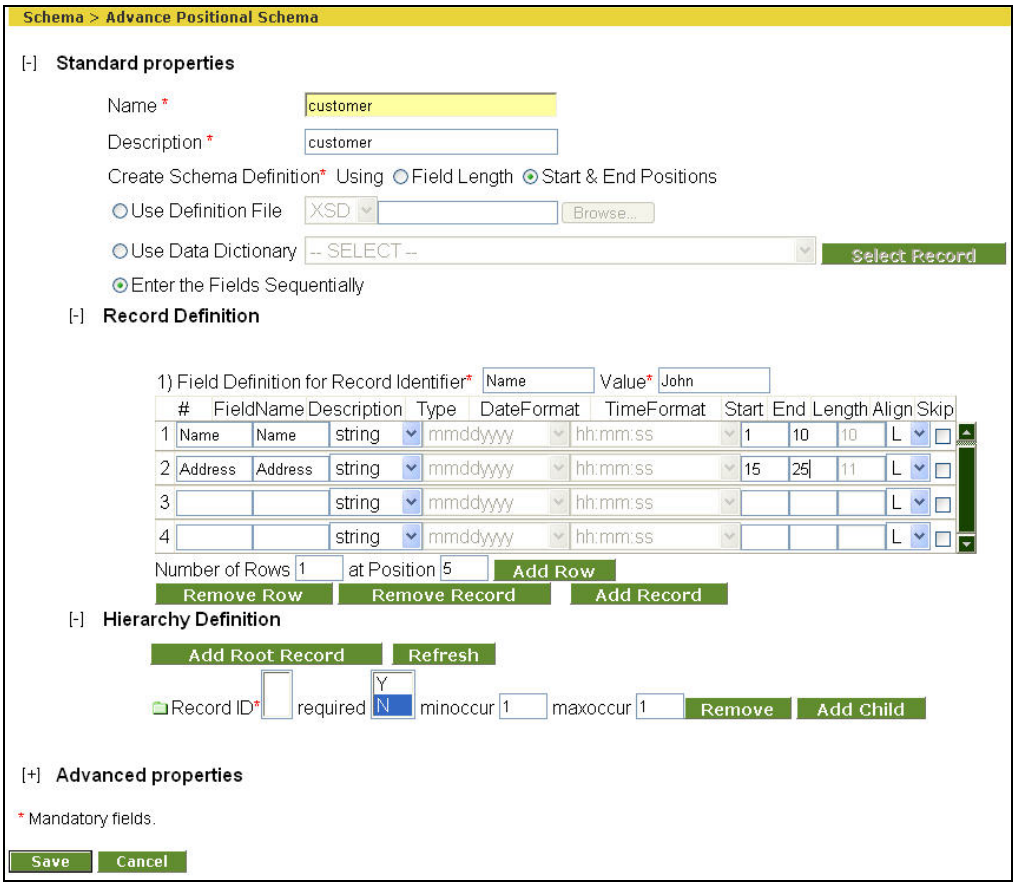


Figure 7.21: Defining Field Positions Non-Sequentially

4. This implies that when the schema is created, the *Name* field is created with 10 positions. However, the *Address* field starts at the 15th position. The positions between 11 and 14 remain blank.

	You can also change the field positions non-sequentially, while editing the schema. This feature is available in the case of Advance Positional and Positional Schemas only.

Viewing Print-Friendly Page

You can view a summary of the schema and its record definition and hierarchies in edit mode.

Steps to view Print-friendly page

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Adv. Positional**. The Manage Advance Positional Schema screen is displayed (refer to Figure 7.10).
4. Click the radio button against the schema activity whose summary you want to view. This selects the schema and activates the **Edit** link. Clicking this link displays the Edit Advance Positional Schema screen (see Figure 7.22).

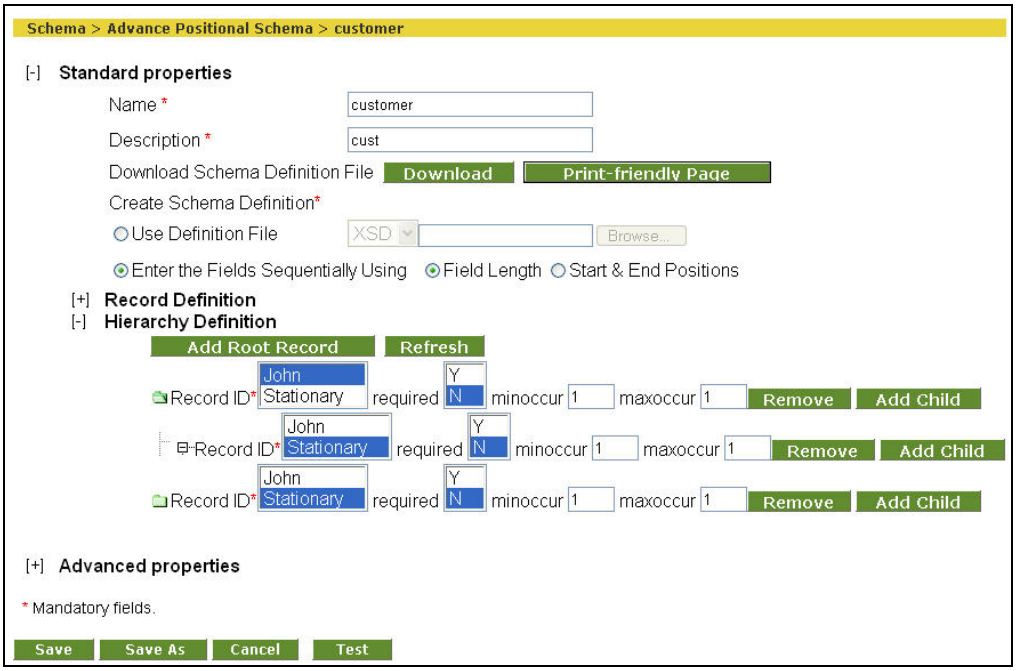
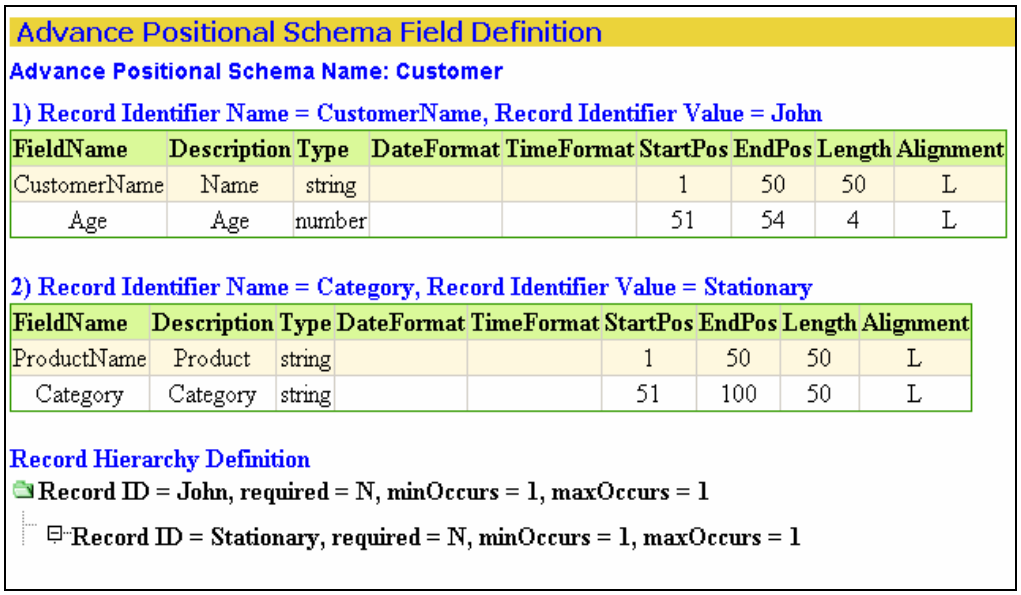


Figure 7.22: Edit Advance Positional Schema

5. Click **Print-friendly Page** button. The summary screen is displayed (see Figure 7.23).



Advance Positional Schema Field Definition

Advance Positional Schema Name: Customer

1) Record Identifier Name = CustomerName, Record Identifier Value = John

FieldName	Description	Type	DateFormat	TimeFormat	StartPos	EndPos	Length	Alignment
CustomerName	Name	string			1	50	50	L
Age	Age	number			51	54	4	L

2) Record Identifier Name = Category, Record Identifier Value = Stationary

FieldName	Description	Type	DateFormat	TimeFormat	StartPos	EndPos	Length	Alignment
ProductName	Product	string			1	50	50	L
Category	Category	string			51	100	50	L

Record Hierarchy Definition

Record ID = John, required = N, minOccurs = 1, maxOccurs = 1

Record ID = Stationary, required = N, minOccurs = 1, maxOccurs = 1

Figure 7.23: Print-Friendly Page

You can print a summary of the EDI segment definition and hierarchies too.

Similarly, you can view the Print-friendly page for the Positional Data Dictionary, by clicking the **Print-friendly Page** button on the Edit Positional Data Dictionary screen.

If the source file of the schema contains skipped fields, then the skipped fields are represented by 'T' and the unskipped fields by 'F'.

Special Usage Scenario

Enable IsRemoveHeader Attribute

In Advance Positional schema used as target schema, you can set the *IsRemoveHeader* attribute. To skip a record from the source file to be inserted/updated in the target database, you need to set *IsRemoveHeader* attribute to 'True'. This attribute is set in Data Mapper, while mapping source and target schemas.

Steps to set IsRemoveHeader attribute

Load required source and target schema in Data Mapper. In target schema, there will be an *IsRemoveHeader* attribute.

1. Create a constant 'true' and map it to IsRemoveHeader attribute of the target schema.
2. Save the Mapping activity.

	To learn how to use Data Mapper, refer to section **Using Data Mapper**.

CREATING ADVANCE TEXT SCHEMA ACTIVITY

The Advance Text Schema activity is used to define how to read data from advance text files and how to write data in advance text file. Advance text file refers to text file which can have multiple record formats and multiple field separators.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Figure 7.24 displays a sample Advanced Text file.

```
12345678D 07/12/2007 XYZ SYSTEMS, INC. 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
Agent: JOHN SMITH 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
Officer: JOHN SMITH 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
56789101D 07/16/2007 XYZ INTERNATIONAL INC 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
Agent: NICK MASSA 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
Officer: NICK MASSA 443 NORTH CLARK AVE, SUITE 350 CHICAGO, IL 60610
```

Figure 7.24: Sample Advanced Text File

In the Text File shown in Figure 7.24 there are three types of records. The first field of all record is considered as the *Record Identifier*. In this example, the record identifier of the first record is an eight-digit number followed by character 'D' (e.g. *12345678D*). It could be any number followed by 'D'. The second and third record identifiers are *Agent* and *Officer* respectively. In this text file, two field separators ':' colon and space are used.

Steps to create Advance Text Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Adv. Text**. The Manage Adv. Text Schema screen is displayed (see Figure 7.25).

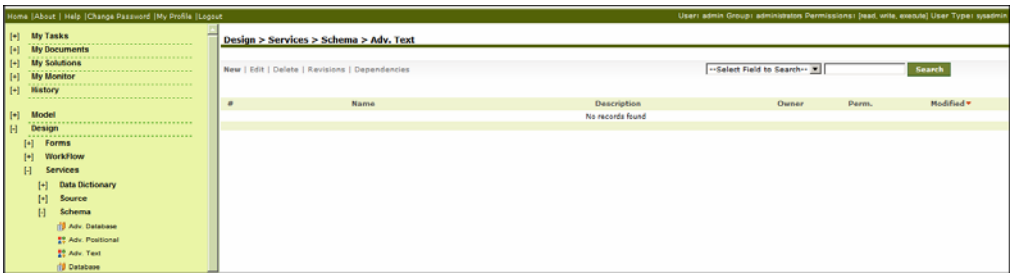


Figure 7.25: Manage Advance Text Schema

4. Click the **New** link. The Create Advance Text Schema screen is displayed (see Figure 7.26).

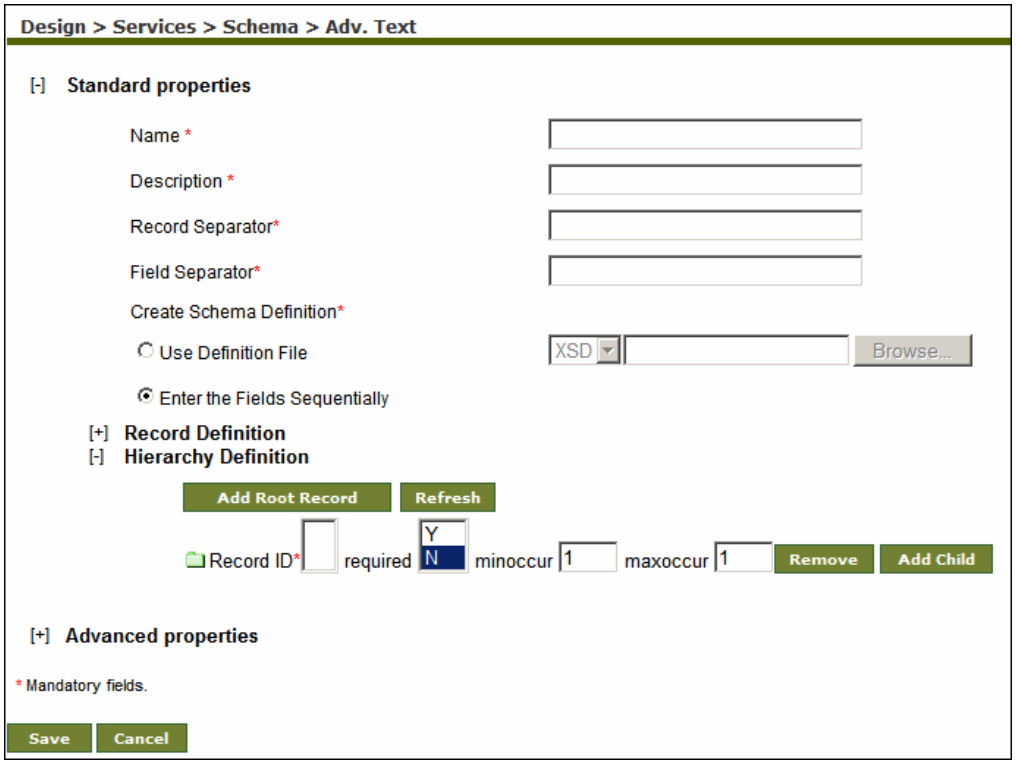


Figure 7.26: Create Advance Text Schema

5. Enter the name and description of new Advance Text schema activity in *Name* and *Description* fields respectively.

6. Enter the record separator, for example \n for new line or \s for space, in the *Record Separator* field. The record separator is used to separate records.

7. Enter the Field Separator, for example \t for Tab or \s for space in the *Field Separator* field. Field Separator is used to separate fields.

	▪ You can specify multiple Record Separators and Field Separators. You can use regular expressions to specify Field Separators. For example "\s|:" can be used to specify space or colon (:) as field

	separator. ▪ Special characters, which are used by regular expression such as '+', '*', or '|' cannot be directly used as field separator. To use such character as field separator you need to use these characters within parentheses e.g. [+], [*] or [|] in *field separator* field. ▪ But if same schema is used at target end, in target data file, field separator will be [+], [*] or [I]. To avoid it enter '+', '*', or '|' etc in the *Target Field Separator* in the *Advanced Properties*.

8. To define schema definition, select one of the following options:
   - Use Definition File
   - Enter the Fields Sequentially
9. To define the schema using definition file, select the *Use Definition File* radio button; select the type of file from the drop-down list and click the Browse button to select the required file.
10. To enter the fields sequentially, select *Enter the Fields Sequentially* radio button and click **(+) Record Definition**. This expands to display the Record Definition fields (see Figure 7.27).

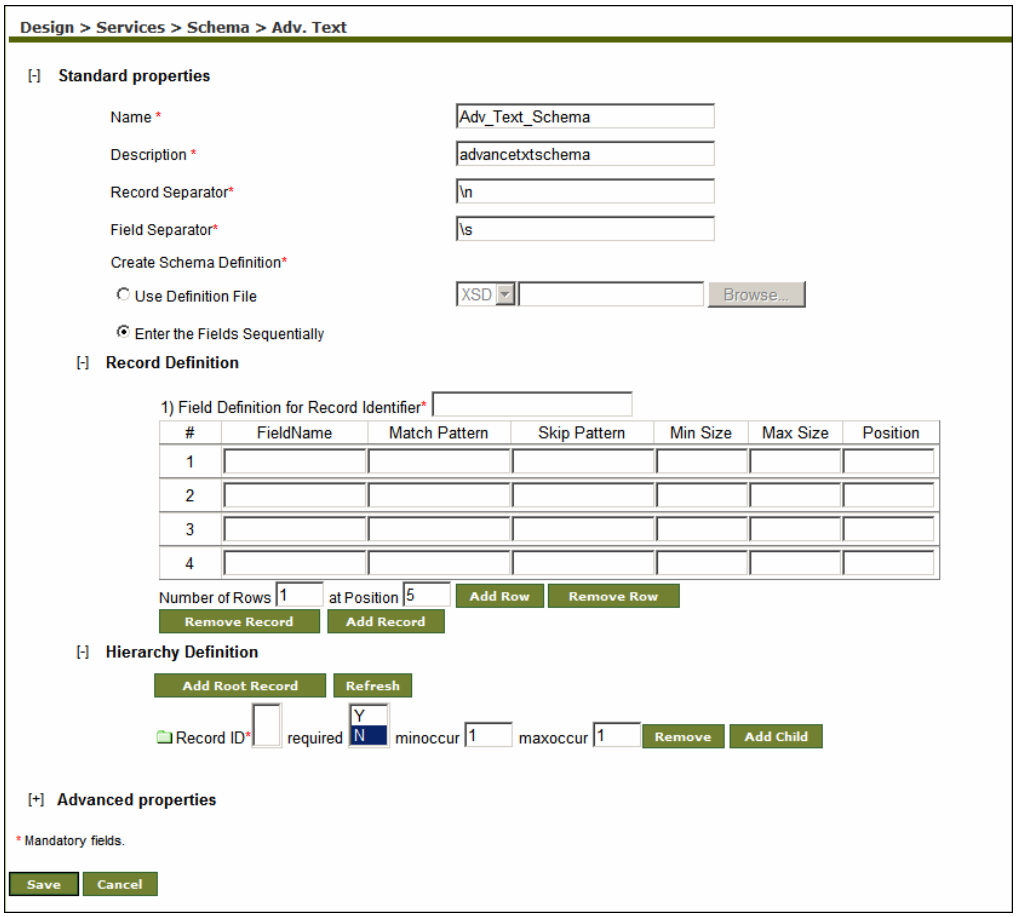


Figure 7.27: Create Record Definition

11. Enter the record identifier in the *Record Identifier* field. You can use regular expression in Record Identifier.
12. Enter name the field in the *FieldName* field.
13. Enter the match pattern, against which you want the record to be matched, in the M*atch Pattern* field.

	You can use regular expression in the Match Pattern. For example, in the data file, there is field Company Name which can have value like **XYZ SYSTEMS, INC**. Since **\S** is used as Field Separator, **XYZ, SYSTEM,** and **INC** will be considered separate fields. But they need to be a part of the same field. To parse this type of data, you can use match pattern. For example **\S+ .*** **(INC\.|INC(ORPORATED)?)** match pattern is used specify that the field can have spaces and end with INC. or INCORPORATED.

14. Enter the Skip Pattern in the *Skip Pattern* field. Skip Pattern is used to skip a particular pattern. You can use regular expression in Skip Pattern.
15. Enter the minimum and maximum size of the field in the *Min Size* and *Max Size* fields respectively.

16. Enter the position in the *Position* field.
17. Click **Add Record** to define another type of Record format, and follow the steps from 10 to 15.

For adding more fields in the same record format, click **Add Row** button.

To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. A maximum of 99 rows can be added at a time.

Advance text schema supports only *String* data type.

To learn about Advanced Properties refer to s**Changing Advanced Properties** section.

You can enable quotes handling, by marking the *Quotes Handling on* checkbox as checked, in Advanced Properties.

If a character (say $) is specified as Field Separator in a record, then any $ character in the field data of that record (Chocolate20perpack) will be considered as a Field Separator, even though it is part of the field data. In the above example the $ after 20 will also be considered as Field Separator, whereas it is the data. To avoid this situation put that field within the double quote i.e. (Chocolate$"20$perpack")

When *Quotes Handling on* is checked regular expression doesn't work on any field.

Header contains the information about the different record structure of the Text file. You may not want to send the header information to target. In this case you can mark the *Ignore Header* checkbox as checked in Advanced Properties. This ignores the header information. This property is applicable only when schema is used at source end. This property is not applicable when schema is used at target end.

If there are multiple record separators in an input file, then to parse this input file, you need to mark the *Look Ahead* checkbox as checked in the Advanced Properties.

At times the input data may contain some characters that are invalid in XML, thus resulting in the mapping getting aborted. You can filter these invalid XML characters by marking the *Filter Invalid XML Characters* checkbox as checked, in Advanced Properties.

18. After defining records you need to define their hierarchy. This is mandatory for creating an Advance Text schema.
19. Once you have added the records and defined the Record Identifiers, click the **Refresh** button under **Hierarchy Definition**. This populates the Record Identifiers of the defined records, in the *Record ID* field under **Hierarchy Definition** (see Figure 7.28).

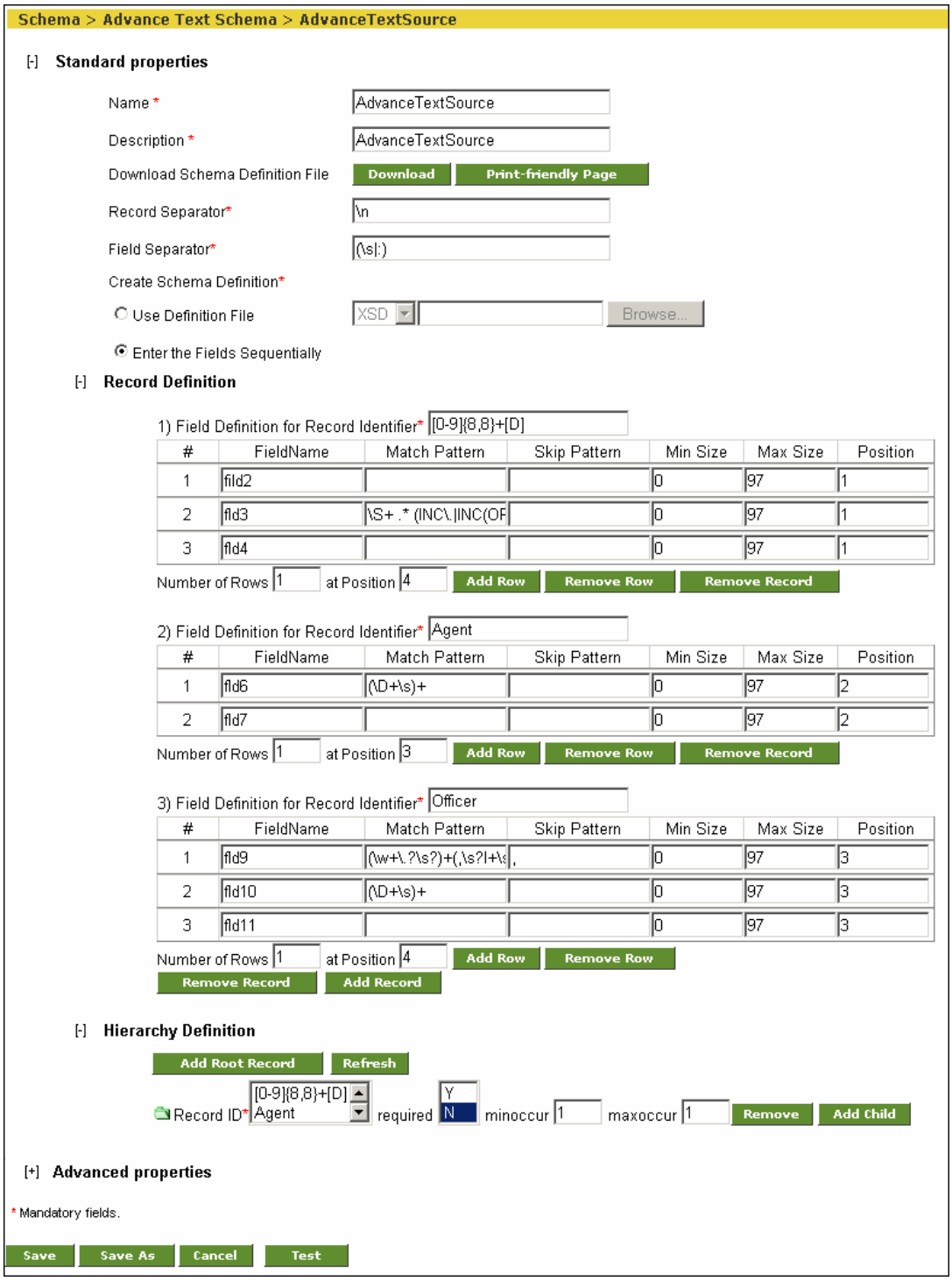


Figure 7.28: Create Record Hierarchy Definition

20. Select a record from the Record ID list box.

21. Select *Y* or *N* from the required list box, to indicate whether the selected record needs to present in the source file.
22. Enter the minimum number of occurrences of the selected record required in the source file, in the *minoccur* field. It should be a minimum of 1.
23. Enter the maximum number of occurrences of the selected record required in the source file, in the *maxoccur* field. It should be a maximum of 2147483647.
24. You can now create a record either at the root level or at the child level.

Using Record at Root Level

To use a Record at the Root Level

1. Click **Add Root Record** button. This creates a record at the same level as that of the displayed record (see Figure 7.29).

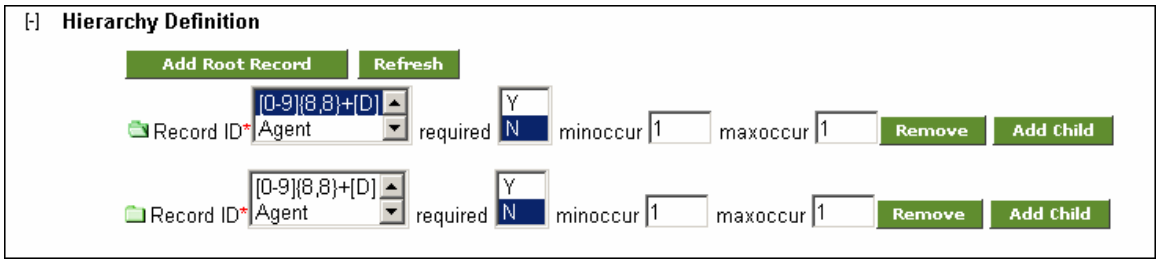


Figure 7.29: Create Root Record

2. Select a record in the *Record ID* field of the displayed record and enter all its related information.
3. Select a record from the *Record ID* listbox of the root record and repeat step 24-25 to create another root Record.

	Each root Record must have a unique Record ID. For example, if *[0-9][8,8]+[D]* is selected as the first root record, then you need to select *Agent* in the next root record.

Using Record at Child Level

To create a record at the Child Level

1. Click **Add Child** button. This creates a record at a level below that of the displayed record (see Figure 7.30).

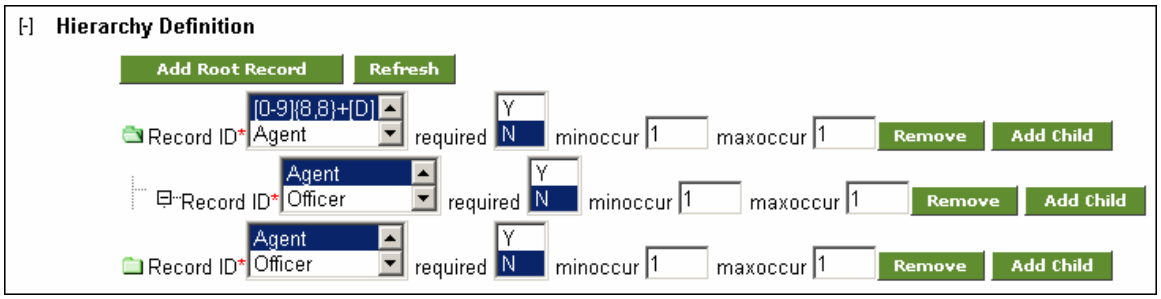


Figure 7.30: Create Child Record

2. Select a record in the *Record ID* field of the displayed record and enter all its related information.

3. Select a record from the *Record ID* listbox of the root record and repeat step 28-29 to create a child record.

	A parent and child record must have a unique Record ID in one hierarchy. For example, if *[0-9][8,8]+[D]* is selected as the parent record, then you need to select *Agent* as the child record. Similarly, if you create another child record under *Agent*, then you need to select *Officer* as its Record ID (see Figure 7.31).

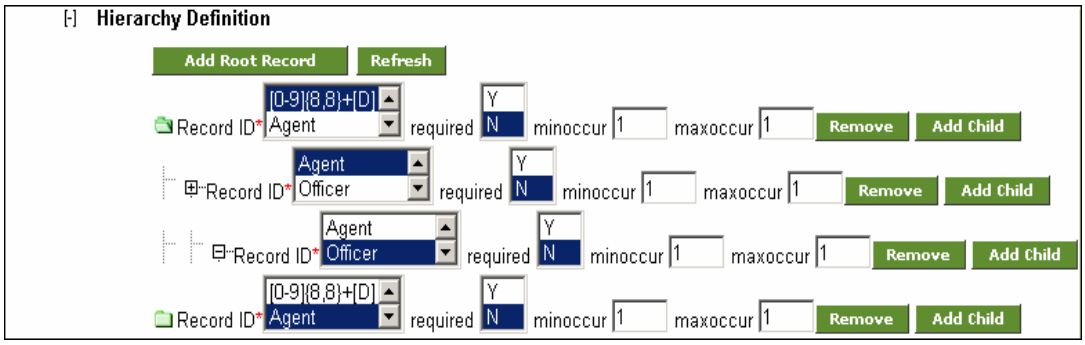


Figure 7.31: Create another Child Record

	You can remove a record by selecting the record and clicking **Remove**. This will display a confirmation delete message, where on clicking **OK**, will delete the record. Alternately, if you delete the last fieldname of a record, and then save the schema, the entire field gets deleted. For example, a record has three fields – *Name, Description* and *Age*. If you delete *Age*, the entire field will get deleted. If a record has one or more child record, deleting the parent record will delete all its child record too.

4. Click **Save** button. This displays a screen confirming that the Advance Text Schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the Advanced Text Schema. (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

6. Click **OK** to save the comments. This displays a screen confirming that the Advanced Text Schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING DATABASE SCHEMA ACTIVITY

The Database schema activity defines the procedure to read data from a database table and to insert/update/delete data into a database table. Database schema uses the predefined Database Info activity to connect to the database.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity must be created before creating *Database Schema* Activity.

Steps to create Database Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Database**. The Manage Adv. Database Schema screen is displayed (see Figure 7.32).

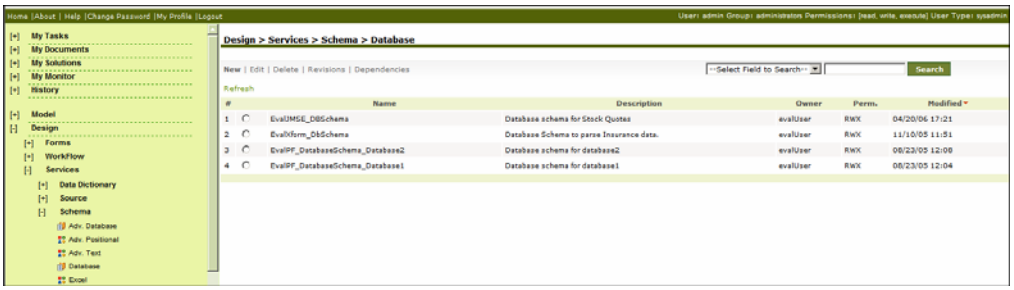


Figure 7.32:Manage Database Schema

4. Click the **New** link. The Create Database Schema screen is displayed (see Figure 7.33).

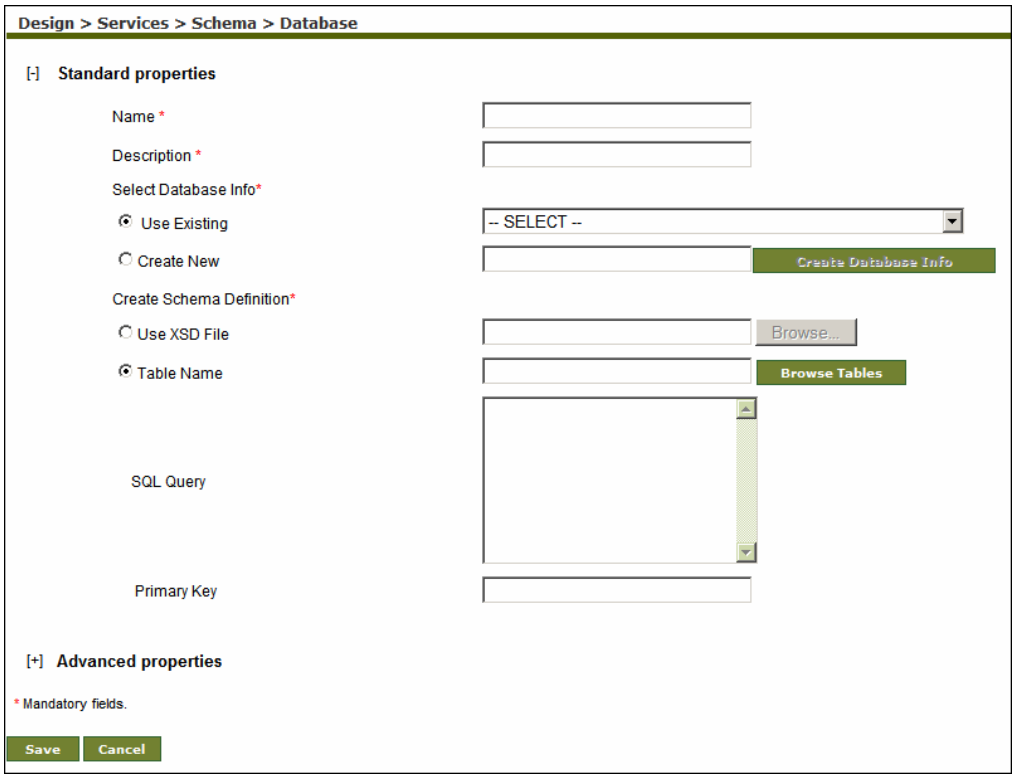


Figure 7.33: Create Database Schema

5. Enter the name and description for new database schema in the *Name* and *Description* fields respectively.

6. Select the database info activity.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administrator Guide*.

7. To select an existing database info activity, select the *Use Existing* radio button and select the database info activity from the drop-down list.

8. To create a new database info activity, select the *Create New* radio button and then click **Create Database Info** button. This displays the Crate Database Info screen (see Figure 7.34).

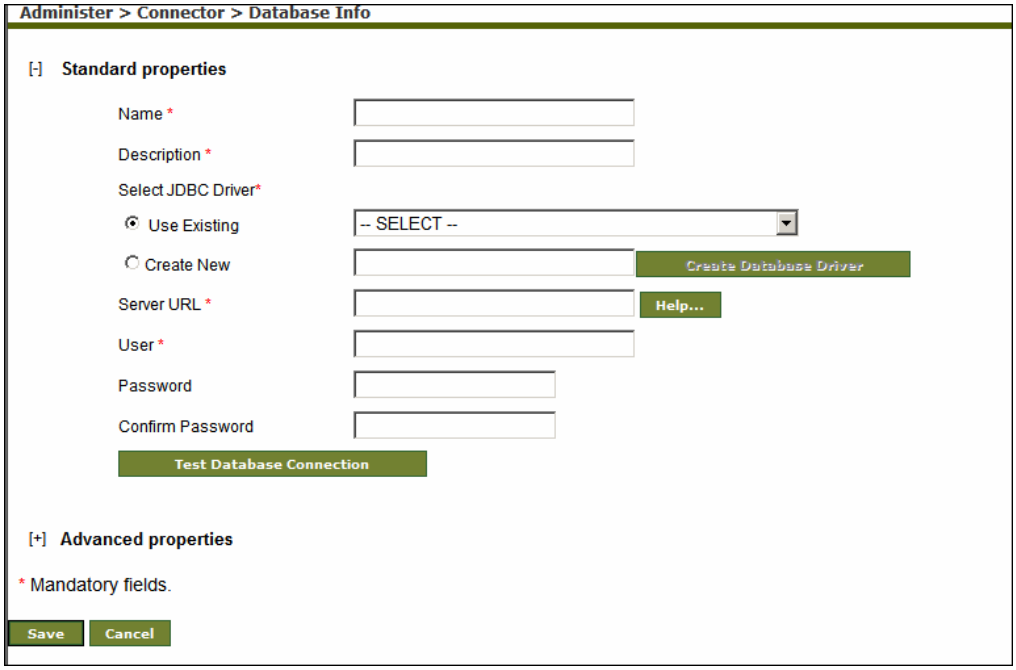


Figure 7.34: Create Database Info

9. Enter the required parameters and click **Save** to save the database info activity and return to Create Database Schema page.
10. To define schema definition, select one of the following options:
    - Use XSD File
    - Table Name
11. To select the XSD file, which contains schema information, select the *Use XSD File* radio button and click **Browse**.
12. To define schema using a database table, select *Table Name* radio button and click Br**owse Tables** button. The Select Schema screen is displayed with the list of RDBMS Schemas in case of *SQL Server* and *DBO* Database Info (see Figure 7.35).

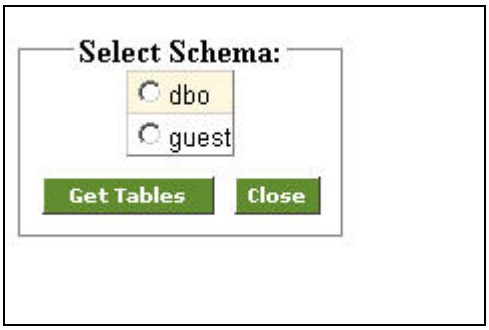


Figure 7.35: Select Schema

13. In the Select Schema screen, select the required RDBMS schema and click **Get Tables**. The Select Table screen is displayed with list of database tables (see Figure 7.36).

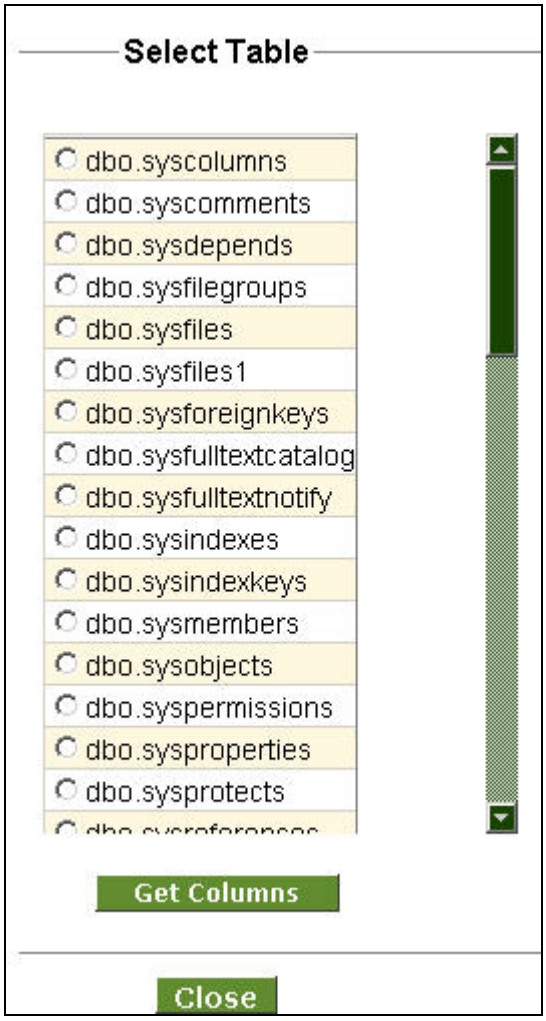


Figure 7.36:Select Tables

	A Close button appears on the Select Table screen, in case of *SQL Server* or *DBO* Database Info. Clicking this button takes the control to the Select Schema screen. If the schema and its tables are created on *DB2*, then you need to remember that the schema name in *DB2* is case-sensitive. Else, it will display the same table more than once.

14. Select the required table and click **Get Columns** button. The Select Table Column(s) screen is displayed with list of columns and their data types along with the data format whether encrypted or plaintext (see Figure 7.37).

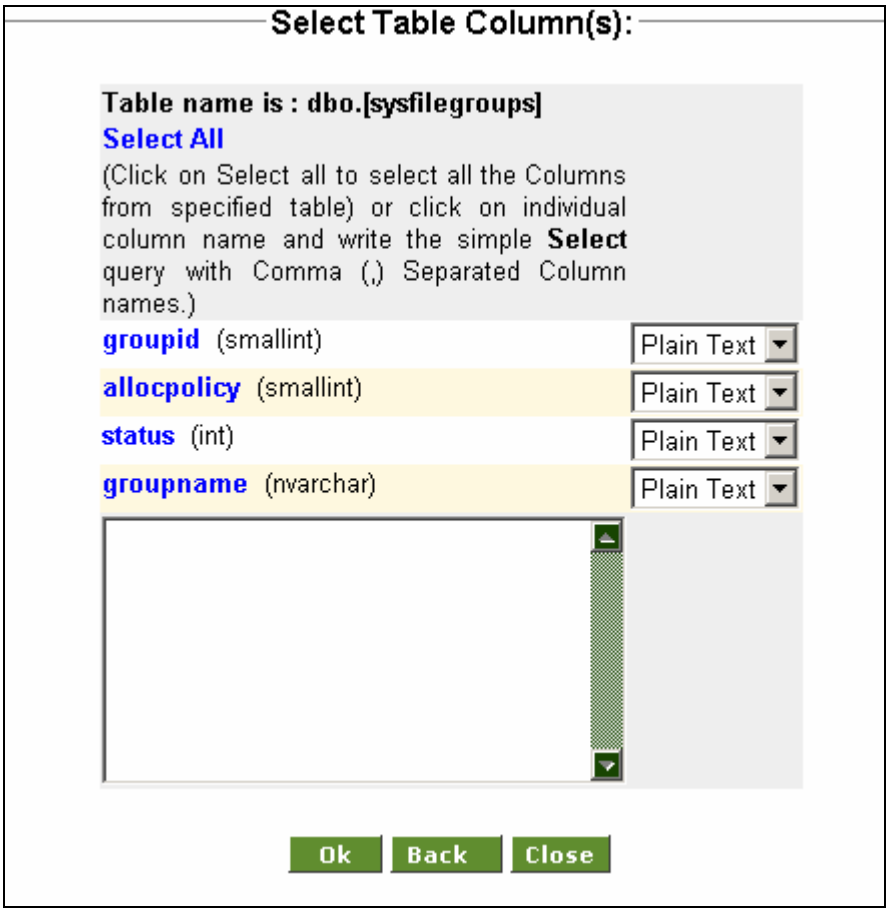


Figure 7.37: Select Table Column(s)

15. Click **Select All** link to get select query for all columns or click on individual column names to get only those columns in select query in the provided text field. You can also specify the where clause in the generated select query.

	There should not be any special character (except $ and SID#) in column name. A column name should not include any standard SQL keyword (for example, *Identity*), as it generates an error, when the database schema is used as a source or target database.

16. Select the mode of data in the column, whether *Encrypted* or *Plain Text* from the drop-down list. If the table column is used at source end and encrypted data is coming from the source, select *Encrypted* from the drop-down list. Similarly if the table column is used at the target end and you want to send the encrypted data to the target, select *Encrypted* from the drop-down list.

- If *Encrypted* is selected in table column, you must select *Encryption/Decryption secret key* from Advanced Properties.
- If *Encrypted* is selected, the encrypted data length can be more than the source data length. In this case you must define the target table column length more than the corresponding source table column length.
- If *Encrypted* is selected in table column, you must set the *Data Action* property in Process Designer, while creating the process flow. To know more about Process Designer, refer to the **Working with Process Flow** section.

17. Click **OK** to return to the Database Schema screen. The generated select query is displayed in the *SQL Query* field. You can edit this query, if required.

- This query is validated, once you click the **Save** button.
- In case some SQL function is used or some calculation is done over the Column Name(s), use the alias name for that Column Name(s).

  For example, for query like:

  SELECT SID,Name+Dept,Salary FROM dbo.Employee

  You can use:

  SELECT SID, (Name+Dept) as AliasName,Salary FROM dbo.Employee

  Reason:

  While getting the result set, you have the assign the output of (Name+Dept) to some new field.

18. Enter the Primary Key in the *Primary Key* field. Primary is the name of the field on the basis of which target database table is updated. Primary key is only used, when the database schema is used with database target.
19. Click **[+]** to expand **Advanced Properties**. Advanced properties of the Database Schema are displayed (see Figure 7.38).

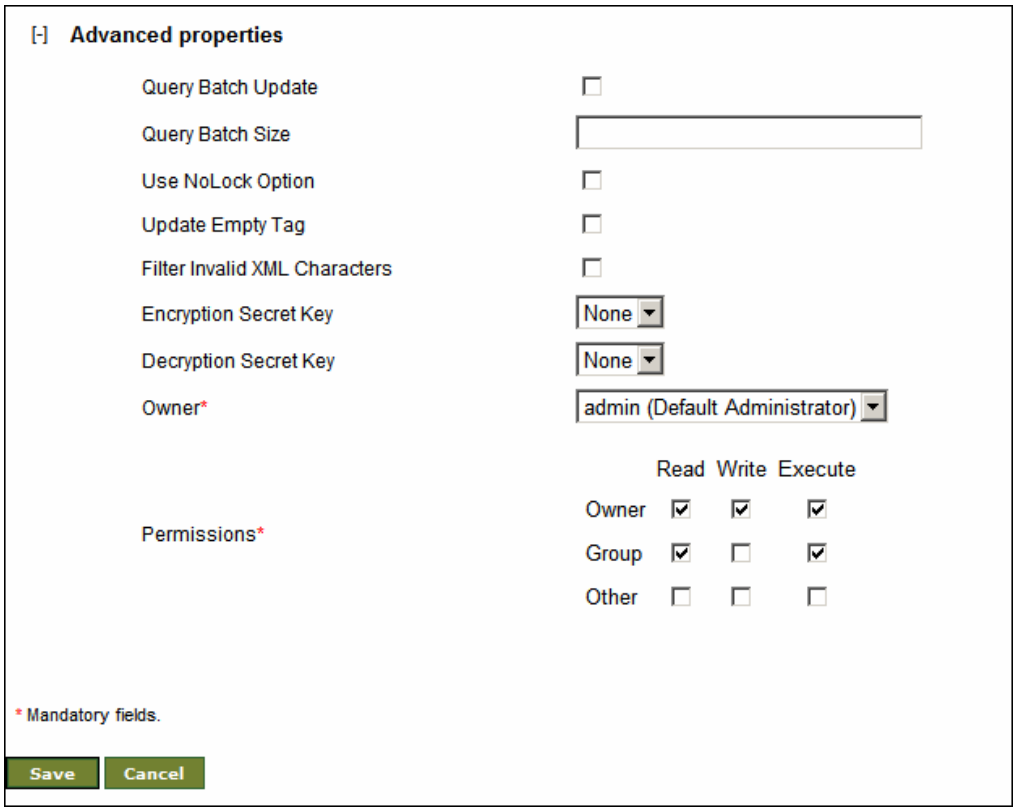


Figure 7.38: Advanced Properties of Database Schema

20. By default one record at a time is updated in Database Target. You can specify the number of records that can be updated at a time in database target. To enable batch update, check the *Query Batch Update* checkbox and enter the number of records in the *Query Batch Size* field.

21. To enable the No Lock option, check the *Use No Lock Option* checkbox. Whenever there are chances of locking the database table, you can enable Use No Lock Option. When this option is enabled, database schema can read the data from the database table even if the database table is locked.

22. When *Update Empty Tag* option is checked, empty tag (e.g. -<id/>) in the input XML to the database target is handled as given below:

Number Type	:	updated to null value
Date Type	:	updated to null value
String type	:	updated to empty value

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

23. In the Database Schema screen click **Save** button. This displays a screen confirming that the database schema activity has been created successfully. If the *Comments* property is enabled, then clicking Save will display a screen where you need to enter comments related to creating the database schema (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

25. Click **OK** to save the comments. This displays a screen confirming that the database schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Special Usage Scenario

Delete Target Records

In Database schema, you can set *Action* attribute to delete the records from the target database table, if they are matching with the records of source database table. If Primary Key is defined in the source and target schemas, source and target records are matched based on primary key. If primary key is not defined, whole records are matched. This attribute is enabled in Data Mapper, while mapping source and target schemas.

Steps to set action attribute to delete matching records

Load required source and target schema in Data Mapper. In target schema, there will be an *Action* attribute.

1. Create a constant *'delete'* and map it to *Action* attribute of the target schema.

	In order to delete records where the target end database is ORACLE, IBM DB2, MS SQL server or HSQLDB, the keyword "delete" should be used. However for MS SQL server DBMS, the "cancel" keyword can also be used. "cancel" keyword cannot be used for other RDBMS except Ms SQL server.

2. Save the Mapping activity.

	To learn how to use Data Mapper, refer to section **Using Data Mapper**.

CREATING EDI SCHEMA ACTIVITY

	This feature is a paid service and is thus not available in any of the Adeptia products by default.

The EDI (Electronic Data Interchange) Schema activity is created to define how the data in predefined EDI format; an industry standard is managed.

	This feature is a paid service and is thus not available in any of the Adeptia products by default.

Steps to create EDI Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **EDI**. The Manage EDI Schema screen is displayed (see Figure 7.39).

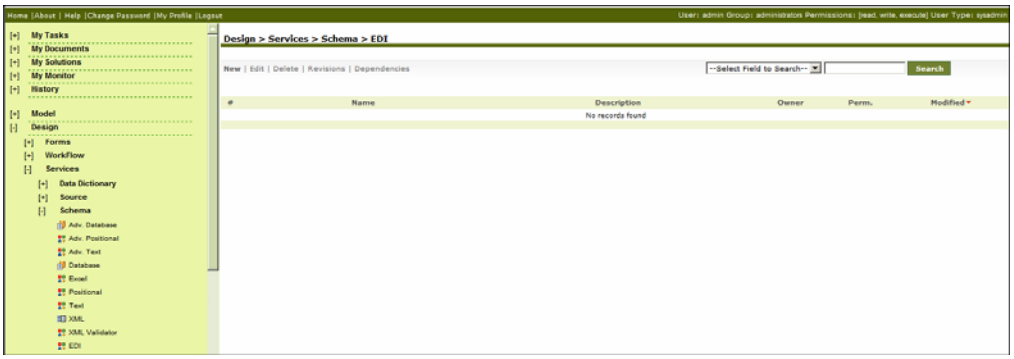


Figure 7.39: Manage EDI Schema

4. Click the **New** link. The Create EDI Schema screen is displayed (see Figure 7.40).

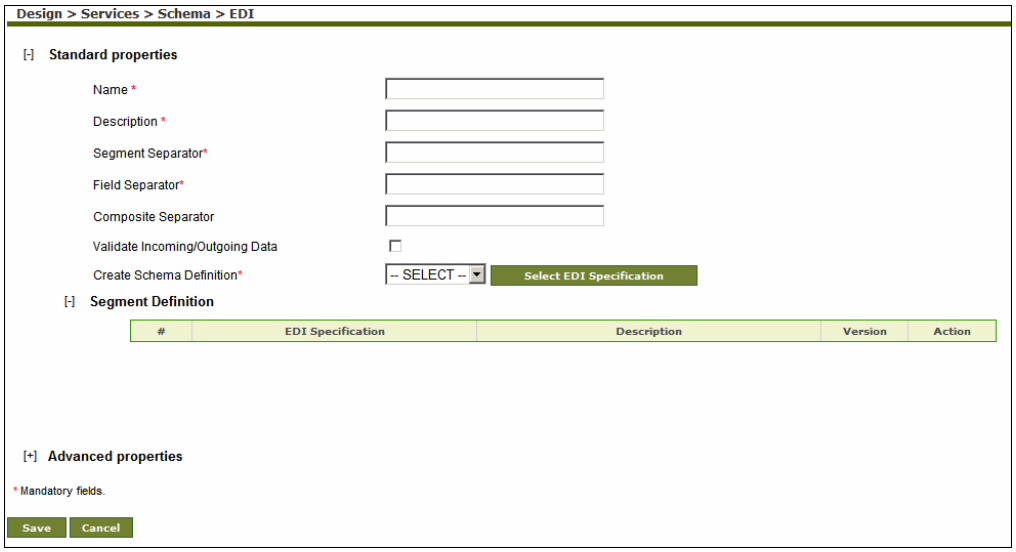


Figure 7.40: Create EDI Schema

5. Enter the name and description of new EDI schema activity in the *Name* and *Description* fields respectively.
6. Enter the Segment Separator, e.g. '~' in the *Segment Separator* field.
7. Enter the Field Separator, e.g. '~' in the *Field Separator* field.
8. Enter the Composite Separator, e.g. '+' in the *Composite Separator* field. A Composite Separator is used to separate composite fields (a field which contains more than one sub-fields) inan EDI file.
9. Check *Validate Incoming/Outgoing Data* checkbox if you want to validate the incoming and outgoing EDI Data. If this schema is used at source end, it validates the incoming EDI file and generates intermediate XML stream and an EDI acknowledgment stream. If this schema is used at target end, then it validates the input XML and generates error records if there is any error.
10. Select the required EDI Data Dictionary from the *Create Schema Definition* field and click **Select EDI Specification** button. This displays the Select EDI Specification screen (see Figure 7.41).

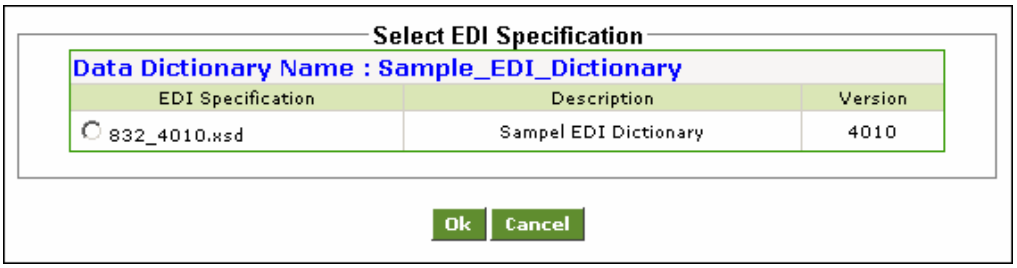
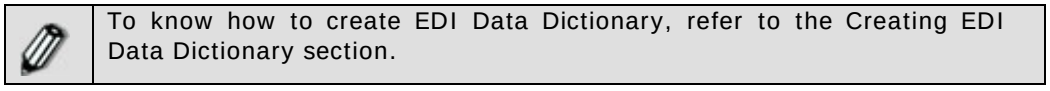


Figure 7.41: Select EDI Specification



To know how to create EDI Data Dictionary, refer to the Creating EDI Data Dictionary section.

11. Select the required EDI Specification and click **Ok** button. This closes the Select EDI Specification screen. The selected EDI specification is populated under Segment Definition in Create EDI Schema screen (see Figure 7.42).

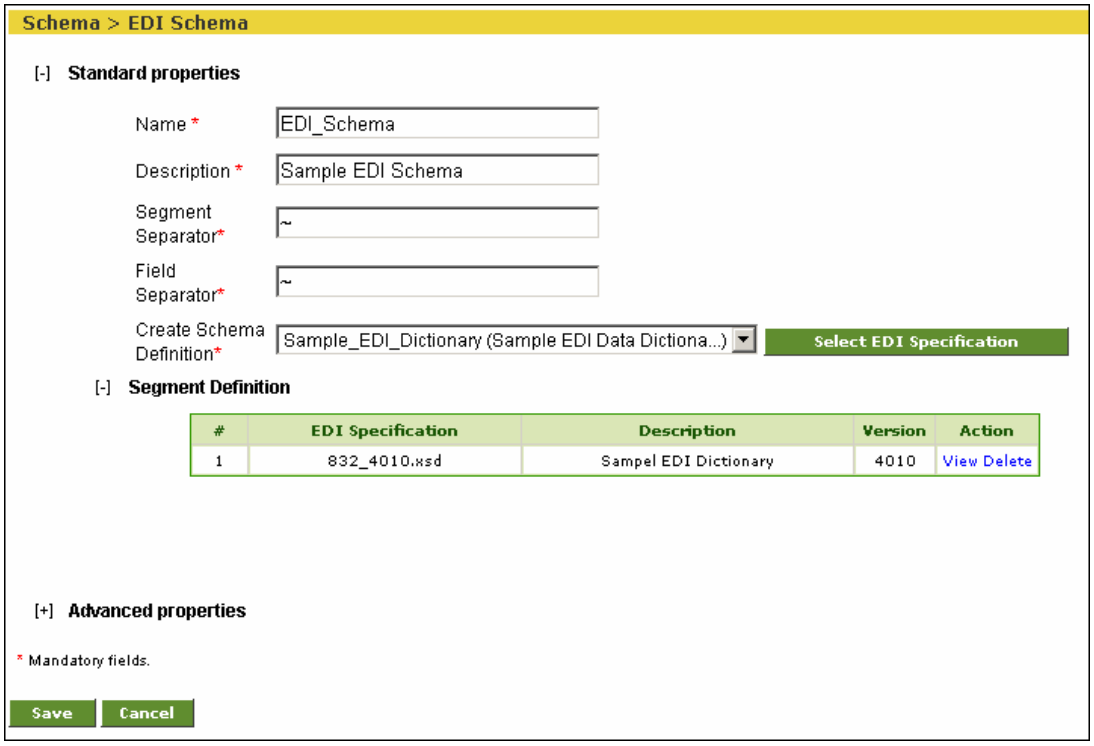


Figure 7.42: Populated Create EDI Schema screen

	To learn about Advanced Properties refer to **Changing Advanced Properties** section

12. Click **Save** button. This displays a screen confirming that the EDI schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the EDI schema (refer to Figure 3.6).
13. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

14. Click **OK** to save the comments. This displays a screen confirming that the EDI schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Manipulating EDI Schema Hierarchy

At times you may not need all the elements of an EDI schema hierarchy. In such cases, you can remove the elements that you do not require, from the hierarchy. If you remove an element at the parent level, then all its sub-level elements are also removed.

Steps to manipulate EDI Schema hierarchy

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **EDI**. The Manage EDI Schema screen is displayed (refer to Figure 7.39).
4. Click the EDI schema whose hierarchy you want to manipulate, and click the **Edit** link. This displays the EDI schema in the Edit mode (see Figure 7.43).

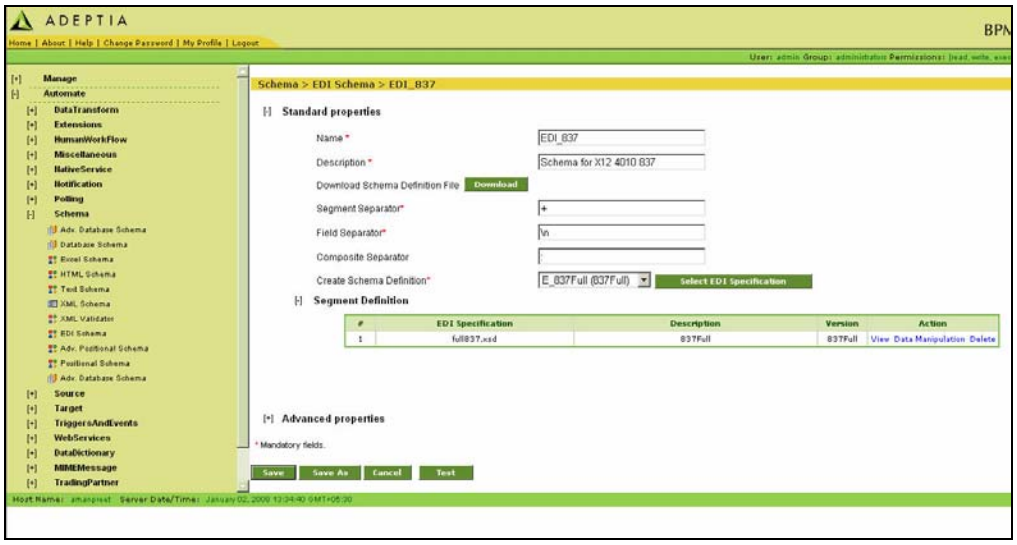


Figure 7.43: Edit EDI Schema

5. Click the **Data Manipulation** link against the EDI Segment whose hierarchy you want to manipulate. This displays the Schema XSD Tree at the root level (see Figure 7.44).

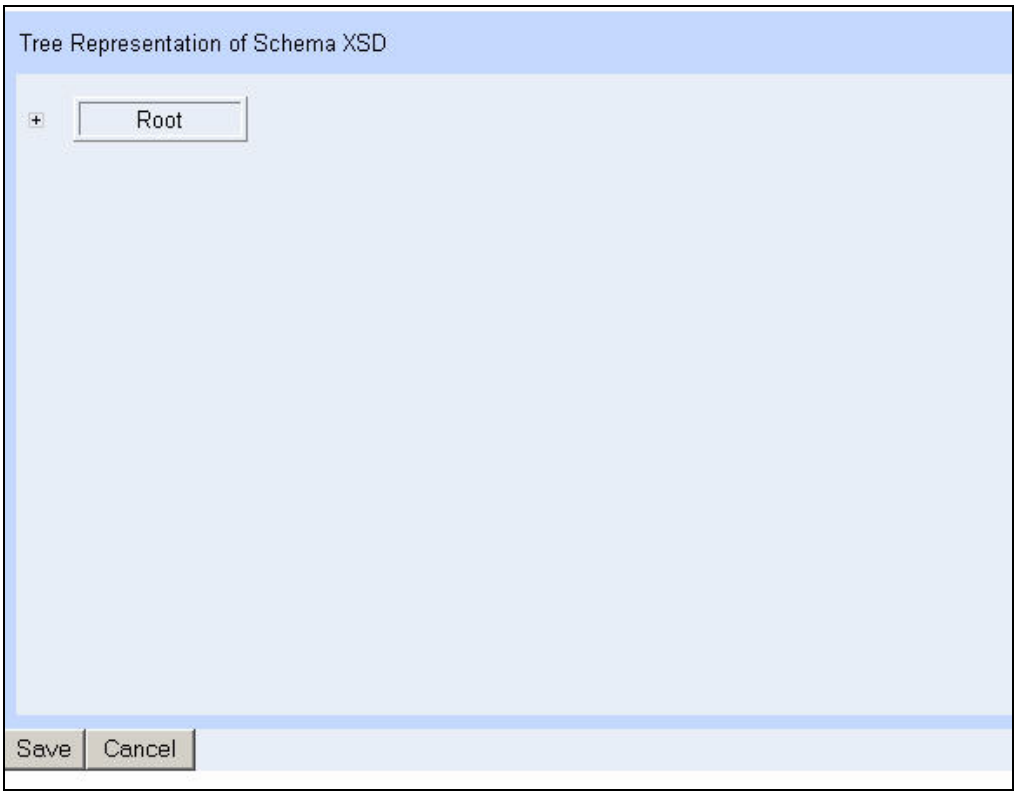


Figure 7.44: Tree Representation of Schema XSD

6. Expand the **Root** node to view the segment hierarchy (see Figure 7.45).

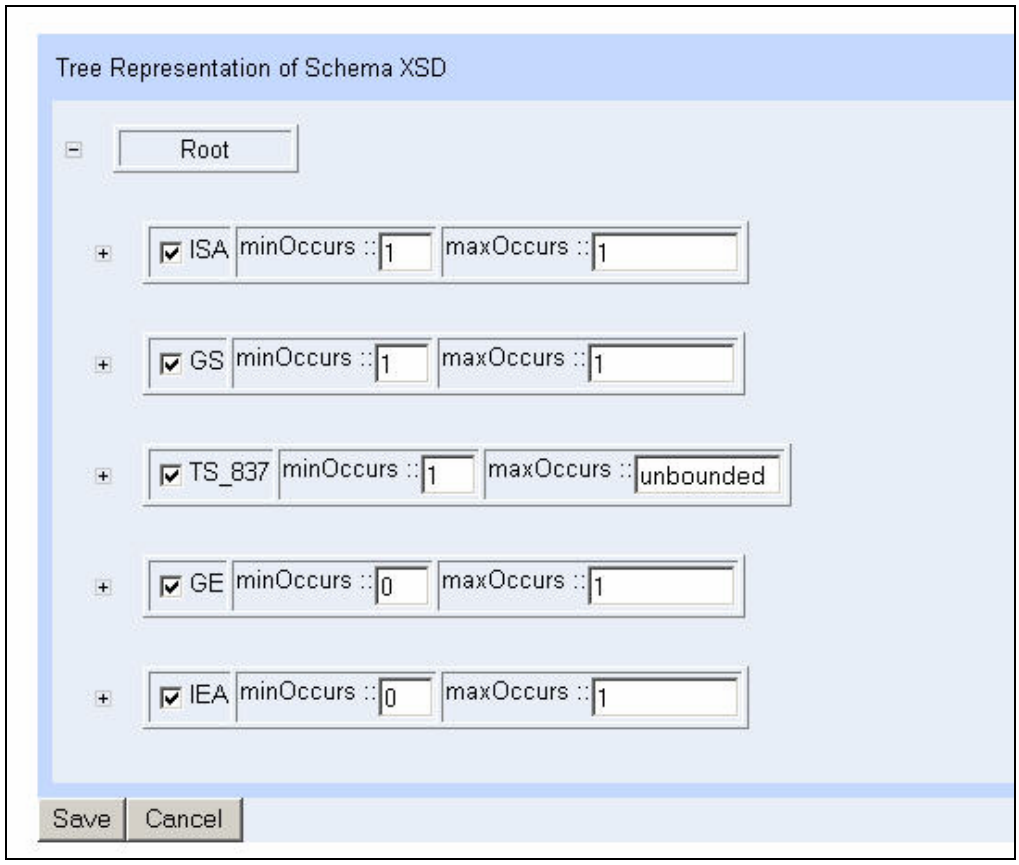


Figure 7.45: Expand Root Node

7. Expand a parent node to view its child nodes (see Figure 7.46).

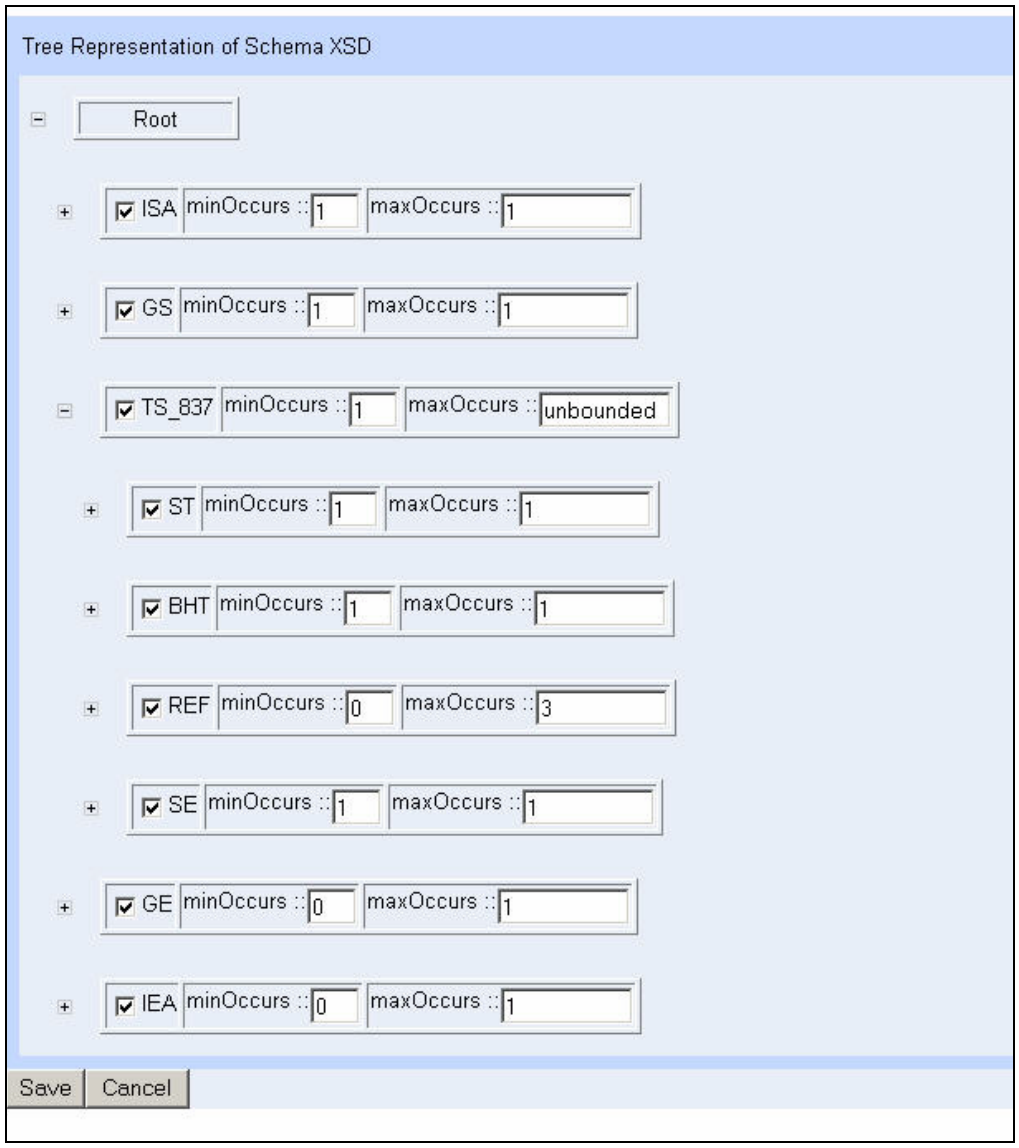


Figure 7.46: Expand Parent Node

8. You can remove a child node by unchecking the checkbox (see Figure 7.47). If you uncheck a parent node, then all its child nodes will also be removed.

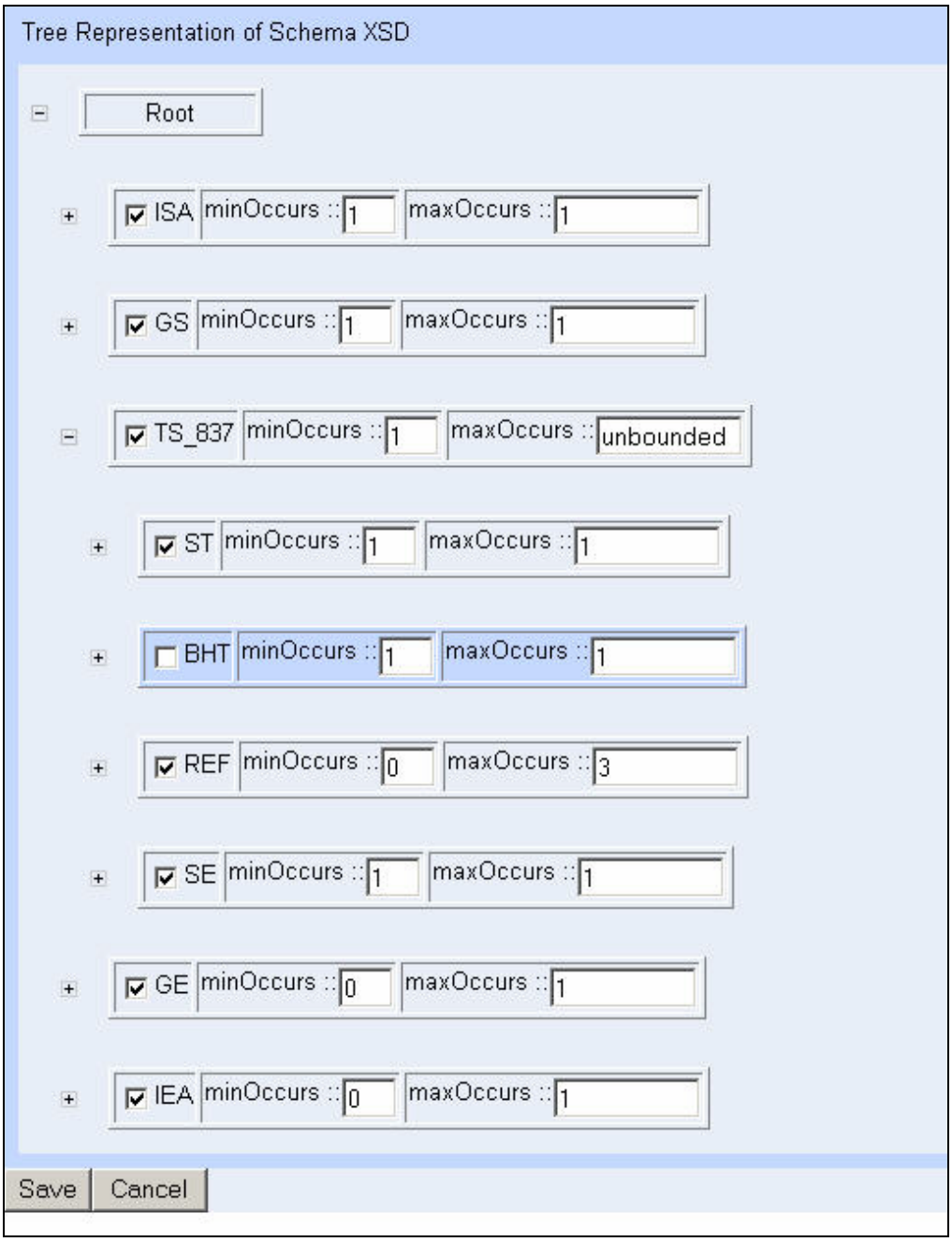


Figure 7.47: Remove Child Node

9. You can also change the occurrence of the elements in the hierarchy by changing the values in the *minOccurs* and *maxOccurs* fields.

10. Once all the changes are done, click **Save**. This saves the changes and displays the manipulated hierarchy (see Figure 7.48).

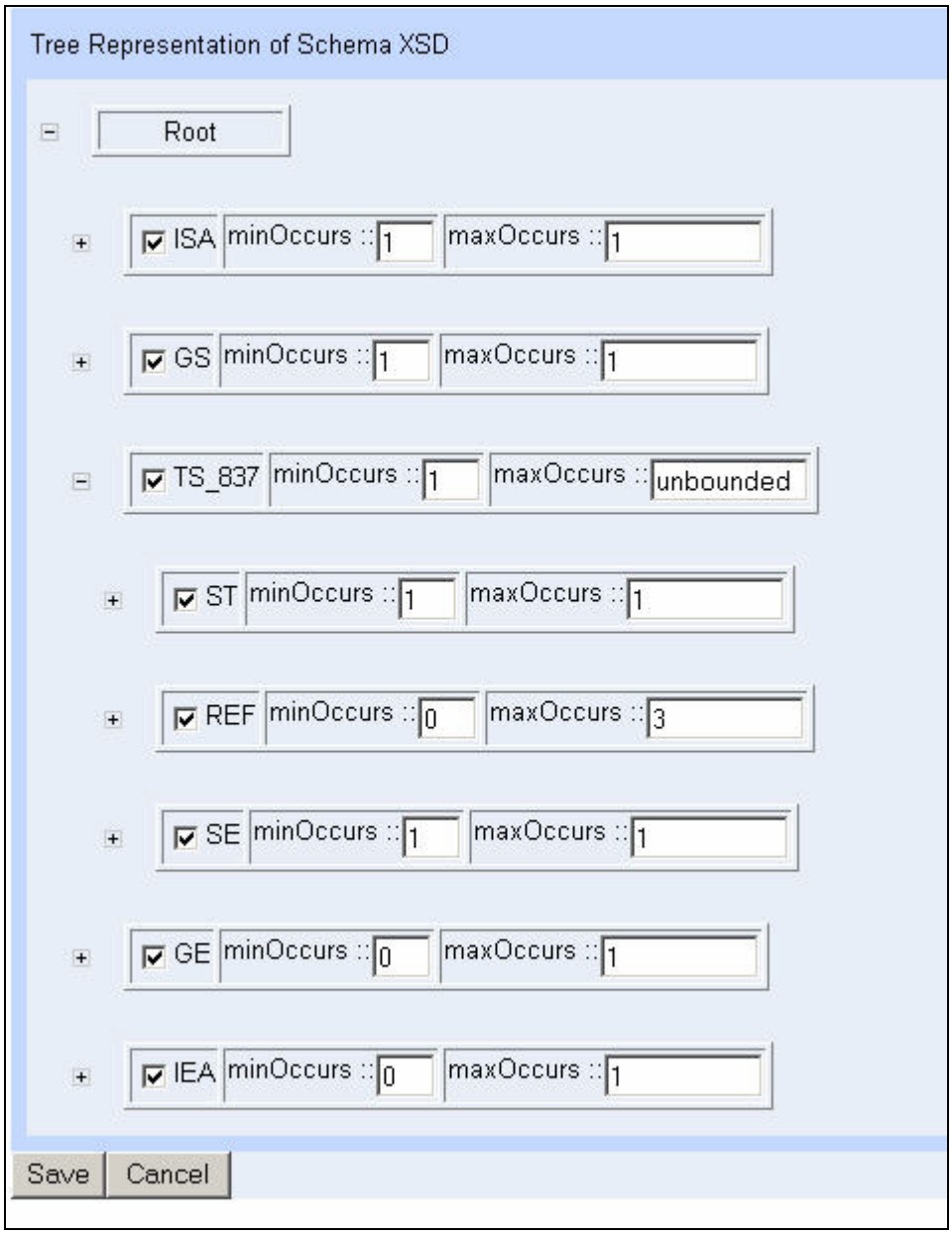


Figure 7.48: Saved Manipulation

CREATING EXCEL SCHEMA ACTIVITY

The Excel Schema activity is used to define how to read data from an Excel file, and write data to an Excel file. To do so, user needs to specify the name of the Excel sheet and required fields, so as to enable identification of those fields.

While creating Excel Schema you can also define hierarchy (parent-child relationship) between the records. Parent-child relationship can be defined only when you create the schema using data file.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create Excel Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Excel**. The Manage Excel Schema screen is displayed (see Figure 7.49).

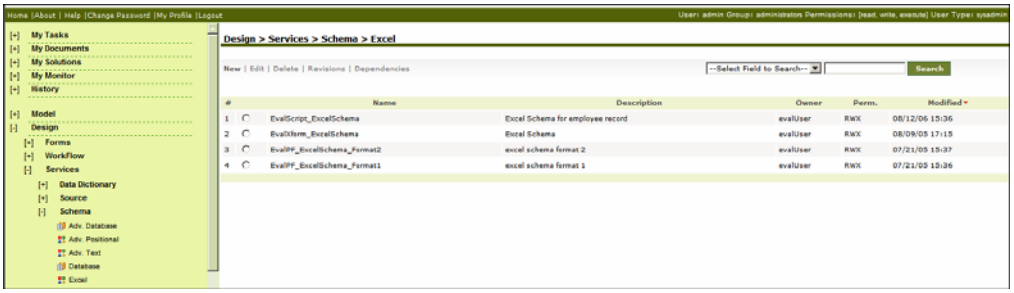


Figure 7.49: Manage Excel Schema

4. Click the **New** link. The Create Excel Schema screen is displayed (see Figure 7.50).

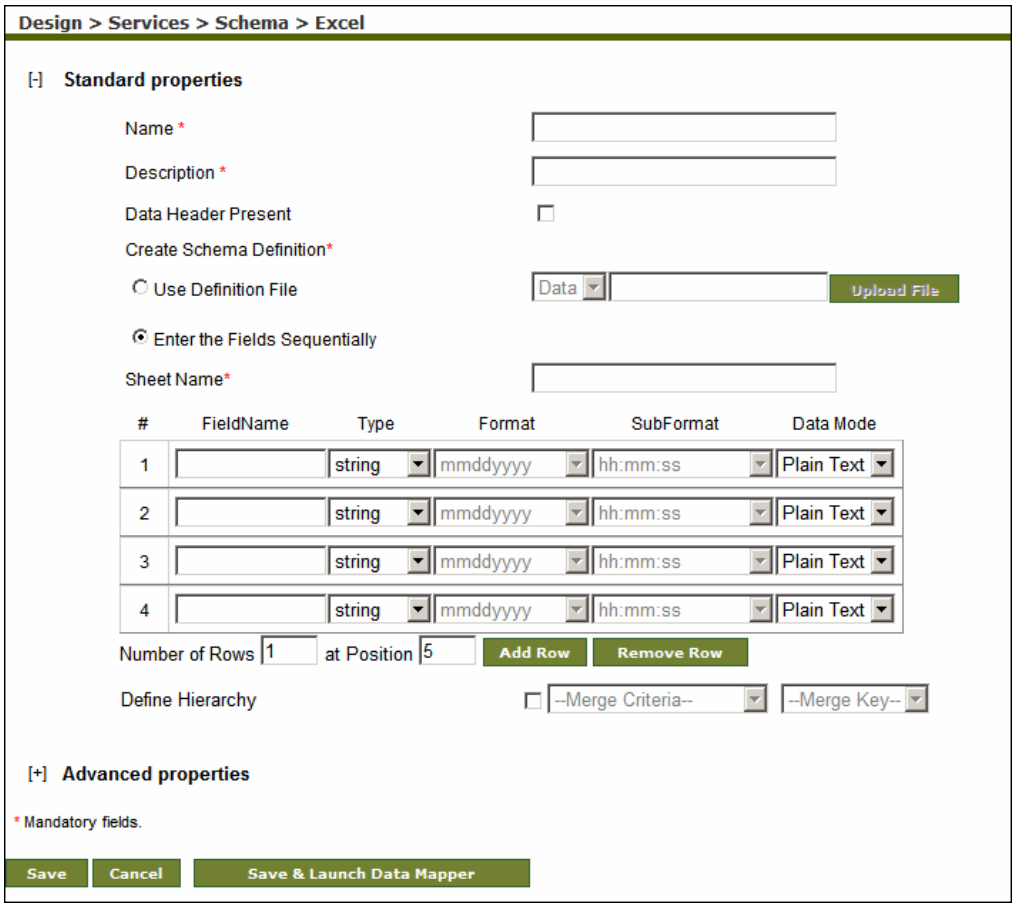


Figure 7.50: Create Excel Schema

5. Enter the name and description of new excel schema in the *Name* and *Description* fields respectively.

6. Data Header usually contains the name of the fields in an excel file. If schema is used at source end, and data header is present in the file, check the *Data Header Present* checkbox. If the schema is used at the target end, and the Data Header Present checkbox is checked, the Header will be written in the target excel file.

	Name of the Headers in the data file and the FieldNames in schema must be same and in same order. If they are not same, then you need to use Dynamic Header Support option. To know how to use Dynamic Header Support, refer to the **Using Dynamic Header Support** section.

7. To define the schema using definition file, select the *Use Definition File* radio button; select the type of file from the drop-down list and click the **Upload File** button to select the required file. The Schema File upload screen is shown (see Figure 7.51).

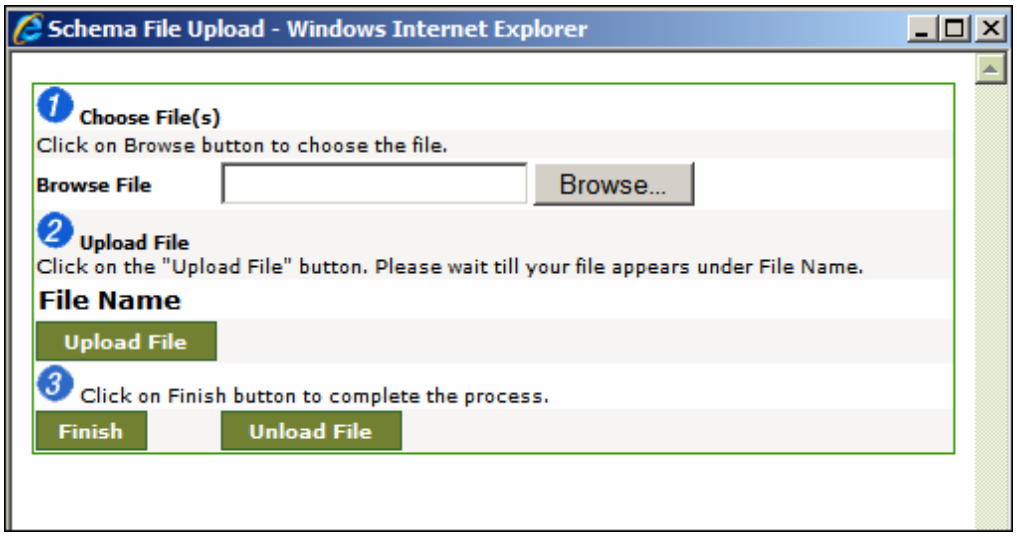


Figure 7.51: Upload file

8. Click **Browse** and select the file, you want to upload. Path of the selected file is shown in the *Browse File* Field.

9. Now click **Upload File** button. Name of the uploaded file is shown in the *File Name* list and list of sheets of the excel file is shown (see Figure 7.52).

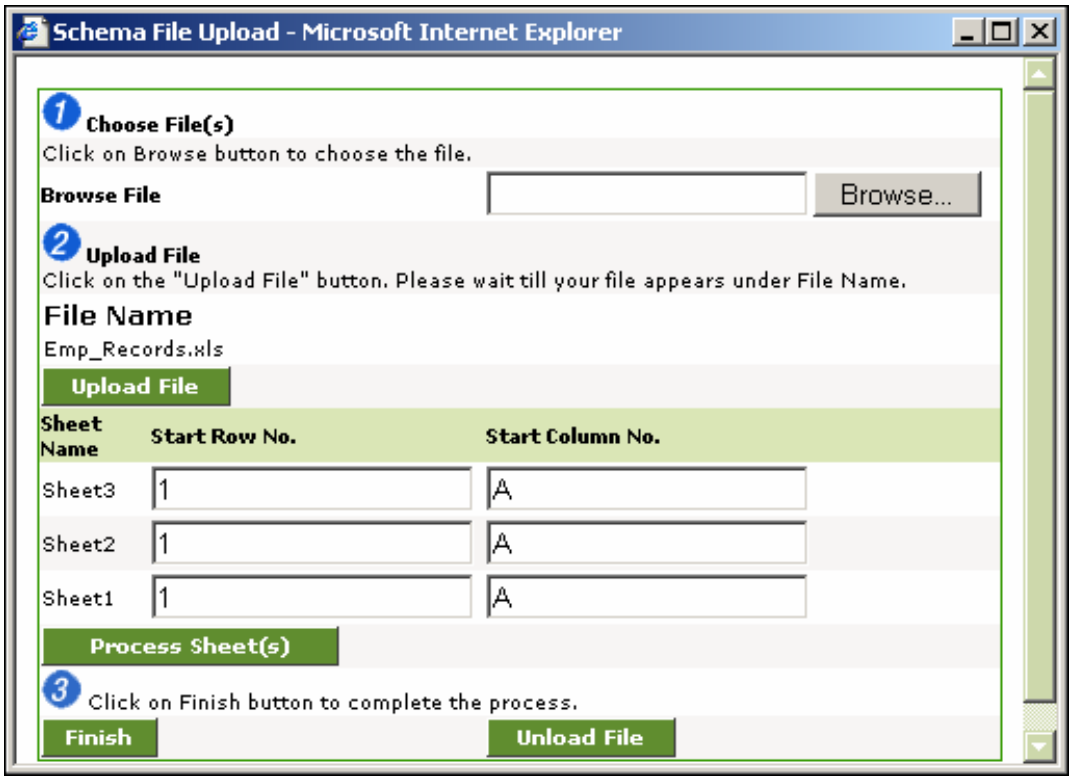


Figure 7.52: Specify Start Row and Start Column No.

10. Specify the Start Row No. and Start Column No. of the sheets in respective fields and then click *Process Sheet(s)*. This will read the sheets and field names.

	Start Row No. and Start Column No. specifies that from which row and Column onwards data should be fetched. For example if you have an excel file in which first 4 rows of sheet1 are blank. So in that case you have to specify 5 in sheet1 *Start Row No.* Another scenario can be that the data is there in first 4 row, but you don't want to fetched those records.

11. Once the processing is done, click **Finish** to close the Schema File Upload screen and return to create schema page (see).
12. Select the sheet name of the excel file for which you are creating this schema activity, from the *Sheet Name* drop-down list. Fields of the selected sheet are populated.
13. If you want to create schema by entering the fields manually, select *Enter the Fields Sequentially* radio button and follow the steps given below:
14. Enter the sheet name of the excel file in the *Sheet Name* field.
15. Enter the name of each field in the *FieldName* field.
16. Select the type of data from the *Type* drop-down list. The datatypes supported by Excel schema are listed in the table below.

Table 7.2: Supported Datatypes

Data Type	Description
String	This data type is selected if the field will accept a string value.
Number	This data type is selected if the field will accept a numeric value.
Date	This data type is selected if the field will accept a Date or Date and Time value.
Currency	This data type is selected if the field will accept a currency value.

	The Currency datatype is supported by Excel Schema only.

17. If data type is *Date*, select the format of date and time from the *Format* and *SubFormat* drop-down lists respectively.

	In case date format is defined in any column of the Excel file, please ensure that every row of that column should have same date format as defined in the first row, else an error record will be generated. In case date format is selected as **3/14/2001*, and the schema is used at source end, the date is parsed according to local date format. This date format is not supported when the schema is used at target end. The time format should be defined as *hh:mm:ss*.

18. If the datatype is selected as *Currency*, select the required currency from *Format* drop-down list. The currencies supported by excel schema are listed in the table below.

Table 7.3: Currencies Supported by Excel Schema

Currency	Description
Dollar($)	Dollar
USD	US Dollar
CAD	Canadian Dollar
CNY	Chinese Yen
EUR	Euro
GBP	British Pound
JPY	Japanese Yen

19. Select the mode of data, whether *Encrypted* or *Plain Text* from the *Data Mode* drop-down list. If the schema is used at source end and encrypted data is coming from the source, select *Encrypted* from the *Data Mode* drop-down list. Similarly if the schema is used at the target end and you want to send the encrypted data to the target, select *Encrypted* from the *Data Mode* drop-down list.

- If Encrypted is selected in Data Mode, you must select *Encryption/Decryption secret key* from Advanced Properties.
- If Encrypted is selected in Data Mode, you must set the *Data Action* property in Process Designer, while creating the process flow. To know more about Process Designer, refer to the **Working with Process Flow** section.
- To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. Maximum 99 rows can be added at a time.
- To remove rows, specify the number and position of the rows to be deleted in the *Number of Rows* and at *Position* fields respectively and click **Remove Row** button.

If you delete the last fieldname of a record, and then save the schema, the entire field gets deleted. For example, a record has three fields – *Name, Description* and *Age.* If you delete *Age*, the entire field will get deleted.

20. Click **[+]** to expand the **Advanced Properties**. The following screen is displayed (see Figure 7.53).

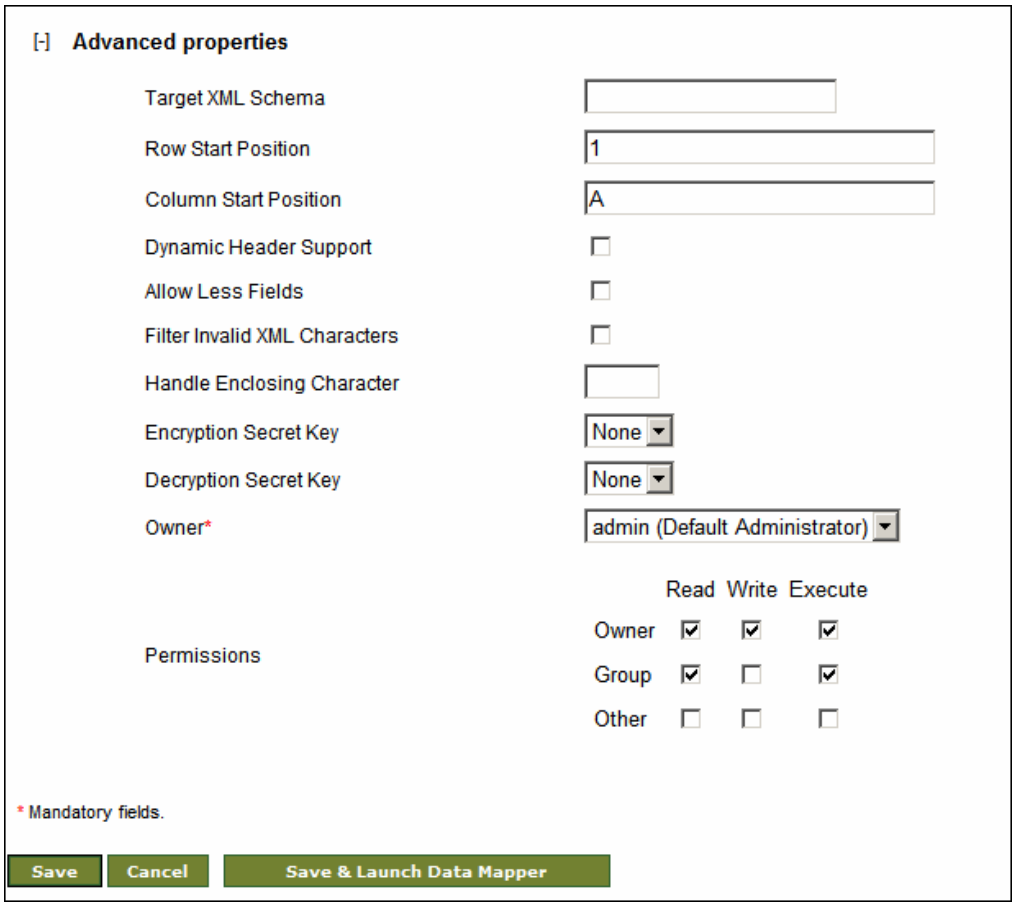


Figure 7.53: View Advanced Properties of Excel Schema

21. Enter the Row Start Position in the *Row Start Position* field. Row Start Position specifies which row of the Excel Sheet is counted as first row. For example if you entered 5 in the Row Start Position field, 5th row of the Excel Sheet is counted as first row. If this schema is used at source end, the data form 5th row onwards is taken for processing. If this schema is used at target end, data is copied into the 5th row onwards. From 1st to 4th row of the target excel sheet will remain blank.

22. Similarly, enter the Column Start Position in the *Column Start Position* field.

23. If you want to enable Dynamic Header Support, check the Dynamic Header Support check box. For detailed information about Dynamic Header Support refer to the section **Using Dynamic Header Support**.

24. In case the input data contains some characters that are invalid in XML, then this may result in the mapping getting aborted. You can filter these invalid XML characters by checking the *Filter Invalid XML Characters* checkbox.

25. If you want to remove enclosing characters (e.g. ‘ , “ , & , < , >) from the data file while parsing, enter the enclosing character that you want to remove, in Handle En*closing Character* field. Currently following enclosing characters are supported:

    - Single Quote (‘)

- Double Quote (“)
- Ampersand (&)
- Less than symbol (<)
- Greater than symbol (>)

26. If schema is used at target end and you want to convert the data from Plain Text to encrypted mode, select the secret key activity from the *Encryption Secret Key* drop-down list. The selected secret key activity is used to encrypt the data.

27. If schema is used at source end and you want to convert the data from Encrypted Mode to Plain Text, select the secret key activity from the *Decryption Secret Key* drop-down list. The selected secret activity is used to decrypt the data.

	To know, how to create secret key activity, refer to the *Creating Secret Key Activity* section in the *Administrator Guide*. To learn more about Advanced Properties refer to **Changing Advanced Properties** section

28. Click **Save** button. This displays a screen confirming that the excel schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the excel schema (refer to Figure 3.6).

	You can directly open the Data Mapper applet from the *Create Excel Schema* page. You can use this feature if you want to use the same excel schema in mapping, which you are creating. In this case you can only use XML Schema at target end. To launch Data Mapper applet, enter the ID of the XML schema that you want to use at target end, in *Target XML Schema* and click *Save and Launch Data Mapper.* To view the ID of XML Schema, in *Manage XML Schema* page, click on the XML Schema that you want to use. A 30 digit entity ID is shown in the view page.

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

	The comment should be at least 1 character in length.

30. Click **OK** to save the comments. This displays a screen confirming that the excel schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Defining Field Hierarchy

You can define hierarchy (parent-child relationship) between the records of an excel file using *Define Hierarchy* option of excel schema. To understand how to define

hierarchy lets assume that you have an excel file which contains records of insurance policies of families (see Figure 7.54).

ID	EMPFIRSTNAME	EMPLASTNAME	SSN	GENDER	DOB	ADDR1	ADDR2	CITY	STATE	COVERAGETYPE
7812	John	Smith	8989	M	4/11/1960	123 DummySt	Apt 12	Gowen	MI	1
	Mary	Smith	7871	F	1/1/1962	123 DummySt	Apt 12	Gowen	MI	1
	Kevin	Smith	1211	M	5/7/1989	89 Main St		Chicago	IL	4
2311	Larry	Johnson	8672	M	2/4/1967	123 DummySt	Apt 2A	Gowen	MI	1
	Samantha	Johnson	5430	F	11/8/1971	123 DummySt	Apt 2A	Gowen	MI	1
	Anna	Johnson	9231	F	9/10/1995	123 DummySt	Apt 2A	Gowen	MI	4
	Rea	Johnson	9231	F	3/15/1998	123 DummySt	Apt 2A	Gowen	MI	4

Figure 7.54: Policy Details

As you can see in the above table for ID 7812 there are there policies and similarly for ID 2311 there are four policies. While defining schema you can specify the criteria based on that the hierarchy can be defined. For example you can select that if the ID field is blank, merge the record in the previous record which is having ID. Similarly you can define the criteria for the matching record. For example if value of the ID field is matching with that of the previous record, merge the record with the previous record.

Define Hierarchy option works only when you define the schema using data file.

Steps to define hierarchy

1. While creating schema, select *Use Definition File* radio button and then select Data from the drop-down list.
2. Click **Upload File** button. The Schema File Upload screen is displayed (refer to Figure 7.51).
3. Click **Browse** and select the file, you want to upload. Path of the selected file is shown in the *Browse File* Field.
4. Now click **Upload File** button. Name of the uploaded file is shown in the *File Name* list and list of sheets of the excel file is shown (refer to Figure 7.52).
5. Specify the Start Row No. and Start Column No. of the sheets in respective fields and then click **Process Sheet(s)**. This will read the sheets and field names.

Start Row No. and Start Column No. specifies that from which row and Column onwards data should be fetched. For example if you have an excel file in which first 4 rows of sheet1 are blank. So in that case you have to specify 5 in sheet1 *Start Row No.* Another scenario can be that the data is there in first 4 row, but you don't want to fetched those records.

6. Once the processing is done, click **Finish** to close the Schema File Upload screen and return to create schema page.
7. Once the file is uploaded the *Sheet Name* field is converted into Drop-down list and all the sheet names of the selected excel file are populated in this drop-down list.

8. Select the sheet name from the *Sheet Name* drop-down list. All the fields of the selected sheet are populated.
9. To define the hierarchy, select *Define Hierarchy* checkbox and select the merge criteria from the *Merge Criteria* drop-down list.

	Currently two merge criteria are supported: **Matching Child Record**: Records are merged in case the value Key fields are matching. **Blank Child Record:** In case the value of key field is blank, the records are merged with the previous record which is having some value in the key field.

10. Select the field name from the *Merge Key* drop-down list on basis of which record are merged.
11. Click **Save** to save the excel schema.

Using Dynamic Header Support

Dynamic Header is an advanced feature of Excel Schema, which is used to parse an excel file, if:

- *FieldNames* defined in the Excel Schema and the Data Headers (Column Name) of the excel file are same but not in same order.
- *FieldNames* defined in the Excel Schema and the Headers of the excel file are not same. They may or may not be in same order.

Pre requisites

- Data Header must be present in the Excel file
- Data Type must be same in the Excel Schema and the excel file

To parse an excel file, whose Headers are same but not in order with the FieldNames of the Excel Schema, check the Dynamic Header Support checkbox in the Advanced Properties. Now the Excel Schema will parse the data from the respective columns.

To parse an excel file, whose headers are different and not even in order with the fieldnames of the Excel Schema, check the *Dynamic Header Support* checkbox in the Advanced Properties. Apart from this an XML file is used by the schema, which contains the mapping between the headers of the excel file and the fieldnames of the Excel Schema. Following is the sample XML (see Figure 7.55).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<HeaderMap type="Name">
<Map>
<SchemaHeader>Name</SchemaHeader>
<DataFileHeader>EMP_Name</DataFileHeader>
</Map>
<Map>
<SchemaHeader>DOB</SchemaHeader>
<DataFileHeader>EMP_DOB</DataFileHeader>
</Map>
<Map>
<SchemaHeader>Age</SchemaHeader>
<DataFileHeader>EMP_Age</DataFileHeader>
</Map>
<Map>
<SchemaHeader>Address</SchemaHeader>
<DataFileHeader>EMP_Address</DataFileHeader>
</Map>
</HeaderMap>
```

Figure 7.55: XML to map Field Name with Data Header

where:

SchemaHeader is the *FieldName* defined in the Excel Schema.

DataFileHeader is the name of the *Header* in excel file.

Excel Schema reads this XML file from process flow context. So you have to pass this XML file to the process flow context. The following figure shows a sample process flow to depict this scenario:

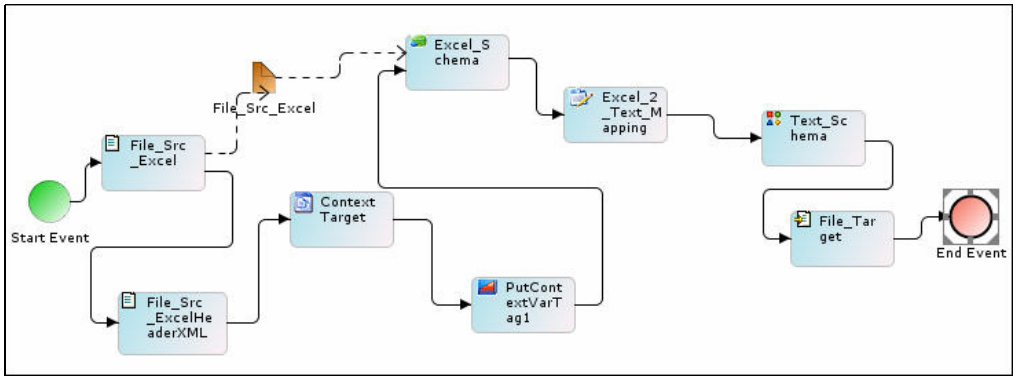


Figure 7.56: Sample Process Flow

In the process flow shown in Figure 7.56, a file source activity is used to read an excel file. An Excel Schema is used to parse the data from the excel file. FieldNames defined in the excel Schema and Headers in Excel file are different. To parse the data from the excel file, an XML file, which contains the mapping between FieldName and the Header is passed to the process flow context. To pass the XML file to process flow context, another File source activity *(File_Src_ExcelHeaderXML*) is used. This file source activity reads the XML file from a specified location and passes it to the context target activity. In context Target activity, you need to define the value of the *parameterName* property. Properties of the Context Target activity are shown in Figure 7.57:

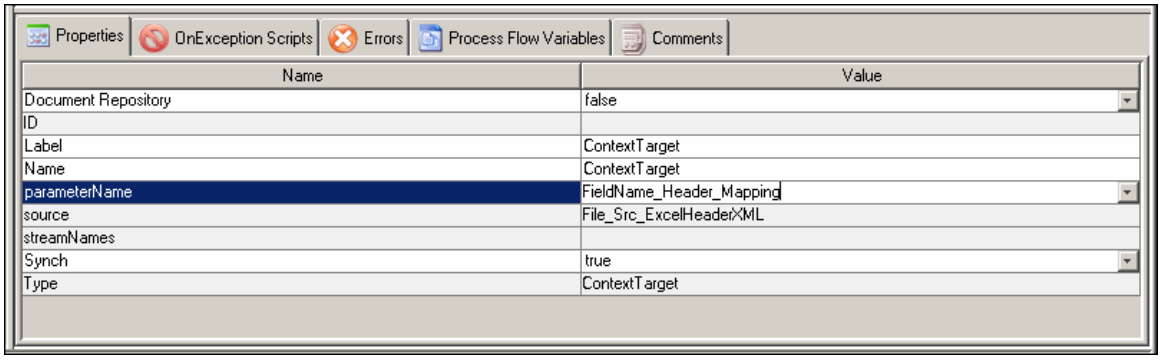


Figure 7.57: Context Target Properties

In this process flow, value of *parameterName* is given as *FieldName_Header_Mapping*.

After context target, *PutContextVar* action is used. This action is used to set the value of a variable *Service.<ActivityName>.excelHeaderMappingXML* of XML Schema. The properties of *PutContextVar* action are shown in Figure 7.58:

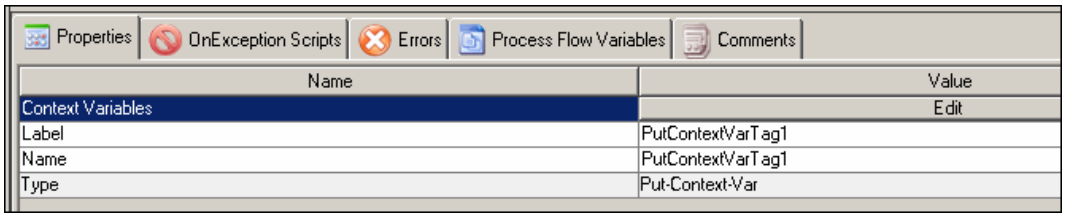


Figure 7.58: PutContextVar Properties

1. Click **Edit** to define the variable name and the value to be set. The Edit Context variable screen is displayed (see Figure 7.59).

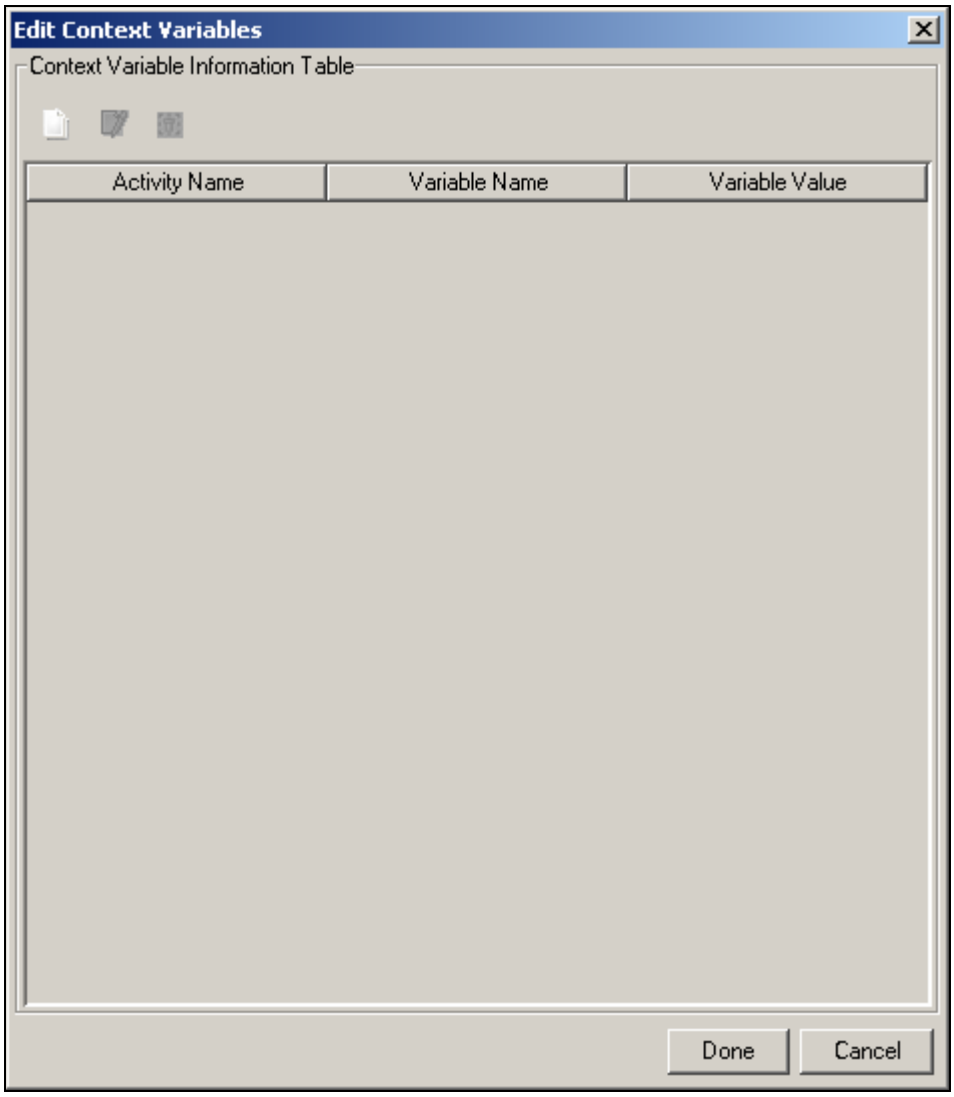


Figure 7.59: Edit Context Variables

2. Click **New** () icon. The Context Variable Information Dialog box is displayed (see Figure 7.60).

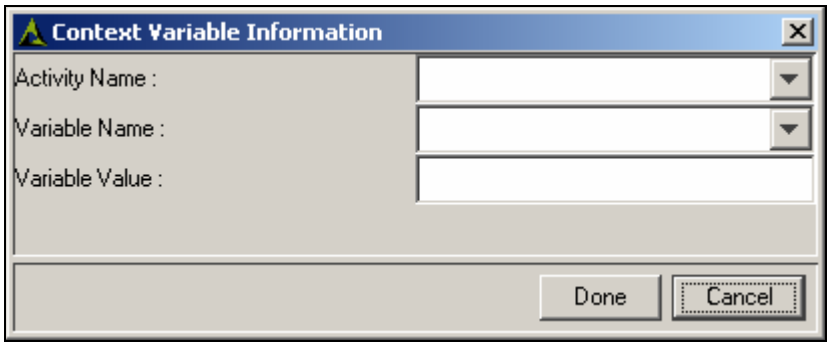


Figure 7.60: Context Variable Information

3. Enter *Service.<ActivityName>.excelHeaderMappingXML* in the *Variable Name* field. For Example *Service.Excel_Schema.excelHeaderMappingXML*

where

Activity Name is name of the Excel Schema Activity. For Example *FieldName_Header_Mapping* .

4. In the Variable Value field enter the value, which you defined in the parameterName in the ContextTarget activity.
5. Click Done twice to return to graph canvas area.
6. Make sure to create a stream from file source to Excel Schema activity.

	To Know how to create a process flow, refer to the **Creating Process Flow** section.

CREATING POSITIONAL SCHEMA ACTIVITY

The Positional Schema activity defines the procedure to read data from a Positional file, and write data in a Positional file. User needs to specify the names and the positions of required fields in order to enable identification of those fields.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create Positional schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Positional**. The Manage Positional Schema screen is displayed (see Figure 7.61).

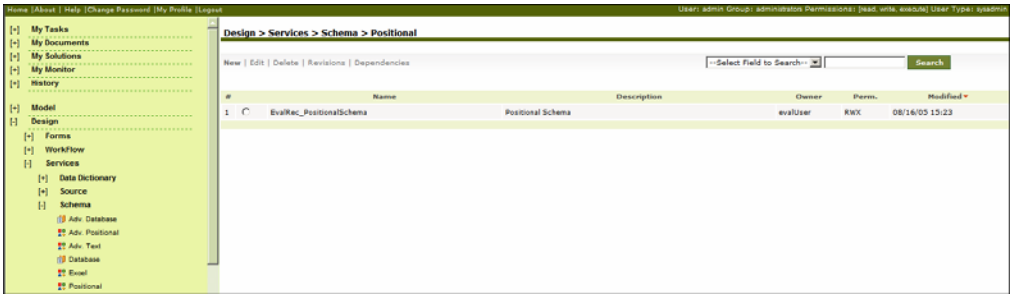


Figure 7.61: Manage Positional Schema

4. Click the **New** link. The Create Positional Schema screen is displayed (see Figure 7.62).

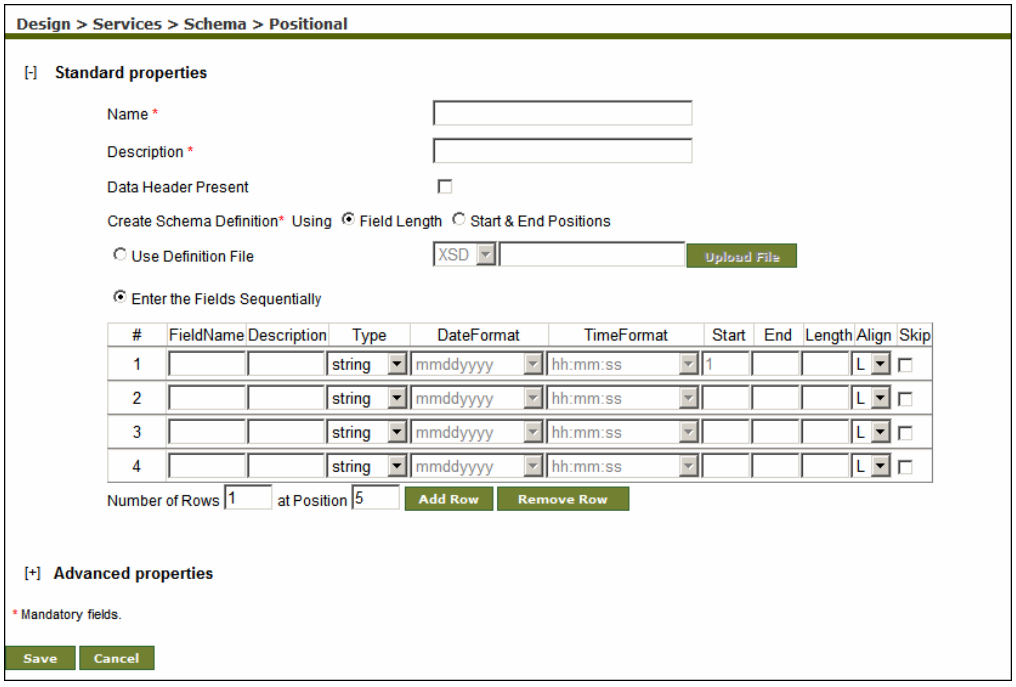


Figure 7.62: Create Positional Schema

5. Enter the name and description for new Positional Schema in the *Name* and *Description* fields respectively.
6. 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.

7. To define the schema using definition file, select the *Use Definition File* radio button; select the type of file from the drop-down list and click the **Upload File** button to select the required file. The Schema File upload screen is shown (refer to Figure 7.51).
8. Click **Browse** and select the file, you want to upload. Path of the selected file is shown in the *Browse File Field.*
9. Now click **Upload File** button. Name of the uploaded file is shown in the *File Name* list (see Figure 7.63).

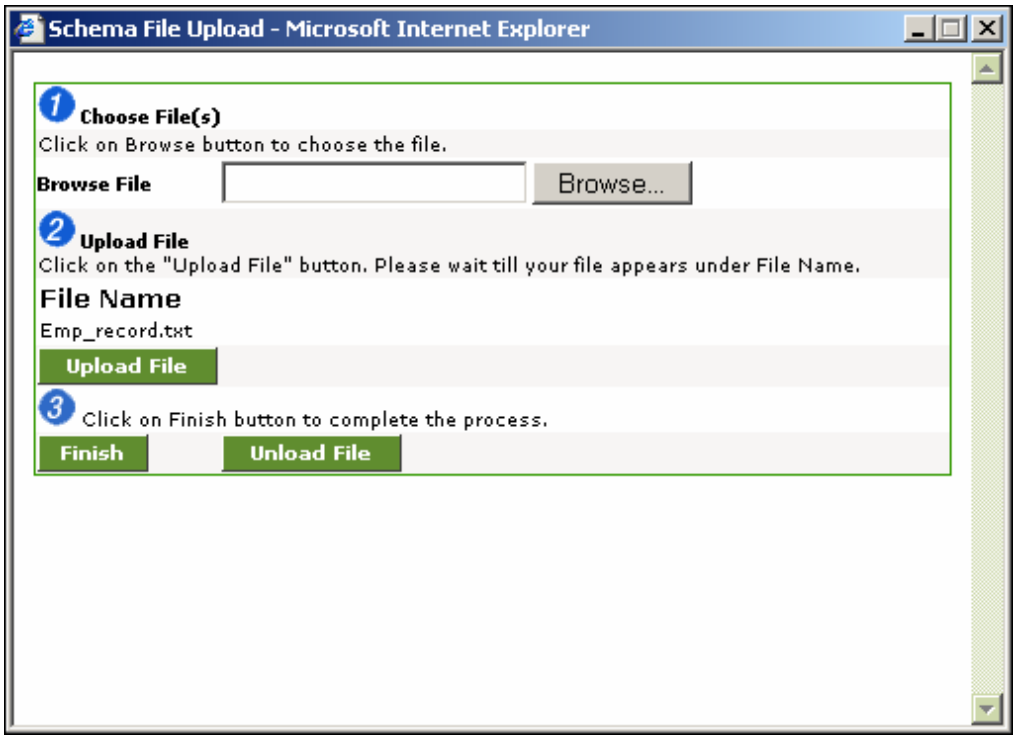


Figure 7.63:Uploaded File

10. Click **Finish** to close the Schema File Upload screen and return to create schema page.
11. If you want to create schema by entering the fields manually, select *Enter the Fields Sequentially* radio button and follow the steps given below:
12. Enter name and description of the field in the *Name* and *Description* fields respectively.
13. Select the type of data from the *Type* drop-down list. For data type selection, refer to Table 7.2.
14. If data type is *Date*, select the format of date and time from the *DateFormat* and *TimeFormat* drop-down lists respectively.
15. To define field position select one of the following option:
    - Field Length
    - Start & End Positions

16. To define the field position using field length, select *Field Length* radio button and enter the length of the field in the *Length* field.
17. To define the field position using start and end position, select the *Start and End Position* radio button.
18. Enter the start position of the field in the *StartPos* field.
19. Enter the end position of the field in the *EndPos* field.

	The starting position of a row in a positional file is 1. In a positional file, tab is counted as one position and not eight positions. By default, field positions are created in sequence. You can also create a schema with fields that are not in sequence. For details, refer to the Defining Field Positions Non-Sequentially section.

20. Select the alignment of the field from the *Align* drop-down list.

	From **Align** select **L** if the field is left aligned. **R** if the field is right aligned. To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. Maximum 99 rows can be added at a time. To remove rows, specify the number and position of the rows to be deleted in the *Number of Rows* and at *Position* fields respectively and click **Remove Row** button. To learn about Advanced Properties refer to **Changing Advanced Properties** section

	If you delete the last fieldname of a record, and then save the schema, the entire field gets deleted. For example, a record has three fields – *Name, Description* and *Age*. If you delete *Age*, the entire field will get deleted.

21. Check the *Skip* checkbox if you want to skip this field while generating the XML. This selection skips the fields that are not required for the schema. For example, the source file has over 1500 fields, but you just need to use 1000 fields. This selection skips the 500 unrequired fields, and does not read them, when the data is parsed to the XML. When the data file is created, the skipped fields are displayed in the file but are not read. If the schema is created using an existing XSD, the skipped fields will appear as a blank value. However, when the schema is used in other activities such as Mapping, all its fields are displayed.

	The skipping of unrequired fields is useful in case of a standard XSD with a large number of fields, as it reduces the size of the generated XML which now contains only the required fields. While editing the schema, when the data file is downloaded, the skipped

	fields are represented by 'T' and the unskipped fields by 'F' (refer to Figure 7.15). Similarly, while viewing the *Print-Friendly Page*, the skipped fields are represented by 'T' and the unskipped fields by 'F'.

22. Click **[+]** to expand **Advanced Properties**. Advanced properties of positional schema are displayed (see Figure 7.64).

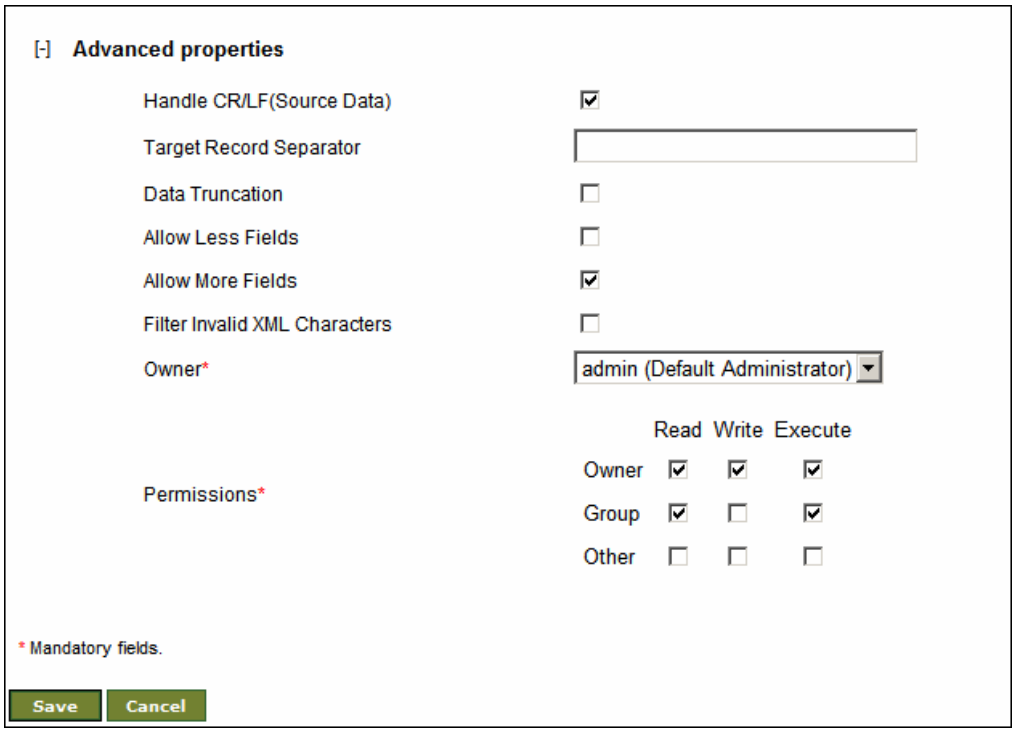


Figure 7.64: Advanced Properties of Positional Schema

23. Uncheck the *Handle CR/LF (Source Data)* checkbox, if the source file does not have any carriage return. By default, this option is checked, and schema expects file with carriage return. Handle CR/LF (Source Data) option is applicable only for the schema used at the source end.

24. Enter the target record separator in the *Target Record Separator* field, if you want to write each record in new line in target file. By default records are written in single line. This option is applicable only for the schema used at target end.

25. Check the *Data Truncation* checkbox, in case the data length is more than specified in the schema and you want to pass the specified length of data and ignore the rest of the data.

26. Check the *Allow Less Fields* checkbox, if you want to parse the data even if the number of fields in the data file is less than the number of field specified in the schema. If Allow Less Fields checkbox is checked and the schema is used at source end, schema will parse the input data and insert the empty tag of missing fields. If the schema is used at target end, it will write all the tags coming in input XML.

27. Incase number of fields in source data is more than the number of fields specified in the schema, only those fields are parsed, which are specified in schema. Other fields are ignored. If you want to generate error records, when

number of fields in source data is more than the number of fields specified in schema, uncheck the *Allow More Fields* checkbox.

28. In case the input data contains some characters that are invalid in XML, then this may result in the mapping getting aborted. You can filter these invalid XML characters by checking the *Filter Invalid XML Characters* checkbox.
29. Click **Save** button. This displays a screen confirming that the positional schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the positional schema (refer to Figure 3.6).
30. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

31. Click **OK** to save the comments. This displays a screen confirming that the positional schema activity has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING TEXT SCHEMA ACTIVITY

The Text Schema activity is used to define how a text file is to be read or written in a predefined format. To create a Text Schema activity, you need to specify the format of text file.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a text schema activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (see Figure 7.65).

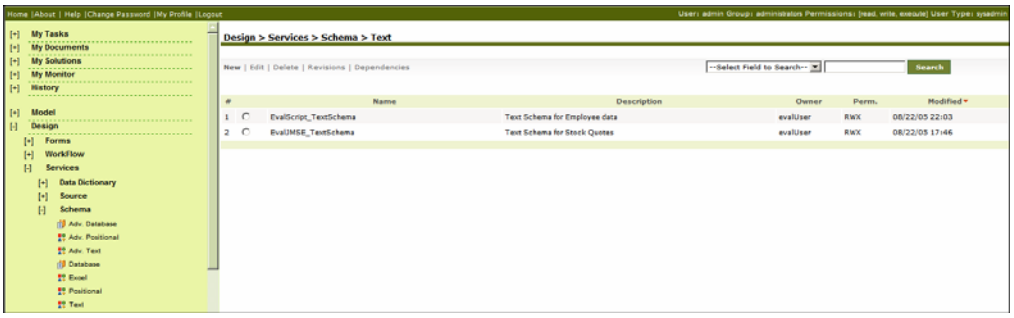


Figure 7.65: Manage Text Schema

4. Click the **New** link. The Create Text Schema screen is displayed (see Figure 7.66).

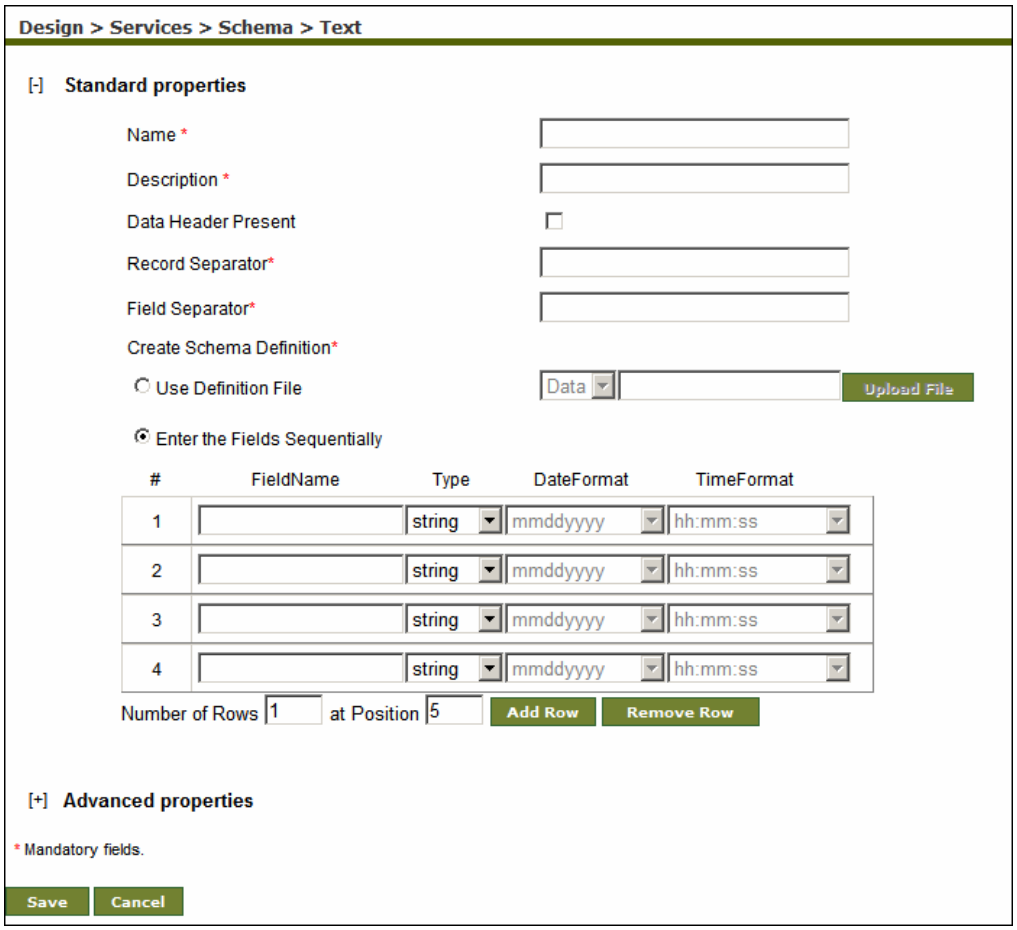


Figure 7.66: Create Text Schema

5. Enter the name and description of the new text schema in the *Name* and *Description* fields respectively.

6. 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.

7. Enter the record separator, for example \n for new line or " " for space, in the *Record Separator* field. The record separator is used to separate records.
8. Enter the Field Separator, for example \t for Tab or " " for space. In the *Field Separator* field. Field Separator is used to separate fields.
9. To define the schema using definition file, select the *Use Definition File* radio button; select the type of file from the drop-down list and click the **Upload File** button to select the required file. The Schema File upload screen is shown (refer to Figure 7.51).
10. Click **Browse** and select the file, you want to upload. Path of the selected file is shown in the *Browse File* Field.
11. Now click **Upload File** button. Name of the uploaded file is shown in the *File Name* list (see Figure 7.67).

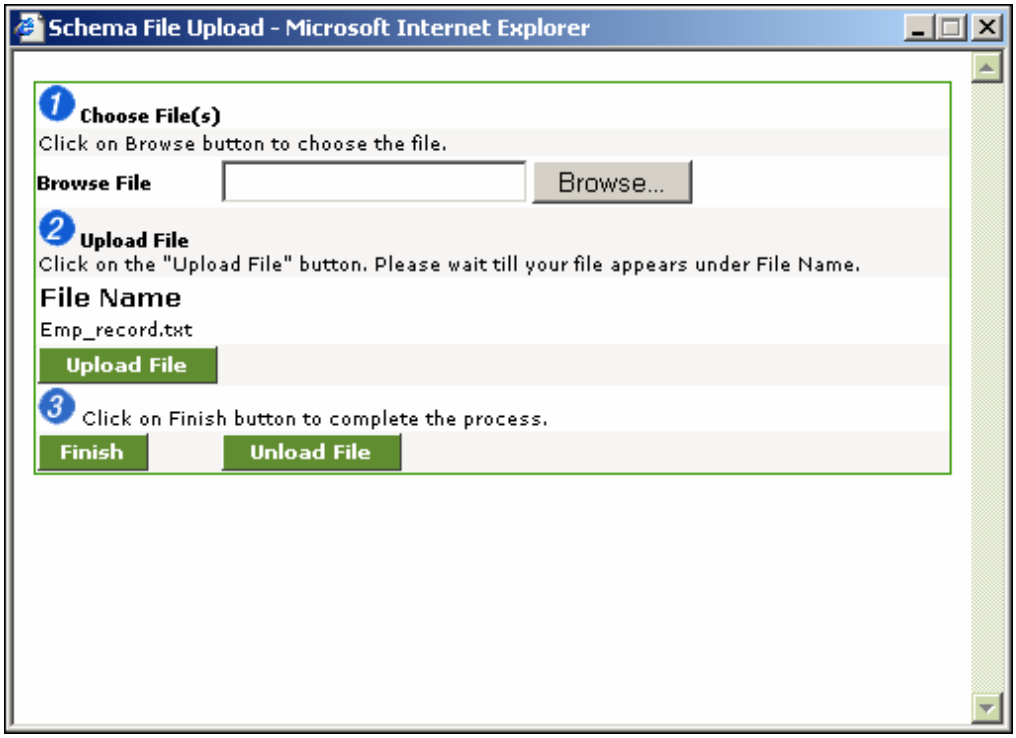


Figure 7.67:Uploaded File

12. Click **Finish** to close the Schema File Upload screen and return to create schema page
13. If you want to create schema by entering the fields manually, select *Enter the Fields Sequentially* radio button and follow the steps given below:
14. Enter the name of each field in the *FieldName* field.
15. Select the type of data from the *Type* drop-down list. For datatype selection, refer to Table 7.2.

	The Currency datatype is supported by Excel Schema only.

16. If data type is *Date*, select the format of date and time from the *DateFormat* and *TimeFormat* drop-down lists respectively.

	To insert rows, specify the number and position of the rows to be added in the *Number of Rows* and at *Position* fields respectively and click **Add Row** button. Maximum 99 rows can be added at a time. To remove rows, specify the number and position of the rows to be deleted in the *Number of Rows* and at *Position* fields respectively and click **Remove Row** button.

	If you delete the last fieldname of a record, and then save the schema, the entire field gets deleted. For example, a record has three fields – *Name, Description* and *Age*. If you delete *Age*, the entire field will get deleted.

17. To enable quotes handling, click **[+]** to expand the **Advanced Properties** and check the *Quotes Handling On* checkbox.

	If a character (say $) is specified as Field Delimiter in a record, then any $ character in the field data of that record (Chocolate20perpack) will be considered as a Field Delimiter, even though it is part of the field data. In the above example the $ after 20 will also be considered as Field Delimiter, whereas it is the data. To avoid this situation put that field within the double quote i.e. (Chocolate$“20$perpack”)

18. If you want to remove enclosing characters (e.g. ‘ , “ , & , < , >) from the data file while parsing, enter the enclosing character that you want to remove, in *Handle Enclosing Character* field. Currently following enclosing characters are supported:
    - Single Quote (‘)
    - Double Quote (“)
    - Ampersand (&)
    - Less than symbol (<)
    - Greater than symbol (>)
19. In case the input data contains some characters that are invalid in XML, then this may result in the mapping getting aborted. You can filter these invalid XML characters by checking the *Filter Invalid XML Characters* checkbox in Advanced Properties.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section

20. Click **Save** button. This displays a screen confirming that the text schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the text schema (refer to Figure 3.6).
21. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING WORD SCHEMA ACTIVITY

Word Schema is used to convert MS Word file into XML format, so that it can be further processed. Word Schema uses a pre-built template XSD XMSW.xsd, which defines the structure of the XML file. When a word schema is created, this XSD is stored into the backend database to define the structure of the schema. Word to XML conversion is done at execution time.

	This feature is a paid service and is thus not available in any of the Adeptia products by default.

Steps to create Word Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to **expand** the tree, and then click **Word**. The Manage Word Schema screen is displayed (see Figure 7.68).

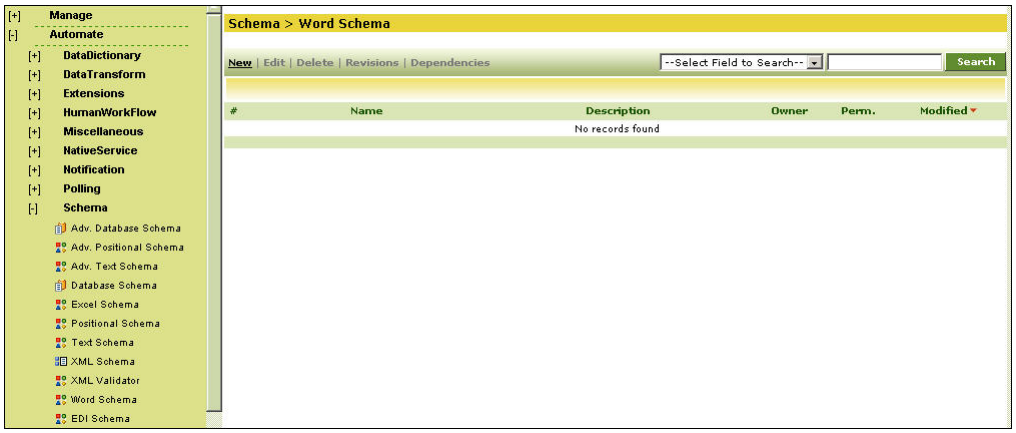


Figure 7.68: Manage Word Schema

4. Click the **New** link. The Create **Word** Schema screen is displayed (see Figure 7.69).

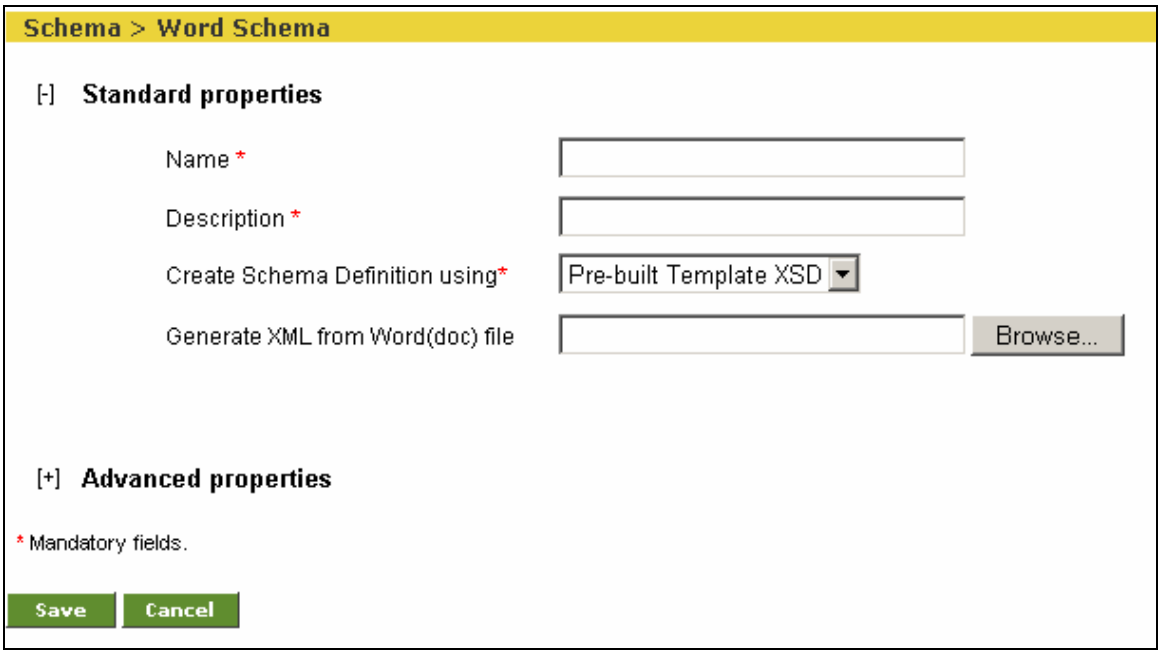


Figure 7.69: Create Word Schema

5. Enter the name and description for Word Schema in the *Name* and *Description* fields respectively.
6. Leave the Create Schema Definition Using field as default.
7. If you want to view the XML of the word file, which you want to convert, click the **Browse** button and select the required word file.

	This field is not mandatory because at execution time, Word file is taken from the File Source activity. This field is required only if you want to view the XML of the Word file. To view the XML of the Word file, first browse and select the required word file. Save the Word Schema activity and then edit the word schema activity that you have created. Click **Download** in the Edit Word Schema screen. The Download Word Schema Definition File screen is displayed (see Figure 7.70).

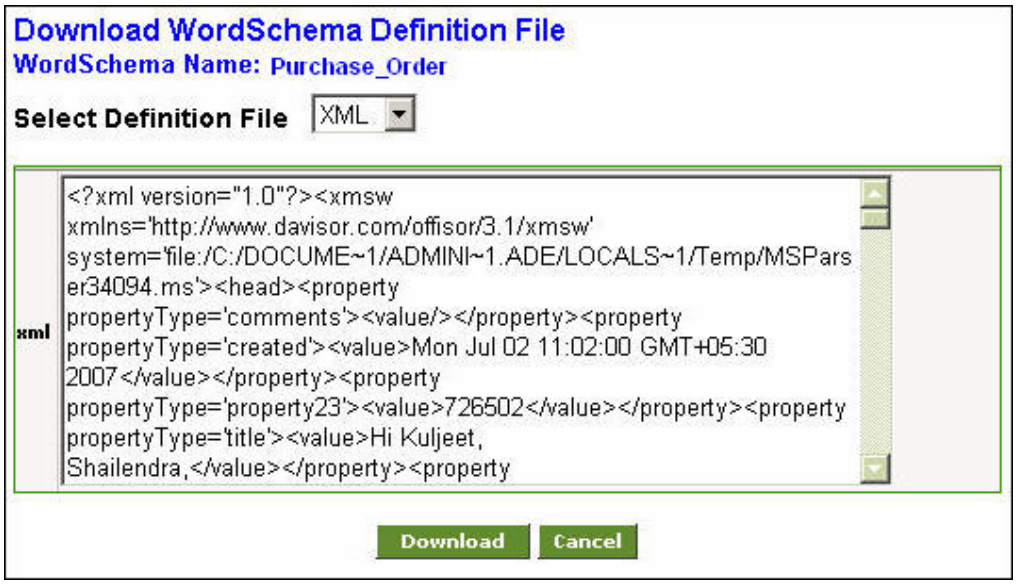


Figure 7.70: Download Word Schema Definition File

	To download the XML, select XML from the *Select Definition File* drop-down list and click **Download**. To download the word file select Word from the select *Definition File* drop-down list and click **Download**. Click **Cancel** to close the Download Word Schema Definition File dialog box.

	To learn more about Advanced Properties refer to **Changing Advanced Properties** section

8. Click **Save** button. This displays a screen confirming that the word schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the Word schema (refer to Figure 3.6).
9. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

10. Click **OK** to save the comments. This displays a screen confirming that the Word schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING XML SCHEMA ACTIVITY

The XML Schema activity defines the procedure to read data from an XML file, and write data in an XML file. To do so, user needs to specify the schema definition location.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create XML Schema

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **XML**. The Manage XML Schema screen is displayed (see Figure 7.71).

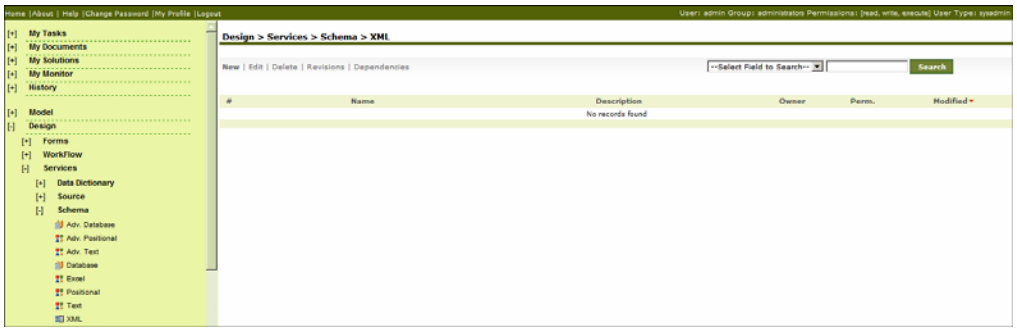


Figure 7.71: Manage XML Schema

4. Click the **New** link. The Create XML Schema screen is displayed (see Figure 7.72).

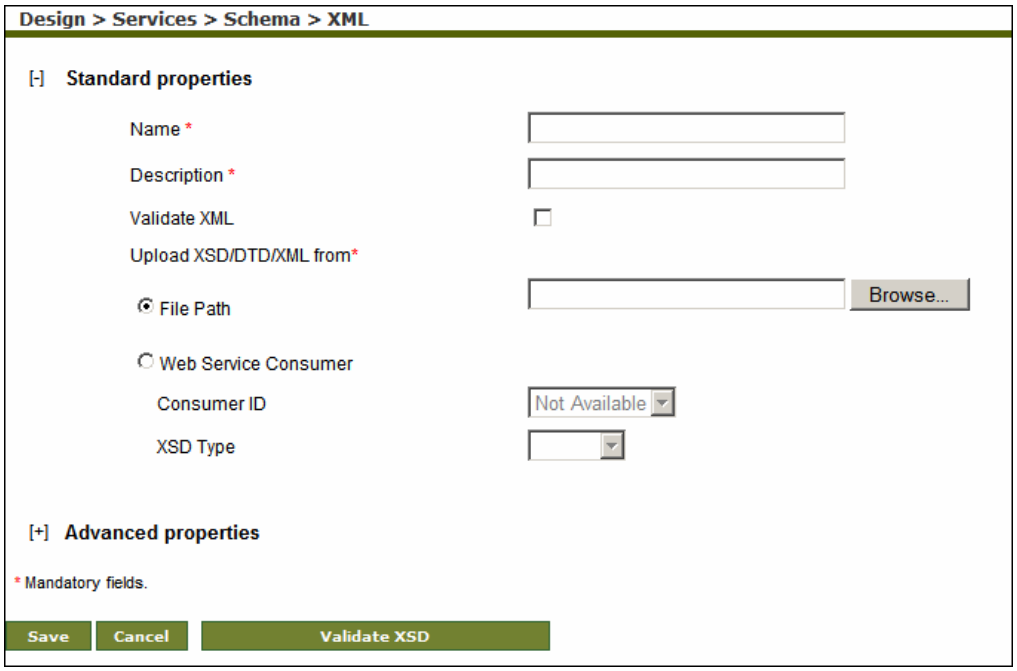


Figure 7.72: Create XML Schema

5. Enter the name and description for XML Schema in the *Name* and *Description* fields respectively.
6. Select one of the following Schema Definition Location:
   - File Path
   - Web Service Consumer
7. Select the desired option in the *Upload XSD/DTD/XML from* field. The selection process of options is explained in the table below.

Table 7.4: Selection Process of Value for Upload XSD/DTD/XML from Field

Option	Process
File Path	Click the **Browse** button and select the DTD, XSD or XML file. If the DTD, XSD or XML file is located on an HTTP URL, enter the URL, where DTD/XSD/XML file is located in the *HTTP URL* field.

Web Service Consumer	In this case, XSD is picked up from pre-defined Web Service Consumer activity. Each WS Consumer activity has two XSD's associated with it (i.e. Input Data XSD and Output Data XSD). Select the Web Service Consumer activity from the *Consumer ID* drop-down list. Select the type of XSD as either Input or Output from the *XSD Type* drop-down list. ▪ Input: XSD for input data format. ▪ Output: XSD for output data format.

Check the *Validate XML* checkbox if you want to validate the XML during execution. It validates the input XML file at runtime.

8. Click **Validate XSD** button to validate file entered in the Filepath field. You can validate all file types such as XML, XSD or DTD. If it is not an XSD file, then it will convert it to XSD format and then validate.
9. Click **[+]** to expand **Advanced Properties**. The Advanced Properties of XML Schema activity is displayed (see Figure 7.73).

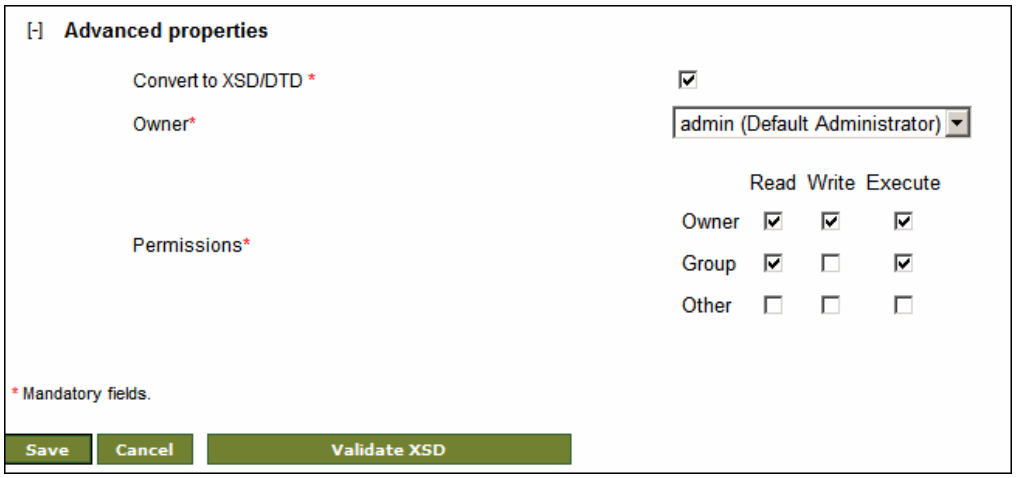


Figure 7.73: Advanced Properties of XML Schema

10. The *Convert to XSD/DTD* checkbox is marked as checked. This implies that all xml file will be converted to XSD or DTD. The XSD file remains as XSD file whereas, DTD file is converted to XSD file. If you don't want to convert the uploaded XML file into DTD, uncheck this checkbox.

XML Schema converts the XML file into DTD before passing it to Mapping activity. Sometime large DTD causes error while loading into Data Mapper. In this case you can uncheck the *Convert to XSD/DTD* checkbox. XML Schema will pass the XML file as it is to Mapping activity without converting them into DTD.

Sometime, in case of recursive XML Schema, proper hierarchy may not be

	displayed, while loading the schema in Mapper. This is because of the limitation is JDOM. In this case, check the *Convert to XSD/DTD* checkbox. To learn more about Advanced Properties refer to **Changing Advanced Properties** section.

11. Click **Save** button. This displays a screen confirming that the XML schema activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the XML schema (refer to Figure 3.6).
12. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the XML schema activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

TESTING SCHEMA ACTIVITY

Once a schema is created, you can verify the schema activity. Schema can be tested for both source and target end. In order to test a schema you have to supply the source file, and schema will show you the output file and the error file if any. When a schema is tested, it parses the supplied file as per the structure defined in the schema and generates the output and error files. The output file contains the record which are parsed successfully error file contains the record which are not parsed. The process of testing a text schema is explained below.

Steps to verify text schema activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 7.65).
4. Select the radio button adjacent to required schema activity that you want to test and click **Edit** link. This displays the Edit Text Schema screen (see Figure 7.74).

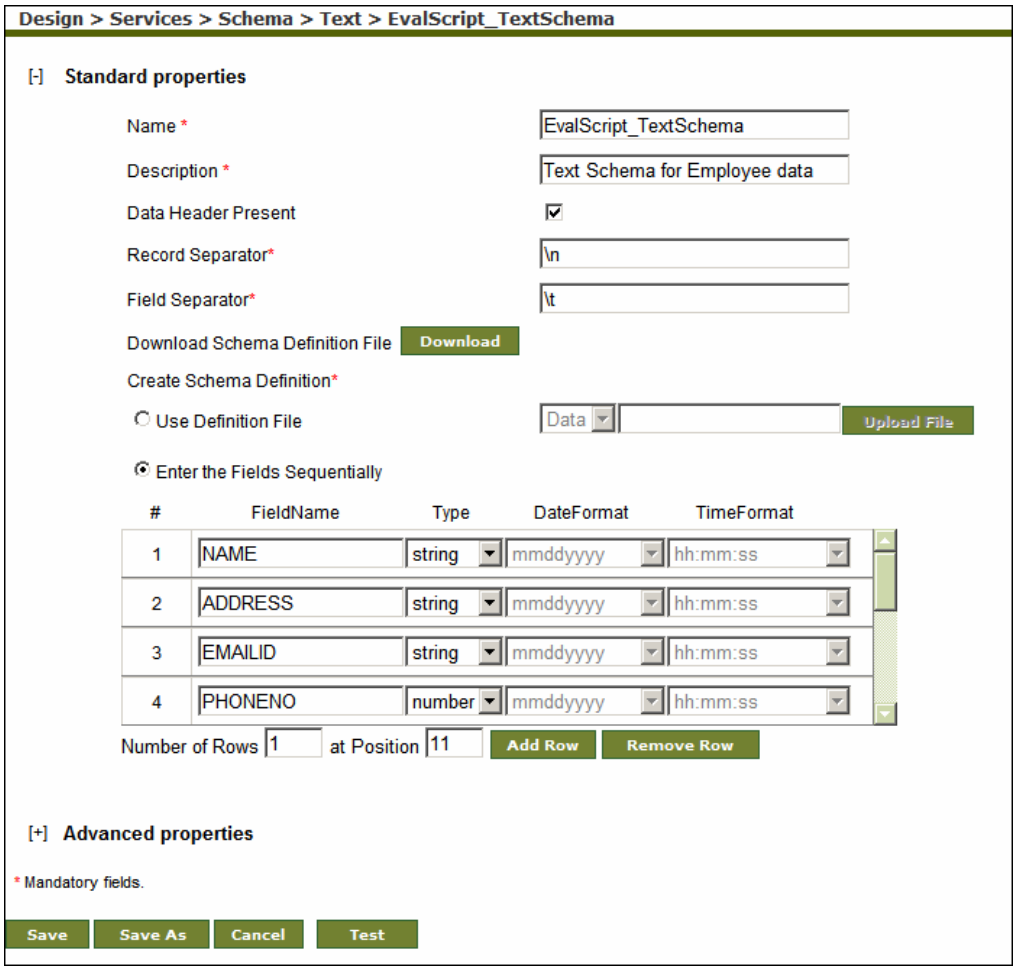


Figure 7.74: Edit Text Schema

5. Click **Test** button. The Test Schema screen is displayed (see Figure 7.75).

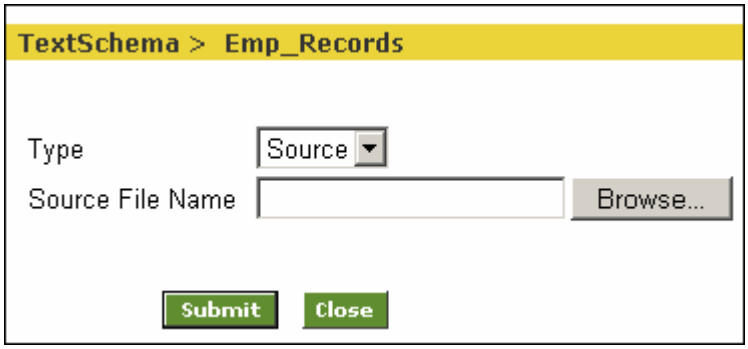


Figure 7.75: Test Schema

6. Select the type of schema to test, from the *Type* drop-down list. By default, Source is selected.

7. Click **Browse** button and select the file which you want to test.

	The file extension in this field will vary based on the schema being tested. If a text schema is being tested, then the file will have *.txt* extension. If an excel schema is being tested, then the extension will be *.xls*.

	If you want to test this schema for the target end, select *Target* from the *Type* drop-down list. Now the source file will be an xml file that is generated by the mapping activity used in the process flow. In this case target file will be the file which will be generated by schema as output. For example if your are testing Text Schema which will be used at target end, the target file will a .txt file.

8. Click **Submit** button. This tests the specified source file according to the defined schema and gives you the link of out files generated (see Figure 7.76).

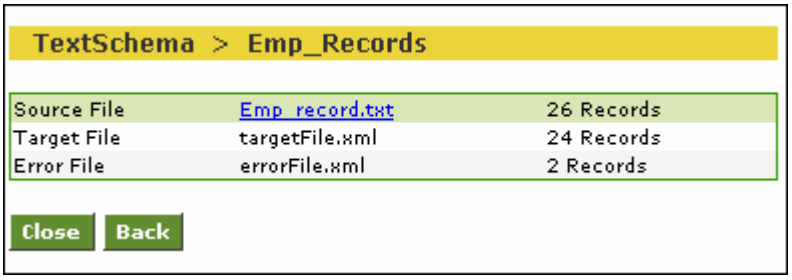


Figure 7.76: Test Schema

	Once you click the **Submit** button, there files are generated: *Source File: File which you have selected* *Target File: File which is generated by schema as output* *Error File: File which contains erroneous record if any.*

9. To view the file click on the link of file you want to view.

	All schemas except Advance Database schema, Database schema and XML Schema can be tested.

CREATING XML VALIDATOR ACTIVITY

XML Validator is used to validate the whole XML file or part of it. For validation user needs to supply the Xpath from which part of the XML starts and the path of the XSD against which that XML file is validated.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite

√		√	√

Steps to create a XML Validator activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **XML Validator**. The XML Validator screen is displayed (see Figure 7.77).

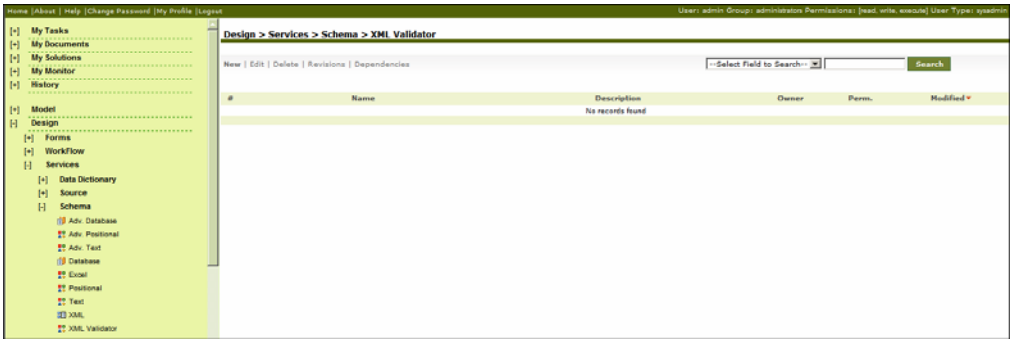


Figure 7.77: Manage XML Validator

4. Click the **New** link. The Create XML Validator screen is displayed (see Figure 7.78).

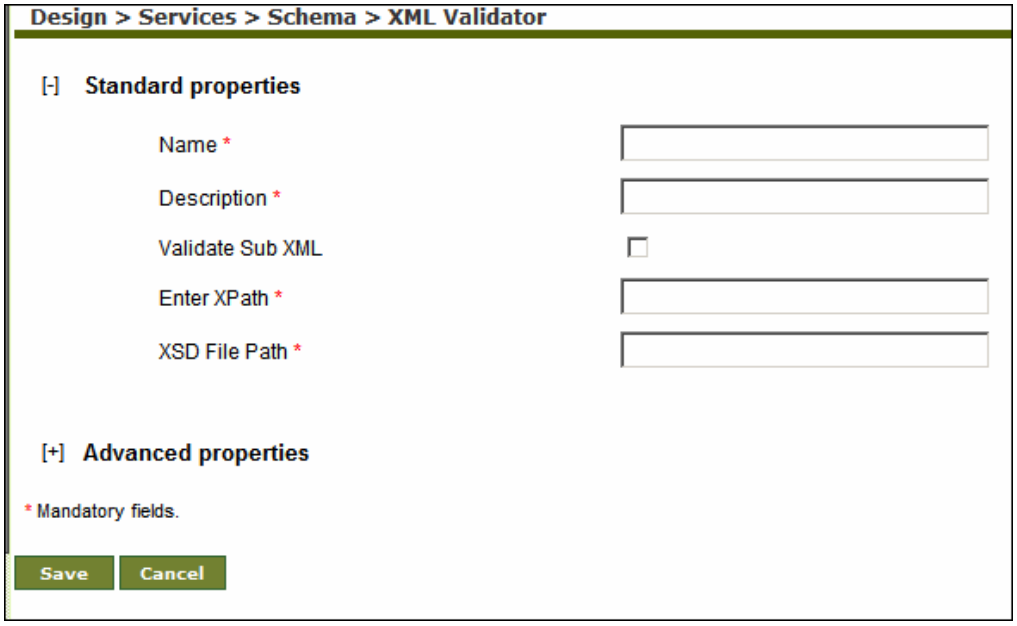


Figure 7.78: Create XML Validator

5. Enter the name and description of new XML Validator activity in *Name* and *Description* fields respectively.
6. If you want to validate the Sub XML, check the *Validate Sub XML* checkbox.
7. Enter the XPath of the XML, which needs to be validated in the *Enter XPath* field.
8. Enter the path of the XSD file, against which the above defined XML is validated, in the *XSD File Path* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. This displays a screen confirming that the XML Validator activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the XML Validator (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the XML Validator activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

8 CREATING TARGET ACTIVITY

A Target Activity allows you to identify specific data to be created and the means to deliver it. The Target activity screen allows you to identify specific data set (i.e., flat file, XML or database record) to be created, its location, and the transport protocol to be used to send this data. The Adeptia Suite allows following types of Target activities:

- **Advanced Database Target**
- **Database Target**
- **File Target**
- **FTP Target**
- **HTTP Target**
- **JMS Target**
- **LAN File Target**
- **Mail Target**
- **WebDAV Target**
- **Context Target**

Context Target is used to put the data in process flow context. To know how to use Context Target refer to the section **Using Context Source and Context Target** activity.

CREATING ADVANCED DATABASE TARGET ACTIVITY

The Advanced Database Target activity provides the ability to specify multiple tables of a database as Target.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity and *Advanced Database Schema* must be created before creating Advanced Database Target activity.

Steps to create an Advanced Database Target Activity

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

2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **Adv.Database**. The Manage Advanced Database Target screen is displayed (see Figure 8.1).

Figure 8.1: Manage Advanced Database Target

4. Click the **New** link. The Create Advanced Database Target screen is displayed (see Figure 8.2).

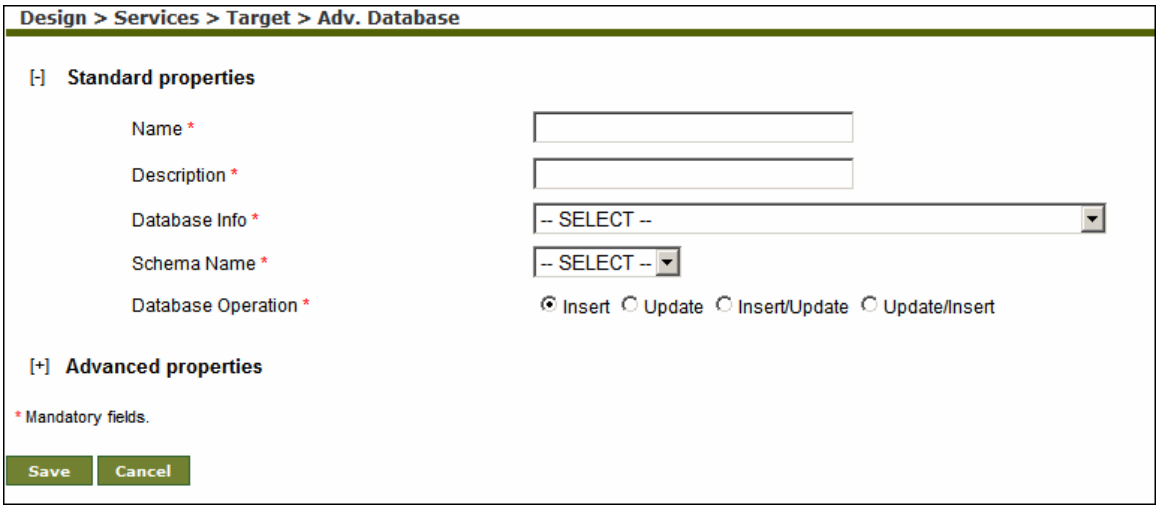


Figure 8.2: Create Advanced Database Target

5. Enter the name of the new Advanced Database Target in the *Name* field. Then, enter the description for the Advanced Database Target in the *Description* field.
6. Select the database info activity and database schema activity from the *Database Info* and *Schema Name* drop-down lists respectively.
7. Database Operation specifies how data records are inserted into database tables. Select one of the following database operations:
   - Insert
   - Update
   - Insert/Update
   - Update/Insert

	If the specified database operation fails on any source data, error records are created. While creating a process using an Advance Database Source, you can specify what to do with error records.

	▪ Error records can be saved into repository file. ▪ Error records can be ignored. ▪ It can be further processed (e.g. can be sent to file target). ▪ Process Flow can be aborted, if there is any error record. In process designer, right click on the advance database source activity and select view properties. Select the value of the "Error Record" properties. To learn, how to use Process Designer, refer to the section Using Process Designer. To learn how to create Database Info activity, refer to the *Creating Database Info* section in the *Administrator Guide*. To learn how to create the Advance Database Schema activity, refer to the Creating Advance Database Schema Activity section.

	When an excel file is used as a database target, only *Insert* operation is supported. The *Update, Insert/Update* and *Update/Insert* operations are not supported. When an excel file is used as a database target, the error records are not generated in case of data type mismatch.

When a database operation is selected, the database is affected. For details, refer to Table 8.2.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

8. Click **Save** button. This displays a screen confirming that the Advanced Database target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the advanced database target (refer to Figure 3.6).
9. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

10. Click **OK** to save the comments. This displays a screen confirming that the Advanced Database target activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Understanding Record Count in Process Flow Logs

The number of records processed by an Advanced Database Target activity during the execution of a process flow is displayed in the Process Flow Log. This record count differs based on its type. The various types of record counts are explained below.

Transaction Commit Count

This record count type includes the total number of records in the input at root level, which are successfully processed and committed by the Advanced Database Target activity. It includes inserted, deleted and updated records.

Insert Query: Submitted / Successful

This record count type includes the Insert Query which are submitted or successful. Submitted is the total number of Insert Query submitted/executed by the Advanced Database Target activity.

Successful is the total number of records, which are inserted and committed in the table for the Insert Query and which are not rolled back.

The 'Successful' count can be equal or less than the 'Submitted' count in case of "Insert" operation.

Delete Query: Submitted / Successful

This record count type includes the Delete Query which are submitted or successful.

Submitted is the total number of Delete Query submitted/executed by the Advanced Database Target activity.

Successful is the total number of records, which are deleted and committed from the table for the Delete Query and which are not rolled back.

Update Query: Submitted / Successful

This record count type includes the Update Query which are submitted or successful.

Submitted is the total number of Update Query submitted/executed by the Advanced Database Target activity.

Successful is the total number of records, which are updated and committed in the table for the insert query and which are not rolled back.

The 'Successful' count can be greater than the 'Submitted' count in case of "Update" or "Delete" operation.

Rollback Transaction Count

This record count type includes the total number of records in the input at root level, which are rolled back by the Advance Database Target activity.

Rollback Insert Count

This record count type includes the total number of records, which are supposed to be inserted but rolled back due to any error.

Rollback Delete Count

This record count type includes the total number of records, which are supposed to be deleted but rolled back due to any error.

Rollback Update Count

This record count type includes the total number of records, which are supposed to updated but rolled back due to any error.

To further understand the different counts, assume the following example.

R1 (10)	R1C1 (20)	R1C1C1 (100)
R2 (20)	R2C1 (40)	R2C1C1 (80)

Suppose there are two records *R1* and *R2* at root level of input. *R1* has a child *R1C1* and further *R1C1* has a child *R1C1C1*. Similarly *R2* has a child *R2C1* and further *R2C1* has a child *R2C1C1*.

Their number of records is displayed in the brackets. For these records, insert queries are issued by the Advanced Database Target activity. Records from *R1*, its child (*R1C1*) and sub-child *(R1C1C1*) are successfully inserted in the target table. At the same time, while inserting the records from *R2C1C1* the Advance Database Target activity encountered an error in 8 records. Now, the record of *R2* and its child (*R2C1*) corresponding to erroneous records will be rolled back. Thus, the number of records rolled back from *R2C1C1, R2C1, R2* will be 8, 4 and 2 respectively.

Thus, in this scenario, the values of the record count types will differ as displayed below.

Table 8.1: Record Count Type Values

Record Count Type	Value	Description
Transaction Commit Count	28	10 records from R1 and 18 records from R2
Insert Query: Submitted/Successful	270/256	Total number of records (270) – Total number of rolled back records

		(14)
Rollback Transaction Count	2	Number of records rolled back from root level
Rollback Insert Count	14	R2 (2) + R2C1 (4) + R2C1C1 (8)

All the other counts will be zero.

CREATING DATABASE TARGET ACTIVITY

The Database Target activity provides the ability to specify a database as target.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity and *Database Schema* must be created before creating Database Target activity.

Steps to create a Database Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **Database**. The Manage Database Target screen is displayed (see Figure 8.3).

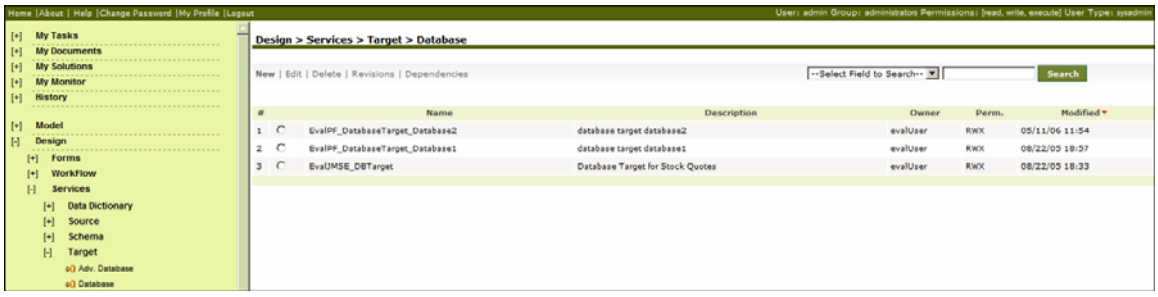


Figure 8.3: Manage Database Target

4. Click the **New** link. The Create Database Target screen is displayed (see Figure 8.4).

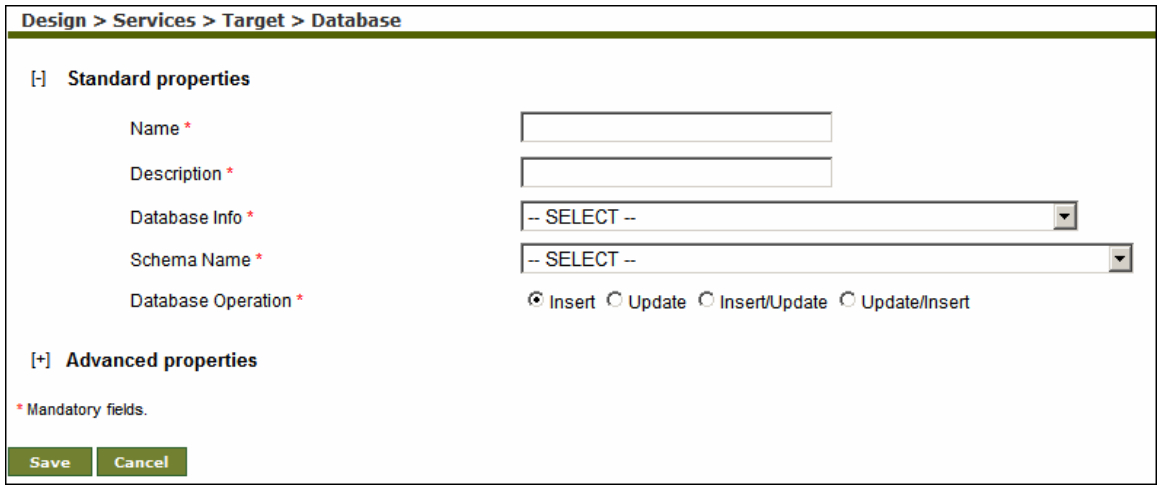


Figure 8.4: Create Database Target

5. Enter the name and description of the new Database Target in the Name and Description fields respectively.

6. Select the database info activity and database schema activity from the *Database Info* and *Schema Name* drop-down lists respectively.

	To learn how to create Database Info activity, refer to *Creating Database Info* section in *Administrator Guide*. To learn how to create Database Schema activity, refer to the Creating Database Schema Activity section.

	Database target can insert/update current date in the database. This current date is added in the input XML in two ways: ▪ Source XML ▪ Mapping It can be added through the source XML which has the current date already defined in its code. It can be added through mapping, if current date is defined as a constant in the Date field, or as a variable which has the value “Current Date”. In this case, when mapping is done, then database target accesses the current date and inserts/updates it accordingly.

7. Database Operation specifies how data records are inserted into database tables. Select one of the following database operations:

   - Insert
   - Update
   - Insert/Update
   - Update/Insert

   The effect that the selection has on the database is listed in the table below.

Table 8.2: Effect on Database based on the selected Database Operation

Database Operation	Effect on Database
Insert	This option is selected when you want to insert records into the database tables. If records already exist in the database table, new records are added in the database table along with existing records. When a column e.g. CompanyName of database table is marked as Primary Key, more than one record cannot exist in the database table for the same Company Name. In this case if data of a company i.e. Microsoft Corporation already exist, insert operation fails and data is not inserted into the database.
Update	This option is selected when you want to update the existing record. To use updated option a column of database must be marked as Primary Key. When Update option is selected, database target first checks which column of the database table is marked as Primary Key. Suppose CompanyName column is marked as Primary Key. Now database target check whether data of a company e.g. Microsoft Corporation already exist or not. If data for Microsoft Corporation already exists, database target updates the existing records. If none of the column is marked as Primary key, Update operation fails.
Insert/Update	This option is selected when the database target first tries to insert the data into database table. If insert operation fails, database target tries to update the data.
Update/Insert	This option is selected when the database target first tries to update the database table. If update operation fails, database target tries to insert the data.

If the specified database operation fails on any source data, error records are created. While creating a process using Database Target, you can specify what to do with error records.

- Error records can be saved into repository file.
- Error records can be ignored.
- It can be further processed (e.g. can be sent to file target).
- Process Flow can be aborted, if there is any error record.

In process designer, right click on the database target activity and select view properties. Select the value of the “Error Record” properties.

To learn, how to use Process Designer, refer to the section Using Process Designer.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

	When an excel file is used as a database target, only *Insert* operation is supported. The *Update, Insert/Update* and *Update/Insert* operations are not supported. When an excel file is used as a database target, the error records are not generated in case of data type mismatch.

8. Click **Save** button. This displays a screen confirming that the database target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the database target (refer to Figure 3.6).
9. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING FILE TARGET ACTIVITY

The File Target activity provides the ability to specify the name and path of the target file to be created.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create a File Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **File**. The Manage File Target screen is displayed (see Figure 8.5).

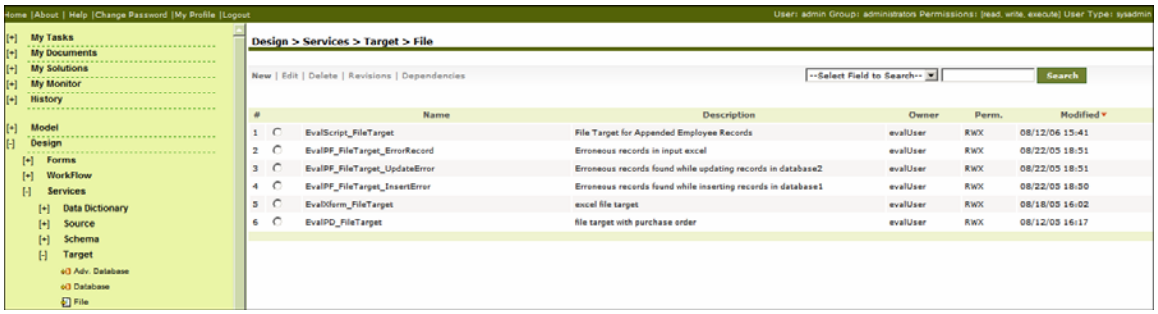


Figure 8.5: Manage File Target

4. Click the **New** link. The Create File Target screen is displayed (see Figure 8.6).

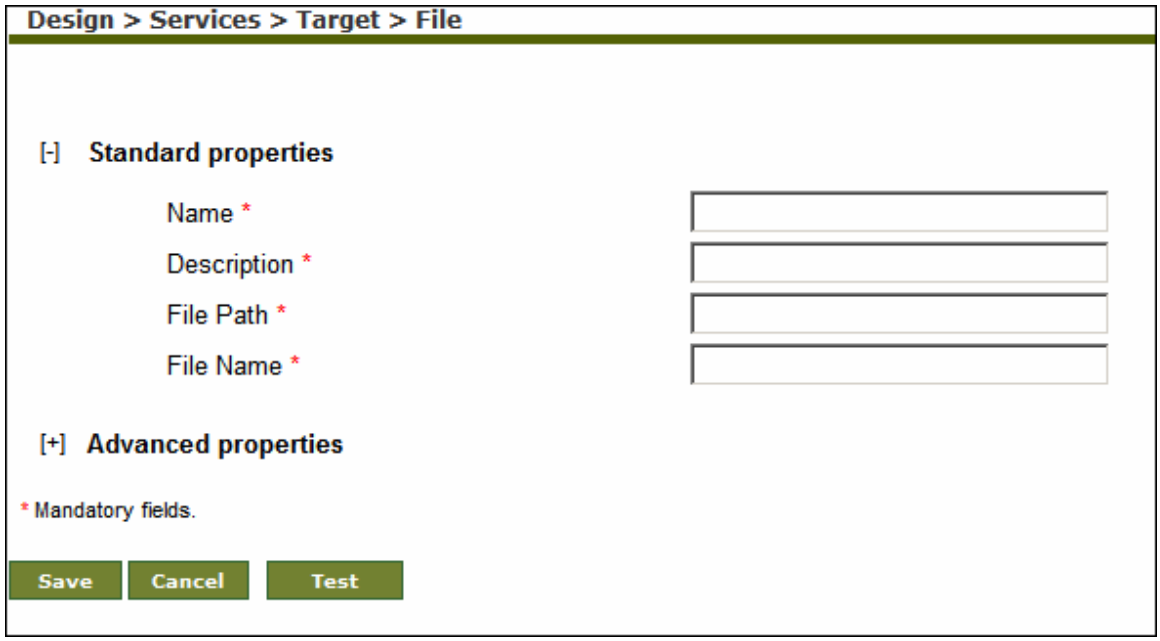


Figure 8.6: Create File Target

5. Enter the name and description of the new File Target in the *Name* and *Description* fields respectively.
6. To specify the target location, enter the full path of the target file in the *File Path* field.
7. Enter the name of the target file in the *File Name* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

8. Click **Save** button. This displays a screen confirming that the file target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen is displayed where you need to enter comments related to creating file target (refer to Figure 3.6).
9. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the file target activity at design time. For this, click **Test**. This verifies the values given in the *File Path* field. When using a file target activity in a process flow, the *append* property is set to *False* by default. This implies that an existing target activity is overwritten by the new target activity. If you want to append the new target activity to the existing target activity, you need to set this property to *True*. However, this feature is supported for Excel and Text files only. In Excel, this append feature is supported for only single sheet. Append feature for multiple sheet is not supported.

CREATING FTP TARGET ACTIVITY

FTP Target provides the ability to specify a target location that is accessible via FTP.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a FTP Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click [**+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **FTP**. The Manage FTP Target screen is displayed (see Figure 8.7).

Figure 8.7: Manage FTP Target

4. Click the **New** link. The Create FTP Target screen is displayed (see Figure 8.8).

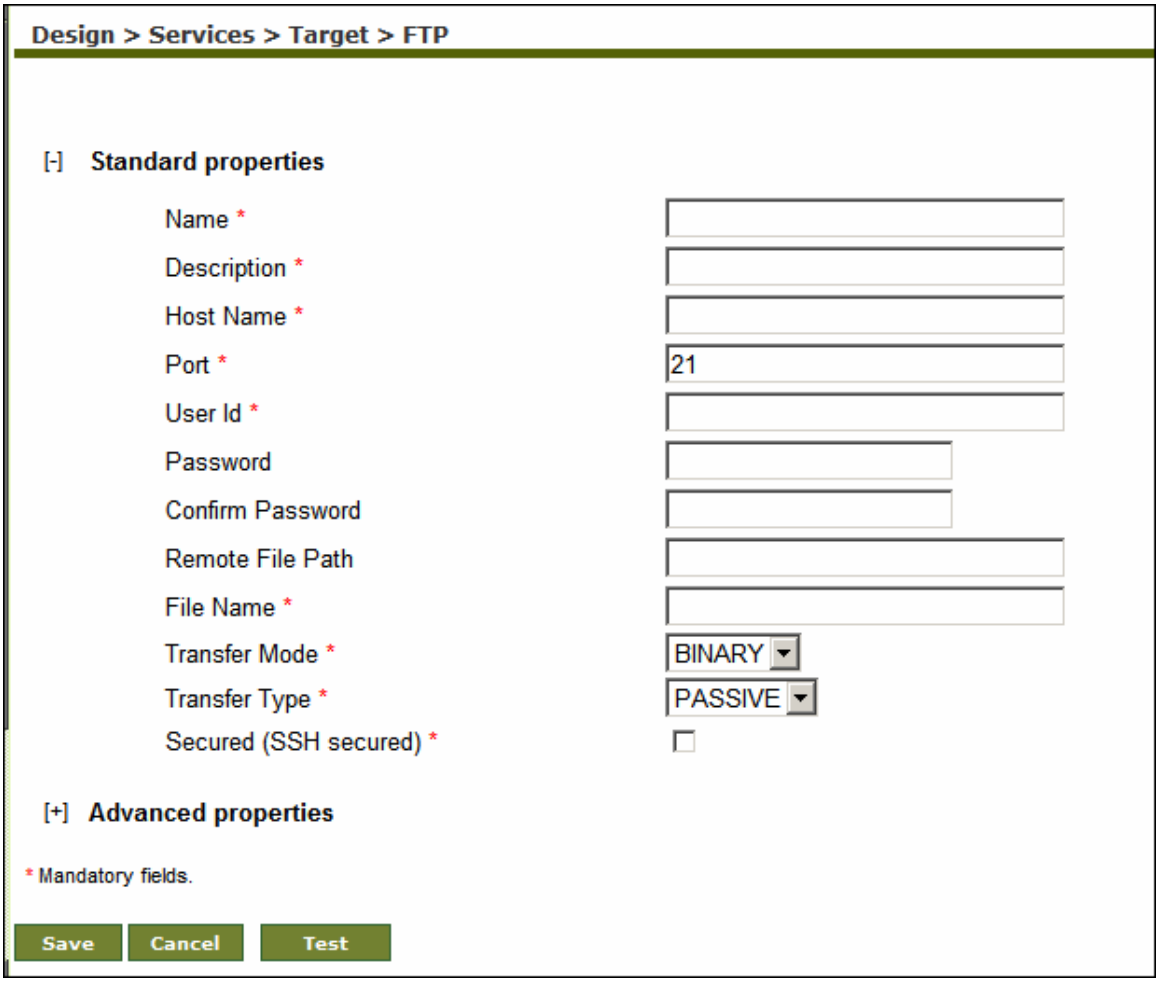


Figure 8.8: Create FTP Target

5. Enter the name and description of the new FTP Target in the *Name* and *Description* fields respectively.

6. Enter the host name/IP address and port number of the FTP Server in the *Host Name* and *Port* field respectively.

7. Enter the username and password of FTP Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.
8. Enter the full path of the target file in the *Remote File Path* field.
9. Enter the name of the target file in the *File Name* field.
10. Select the transfer mode as either BINARY or ASCII from *Transfer Mode* drop-down list. In BINARY mode a file is copied bit for bit from one machine to the other. Both files (the original and the transferred file) will contain exactly the same sequence of bytes. In ASCII mode a file may be changed slightly to maintain the meaning of EOL (End Of Line) characters.
11. Select the transfer type as either Active or Passive from the *Transfer Type* drop-down list. Active transfer is more secure since the client only initiates communication to the Server on one port whereas in case of Passive transfer the client initiates communication to the Server over two ports. Passive mode is useful when you are behind a firewall or a proxy.
12. Check the *SSH (Secure Shell)* checkbox if the FTP target accesses a secure FTP Server. When SSH is used to protect FTP Server, the control connection between the FTP client and Server is encrypted.

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

13. Click **Save** button. This displays a screen confirming that the FTP target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the FTP target (refer to Figure 3.6).
14. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

15. Click **OK** to save the comments. This displays a screen confirming that the FTP target activity has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

You can verify the FTP target activity at design time. For this, click **Test**. This verifies the values given in the *Host Name, Port, User ID, Password,* and *Secured* fields.

CREATING HTTP TARGET ACTIVITY

HTTP Target provides the ability to specify file location that is accessible via HTTP.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite

√		√	

Steps to create a HTTP Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **HTTP**. The Manage HTTP Target screen is displayed (see Figure 8.9).

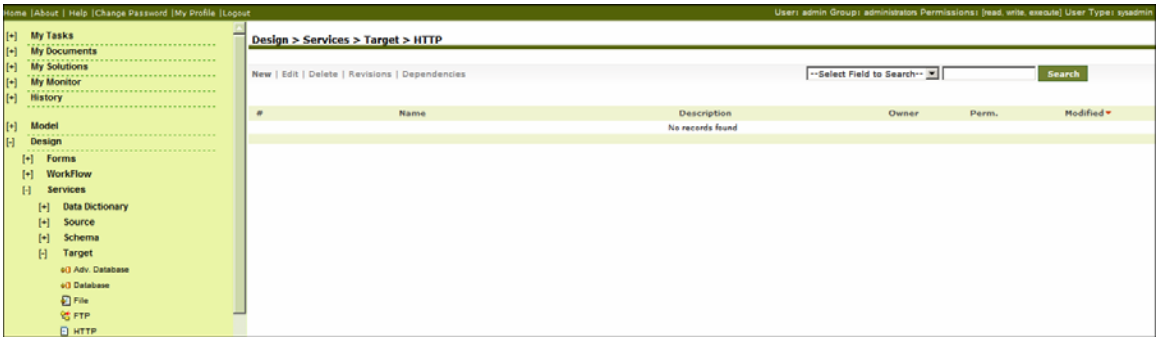


Figure 8.9: Manage HTTP Target

4. Click the **New** link. The Create HTTP Target screen is displayed (see Figure 8.10).

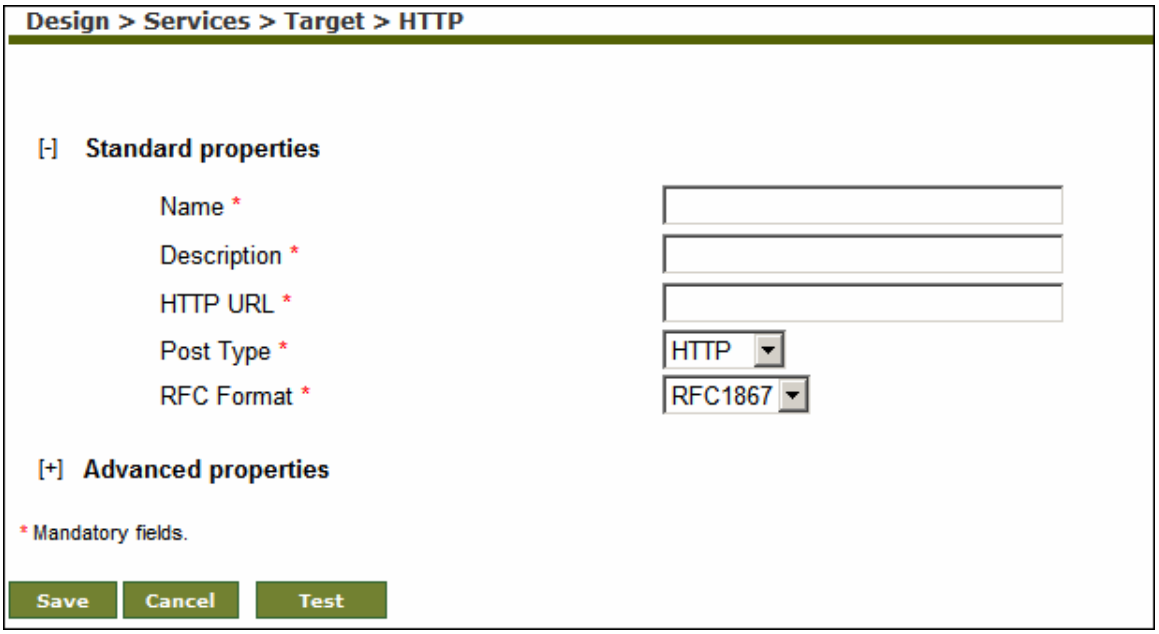


Figure 8.10: Create HTTP Target

5. Enter the name and description of the new HTTP Target in the *Name* and *Description* fields respectively.

6. Enter the URL of the target application to which you want to post the data, in the *HTTP URL* field. For example, *http://www.adeptia.com/Employee/sales.jsp*.

	In above example sales.jsp is an application which handles the posted data. This application could be any server program like ASP and servlet etc.

7. Select the post type whether HTTP or HTTPS from the *Post Type* drop-down list.
8. Select RFC format from the *RFC Format* drop-down list.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. A screen is displayed confirming that the HTTP Target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the HTTP target (refer to Figure 3.6).
10. Enter the comments in *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. A screen is displayed confirming that the HTTP Target activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the HTTP target activity at design time. For this, click **Test**. This verifies the values given in the *HTTP URL* and *Port Type* fields.

CREATING JMS TARGET ACTIVITY

The JMS Target activity provides the ability to specify a queue or topic of a JMS Server as a Target. JMS target is used to publish the target data in a topic or to post the data in the queue of a JMS Server.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Prerequisites:

- JMS Provider activity must be created before creating JMS Target activity.

Steps to create a JMS Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **JMS**. The Manage JMS Target screen is displayed (see Figure 8.11).

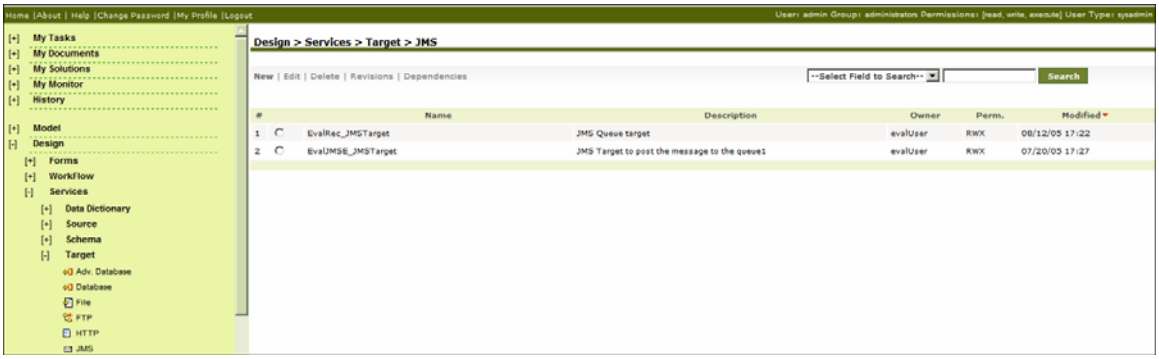


Figure 8.11: Manage JMS Target

4. Click the **New** link. The Create JMS Target screen is displayed (see Figure 8.12).

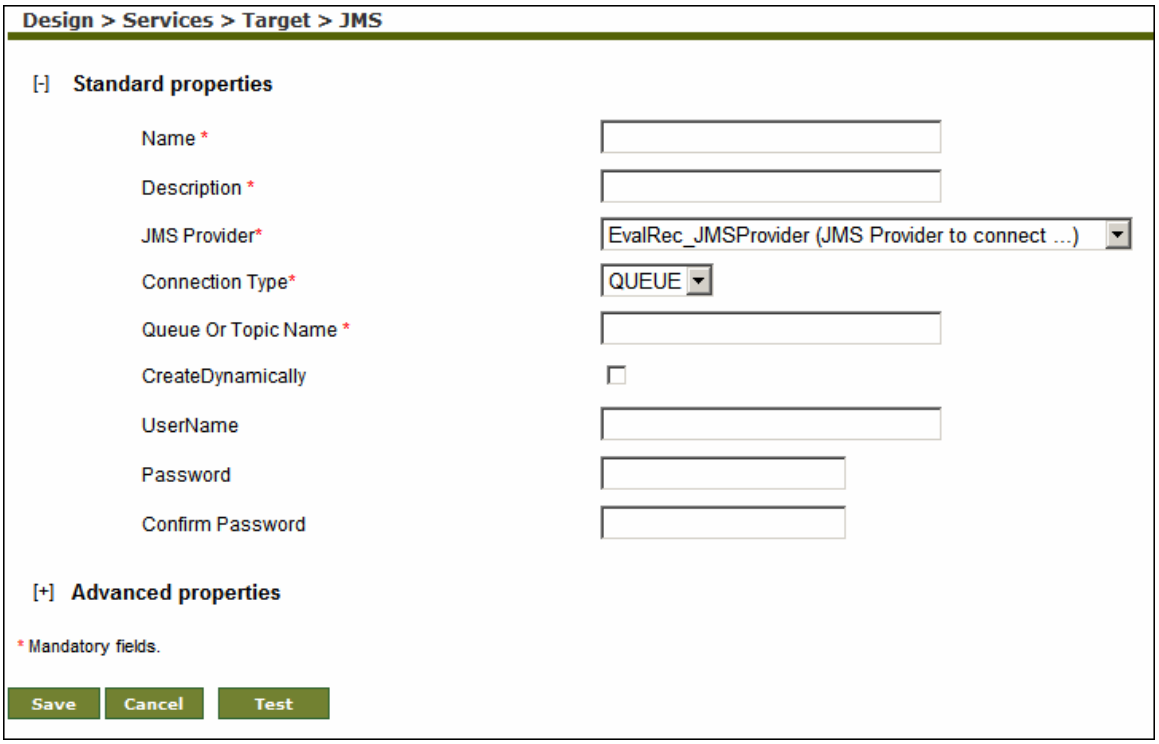


Figure 8.12: Create JMS Target

5. Enter the name and description of the new JMS Target in the *Name* and *Description* fields respectively.

6. Select the JMS Provider activity from the *JMS Provider* drop-down list.

	To learn how to create JMS Provider activity, refer to *Creating JMS Provider Activity* section in *Administrator Guide*.

7. Select the connection type as either queue or topic from the *Connection Type* drop-down list.
8. Enter the name of queue or topic as configured in the JMS Server in the *Queue Or Topic Name* field.
9. Check the *Create Dynamically* checkbox, if queue or topic specified above, doesn't already exist on the JMS Server. If you enable create dynamically check box, it will create the queue or topic specified above on the JMS Server.
10. Enter the username and password required to connect to JMS Server in the *UserName* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

11. Click **Save** button. This displays a screen confirming that the JMS target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the JMS target (refer to Figure 3.6).
12. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the JMS target activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the JMS target activity at design time. For this, click **Test**. This verifies the values given in the *JMS Provider* and *Connection Type* fields.

CREATING LAN FILE TARGET ACTIVITY

LAN File Target provides the ability to specify a target file location that is accessible on the network.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create a LAN File Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **LAN File**. The Manage LAN File Target screen is displayed (see Figure 8.13).

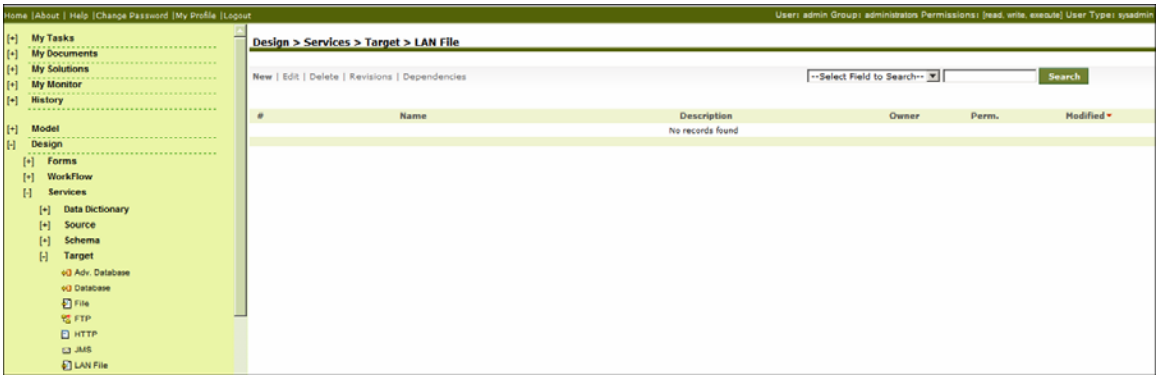


Figure 8.13: Manage LAN File Target

4. Click the **New** link. The Create LAN File Target screen is displayed (see Figure 8.14).

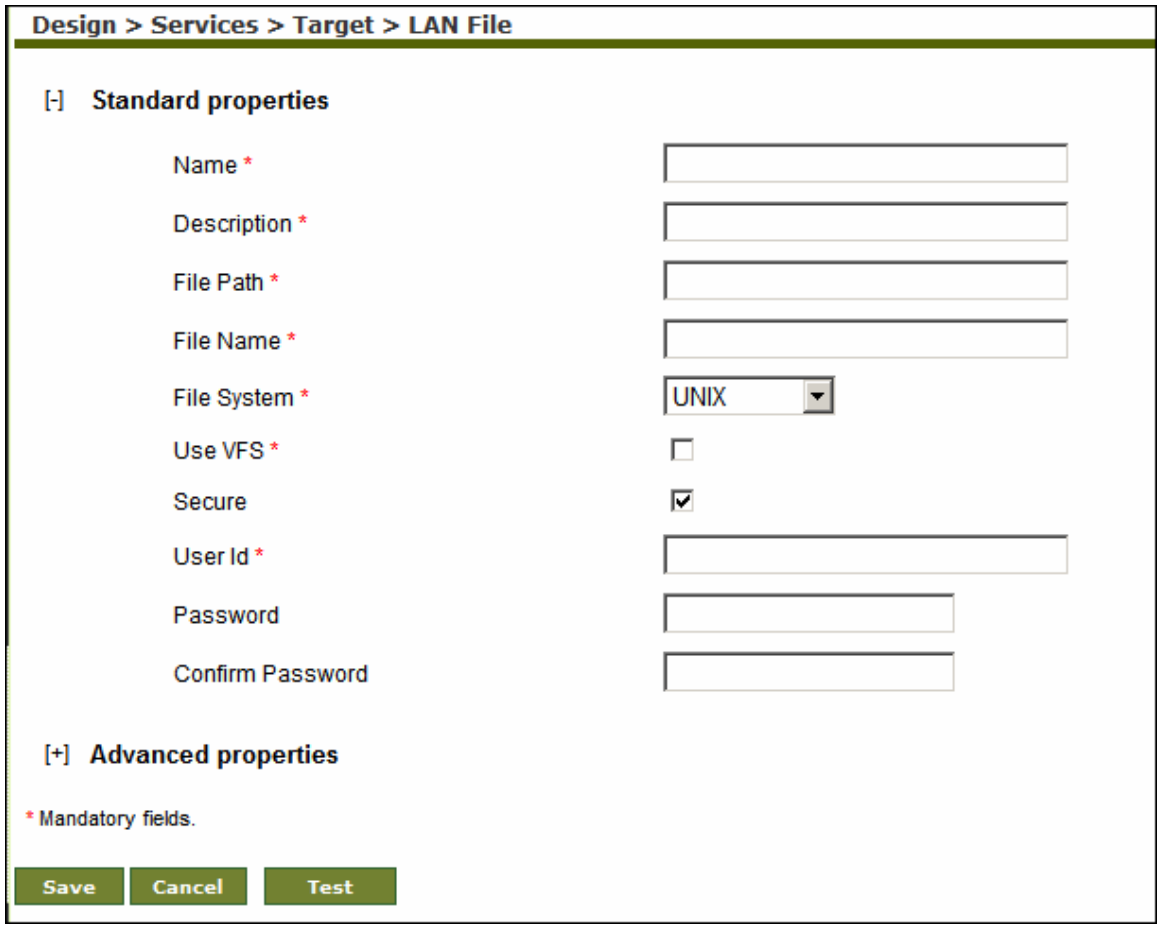


Figure 8.14: Create LAN File Target

5. Enter the name and description for the new LAN File Target in the *Name* and *Description* fields respectively.
6. Enter the network path of the folder where you want the target file to be created in the *File Path* field in the following format:

   \\hostname\folder name

7. Enter the name of target file in the *File Name* field.
8. Select the target file system, whether Windows or Unix from the *File System* drop-down list.
9. When Adeptia Suite is installed on Windows Operating System, LAN File Target uses windows service to connect to remote machine. It just connects once and uses the same connection with the same User ID and Password (which is stored in the cache) every time. If you want to enforce the validation of User ID and Password every time while accessing the remote machine, check the *Use VFS* checkbox.
10. If username and password is required to access the target location, check the *Secure* checkbox.
11. Enter the username and password in the *User ID* and *Password* fields respectively. Then re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

12. Click **Save** button. This displays a screen confirming that the LAN file target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the LAN file target (refer to Figure 3.6).
13. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the LAN file target activity at design time. For this, click **Test**. This verifies the values given in the *File Path, User ID* and *Password* fields. When using a LAN file target activity in a process flow, the *append* property is set to *False* by default. This implies that an existing target activity is overwritten by the new target activity. If you want to append the new target activity to the existing target activity, you need to set this property to *True*. However, this feature is supported for Excel and Text files only.

CREATING MAIL TARGET ACTIVITY

Mail Target provides the ability to specify target location that is accessible via Mail.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a Mail Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **Mail**. The Manage Mail Target screen is displayed (see Figure 8.15).

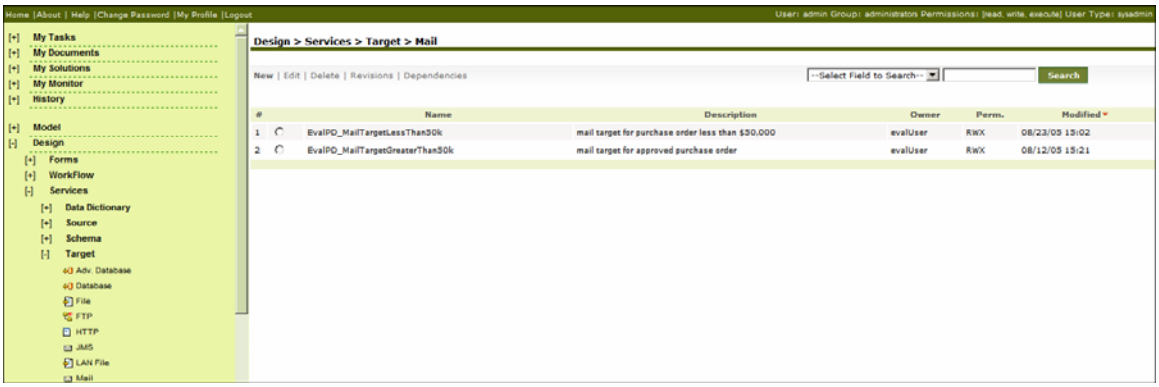


Figure 8.15: Manage Mail Target

4. Click the **New** link. The Create Mail Target screen is displayed (see Figure 8.16).

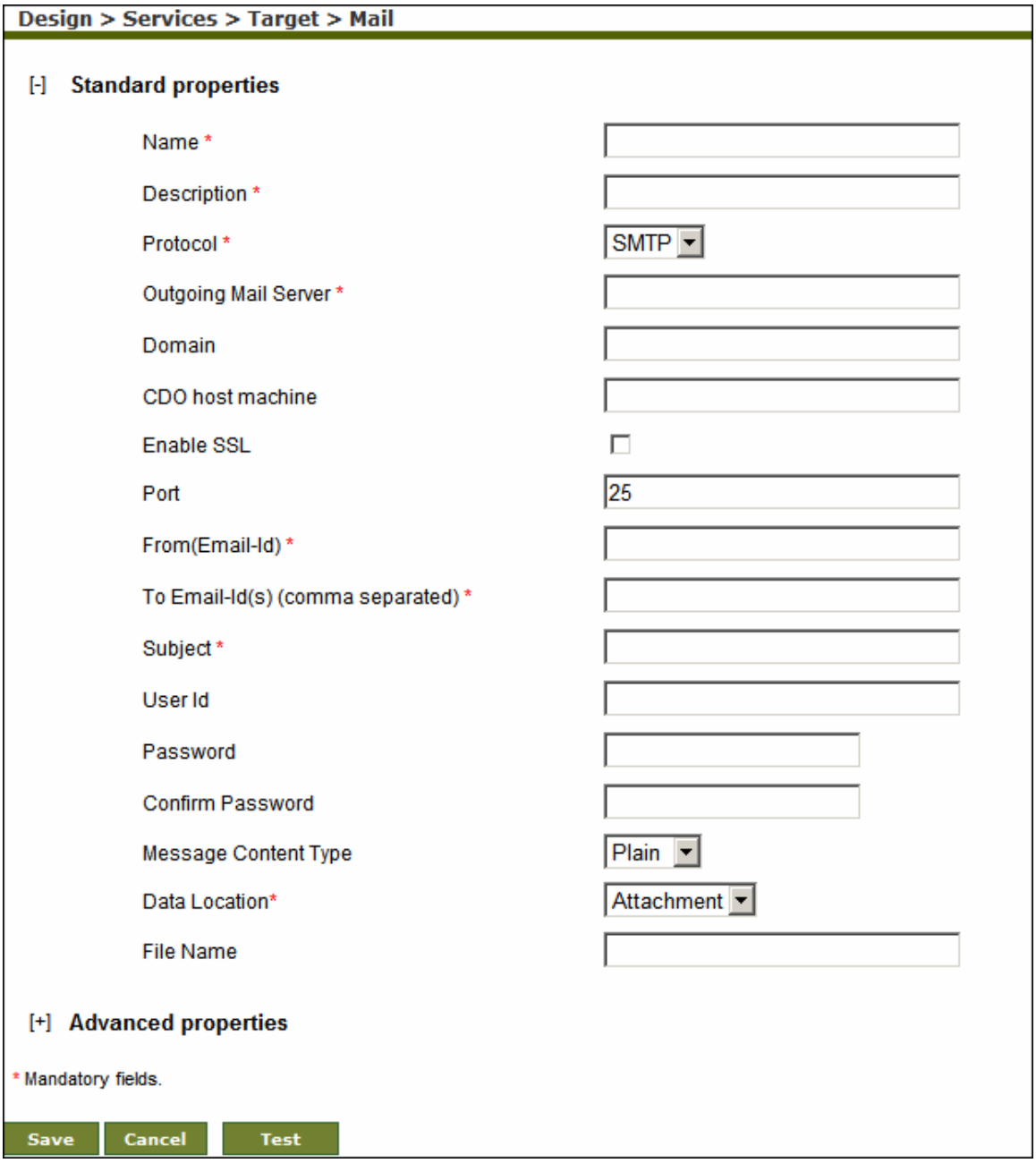


Figure 8.16: Create Mail Target

5. Enter the name and description of the new Mail Target in the *Name* and *Description* fields respectively.
6. Select the Internet standard protocol to be used for sending outgoing mails, from the *Protocol* drop-down list. You can select either the SMTP or the MAPI protocol. Based on the selected protocol, the default port number for that protocol is displayed in the *Port* field.
7. Enter the outgoing mail (SMTP) Server address in the *Outgoing Mail (SMTP) Server* field.

	To access mails from Microsoft Exchange Server use *MAPI* in *Protocol* drop-down list. To connect Adeptia Suite with Microsoft Exchange Server, you need to buy a third party tool called J-Integra for Exchange. J-Integra for Exchange is a high performance middleware bridge that enables Java Exchange interoperability. If you want to retrieve mails from an Exchange Server using J-Integra, Select MAPI in the *Protocol* drop-down list. If *MAPI* is selected in *Protocol* drop-down list: ▪ Enter name of the exchange server in *Outgoing Mail Server* field ▪ Enter the domain name in the *Domain* field ▪ Enter the name of the CDO host machine in *CDO host machine* field. CDOConfig.exe is a tool that comes with the J-Integra for Exchange SDK and is used for configuring CDO. Host where CDO is configured is called CDO host machine. For detailed information about Jintegra for Exchange, refer to **http://j-integra.intrinsyc.com/products/exchange/**.

8. Check the *Enable SSL* checkbox, if the specified outgoing mail server requires a secure connection.
9. Enter the port of Outgoing Mail Server in the *Port* field.
10. Enter the sender's Email-Id in the *From (Email-Id)* field.
11. Enter the Email-Id(s) of the email recipients separated by commas in the *To Email-Id(s) (comma separated)* field.
12. Enter the subject of Target email in the *Subject* field.
13. Enter the username and password of Mail Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.
14. Select the content type of the outgoing message (as Plain or HTML) from *Message Content Type* drop-down list.

	This field is enabled only when SMTP is selected in the *Protocol* field.

15. Select the location of data in the mail whether it is to be sent as an attachment or in email body from the *Data Location* drop-down list.
16. If the data is to be sent as attachment, enter the name of the file in the *File Name* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

17. Click **Save** button. This displays a screen confirming that the mail target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the mail target (refer to Figure 3.6).
18. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the mail target activity at design time. For this, click **Test**. This verifies the values given in the *Outgoing Mail (SMTP) Server, Port, User ID* and *Password* fields.

CREATING WEBDAV TARGET ACTIVITY

The WebDAV Target activity provides the ability to specify a WebDAV Server as a target.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create WebDAV Target Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Target** to expand the tree, and then click **WebDAV**. The Manage WebDAV Target screen is displayed (see Figure 8.17).

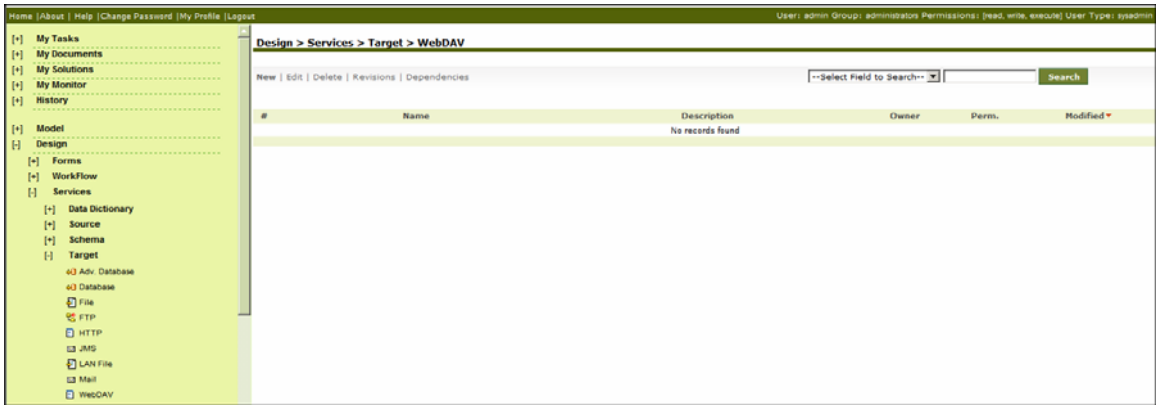


Figure 8.17: Manage WebDAV Target

4. Click the **New** link. The Create WebDAV Target screen is displayed (see Figure 8.18).

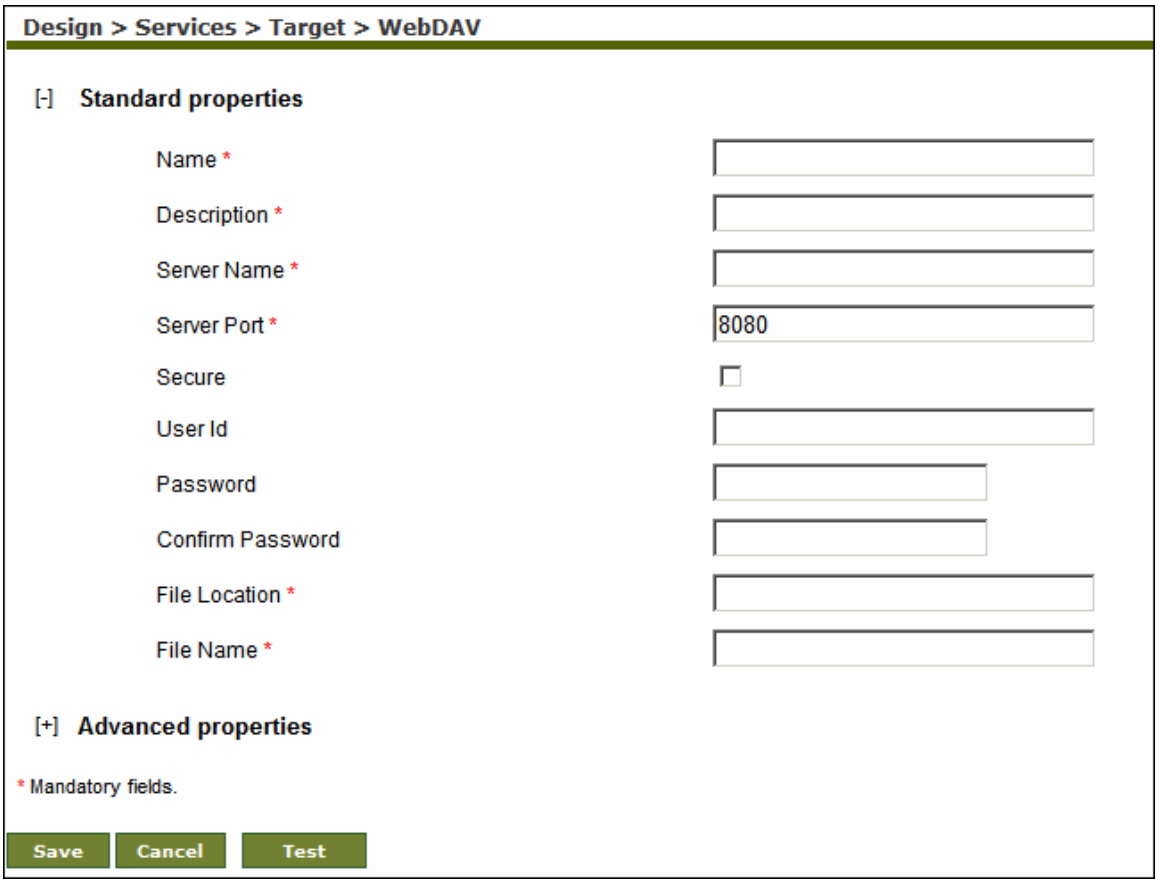


Figure 8.18: Create WebDAV Target Activity

5. Enter the name and description of the new WebDAV target in the *Name* and *Description* fields respectively.
6. Enter the name of WebDAV Server and port on which WebDAV Server is running in the *Server Name* and *Server Port* fields respectively.
7. If the WebDAV is secured i.e. username and password is required to access it, then check the *Secure* checkbox and enter the username and password of the WebDAV Server in the *User ID* and *Password* fields respectively.

	If you are using WebDAV Server, which is built in with Adeptia Suite, the default Username is “Administrator” and the password is “indigo”.

8. Re-enter the password in the *Confirm Password* field.
9. Enter the path of the target file in the *File location* field.
10. Enter the name of the target file in the *File Name* field.

	Directory specified in the File location field, must be available in the WebDAV repository. To learn about Advanced Properties refer to **Changing Advanced**

	Properties section.

11. Click **Save** button. This displays a screen confirming that the WebDAV target activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the WebDAV target (refer to Figure 3.6).
12. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the WebDAV target activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can verify the WebDAV target activity at design time. For this, click **Test**. This verifies the values in the *Server Name* and *Server Port* fields and checks whether the file actually exists in the specified location.

9 CREATING WEB SERVICES

The Web Services module has two components:

- **Consumer**
- **Provider**

The *Consumer* component allows users to access any Web Services on the Web unless the service to be accessed has restricted access. Using the *Provider* component users can publish their Web Services for access to others. Access to Web Services to be published can also be restricted to selected users

This section also describes the creation process of:

- **Security Policy activity for Web Services**
- **Consumer Web Services using UDDI**
- **Consumer Web Services using URI**
- **Provider or Publisher Web Services**

CREATING SECURITY POLICY ACTIVITY FOR WEB SERVICES

Since Web Services expose crucial business information, Web services security is critically important. A Web service can be secured using Security Policy activity. It is recommended that users create an appropriate security policy before they publish Web services using the Web service provider.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	

Steps to create a Security Policy Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Web Services** to expand the tree, and then click **Security Policy**. The Manage Security Policy screen is displayed (see Figure 9.1).

Figure 9.1: Manage Security Policy

4. Click the **New** link. The Create Security Policy screen is displayed (see Figure 9.2).

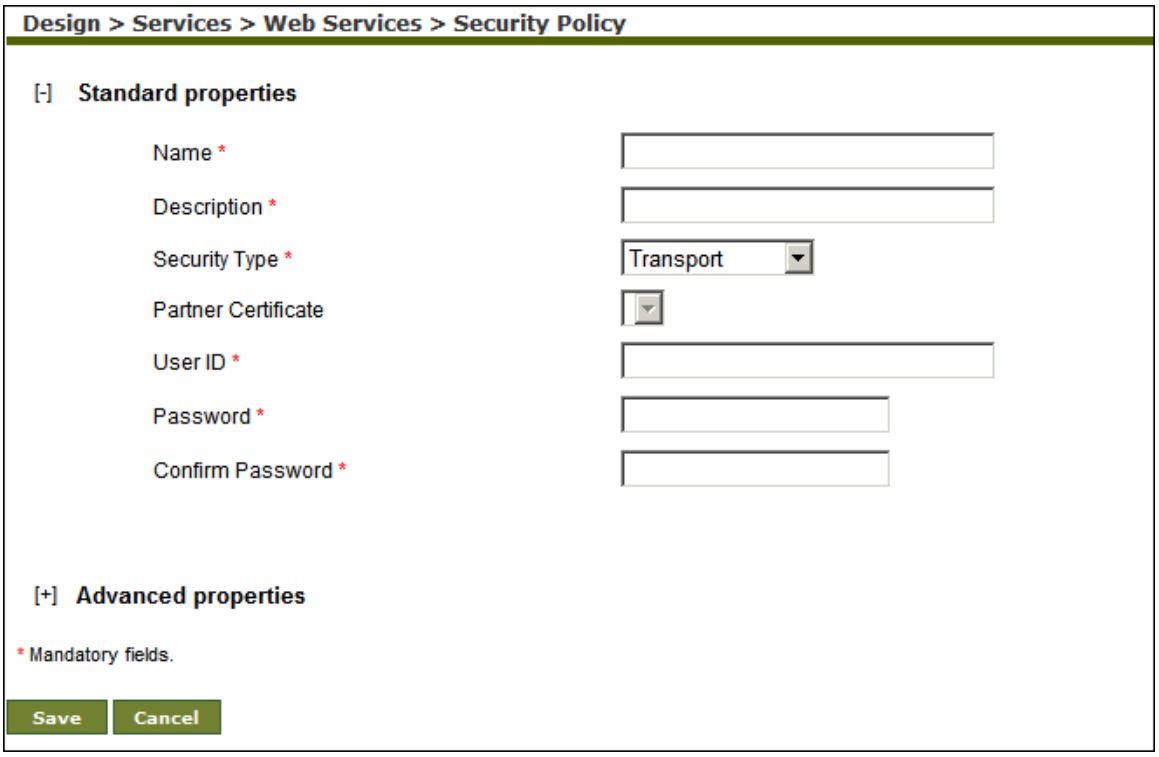


Figure 9.2: Create Security Policy

5. Enter the name and description of the new Security Policy in the *Name* and *Description* fields respectively.

6. Select the type of security as either Transport or Message Level depending upon the security level of Web Service, from the *Security* Type drop-down list. The types of Security are listed in the table below.

Table 9.1: Security Types

Security Type	Description
Transport	If user selects Transport type then SSL and basic authentication security is provided. System asks for the

	User ID and Password.
Message Level	If user selects Message Level type then security is provided as per WS-security standard. System asks for the Partner certificate name, User ID and Password.

7. Select required Partner Certificate from the *Partner Certificate* drop-down list.

	Partner certificate is an electronic signature, which is used to validate the consumer who is accessing the Web Services.

8. Enter the User ID and password in the *User ID* and *Password* fields respectively. Then re-enter password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. This displays a screen confirming that the security policy has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the security policy (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the security policy has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING WEB SERVICE CONSUMER ACTIVITY

A Web Service Consumer locates a Web service and invokes the operations it provides. A Web Service consumer activity is created to access any Web Service.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	

A Web Service can be located in two ways, explained in the table below.

Table 9.2: Web Services

Method	Description
UDDI	UDDI (Universal Description, Discovery, and Integration) provides a mechanism for clients to dynamically find other Web Services. Using the UDDI interface, businesses can dynamically connect to services provided by external business partners. UDDI is a public registry, where one can publish and inquire about Web Services. A UDDI registry has two kinds of clients: businesses that want to publish a service (and its usage interfaces), and clients who want to obtain services of a certain kind and bind programmatically to them.
URI	URI (Uniform Resource Identifiers) provides a way for clients to define the location of Web Services by giving an HTML URL or Local LAN path for a WSDL file.

Creating Web Services Consumer Activity using UDDI

Steps to create a Web Service Consumer Activity using UDDI

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Web Services** to expand the tree, and then click **Consumer**. The Manage Consumer screen is displayed (see Figure 9.3).

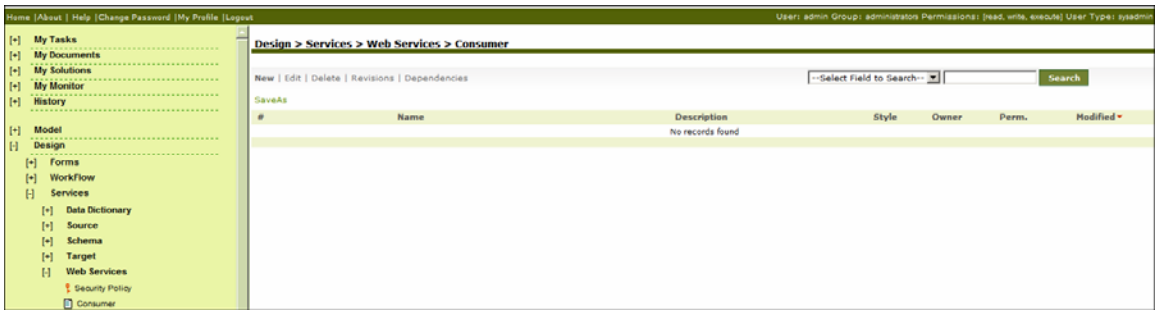


Figure 9.3: Manage Web Service Consumer Activity

4. Click the **New** link. The Create Web Service Consumer screen is displayed (see Figure 9.4).

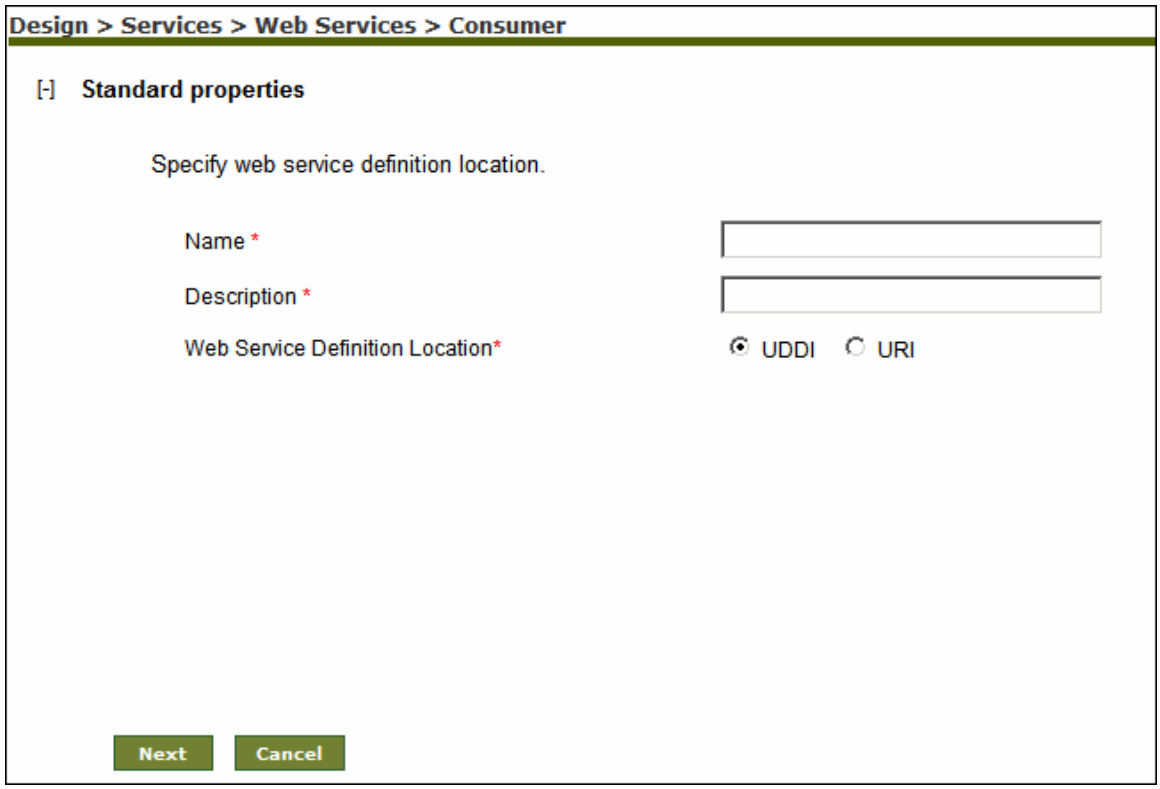


Figure 9.4: Create Web Service Consumer Activity

5. Enter the name and description of the new Web Service Consumer activity in the *Name* and *Description* fields respectively.

6. Select UDDI in *Web Service Definition Location*.

	WSDL (Web Service Definition Language) is a document written in XML. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes.

7. Click **Next** button. The Web Service Consumer screen is displayed (see Figure 9.5).

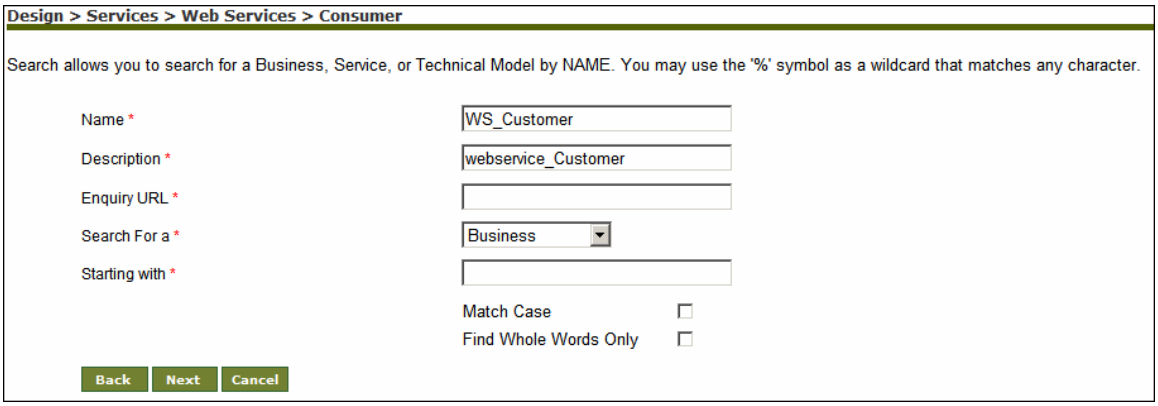


Figure 9.5: Search a Web Service

8. Enter the URL of UDDI search site in *Enquiry URL* field.

   For example, **http://uddi.microsoft.com/inquire**

9. Select the required category from the *Search For a* drop-down list.
10. Enter the relevant letter or word in the *Starting with* field.
11. You can check the *Match Case* and the *Find Whole Words* only checkbox to filter the search result.
12. Click **Next** button. A screen is displayed with the search result (see Figure 9.6).

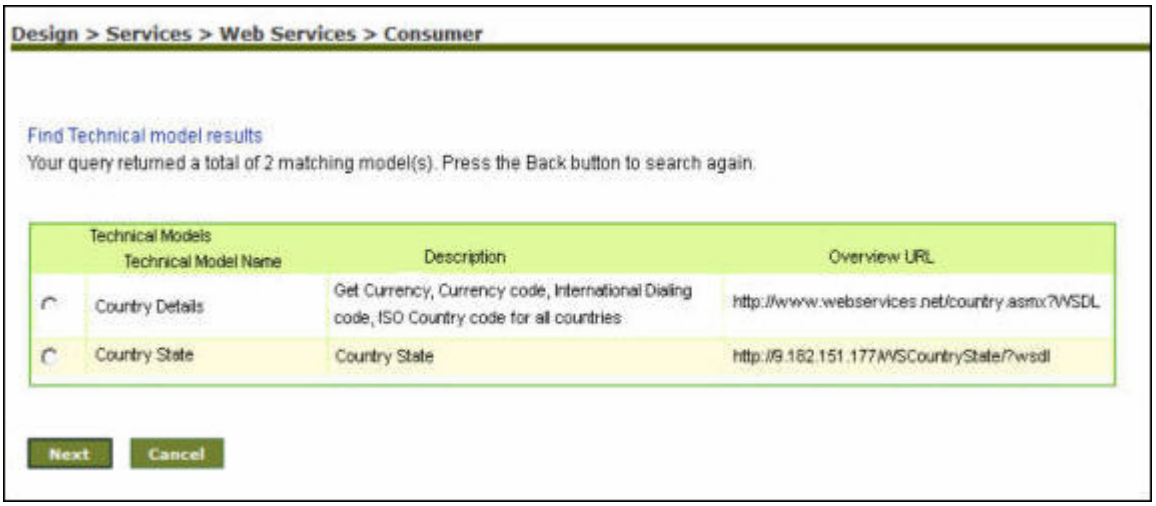


Figure 9.6: Web Service search result

13. Select the required search result and click **Next** button. The Web Service screen is displayed with list of operations (see Figure 9.7).

Figure 9.7: Select Web Service Operation

14. Select the name of Service from the *Service Name* drop-down list.
15. Select the port type from the *Port Type* drop-down list. A port type can support multiple ports. This selection populates the options in the *Ports* drop-down list.
16. Select the port for the selected port type from the *Ports* drop-down list.

	Currently, the Soap12 port is not supported by Adeptia.

17. Select the required operation from the *Operation(s)* field. This selection automatically displays the style name in the *Style Name* field.

	The *Style Name* field is automatically populated according to the specified operation. There are two types of Style: **document** and **rpc**. **Document**: When a WS Consumer invokes a *document* style Web service, the consumer typically sends it an entire document, such as a purchase order, rather than a discrete set of parameters. The Web service accepts the entire document, processes it, and may or may not return a result document. In a *document* style, the input can be read from context or a stream coming from another activity. Similarly, output can be set to context or can be generated as stream to other activities. In case the output is set to context, a variable is created in the context with name as specified in the

	Output Parameter Name field and then output is set into that variable. **rpc**: In *rpc* style when WS Consumer invokes the Web service, the consumer sends parameter values to the Web service, which executes the required methods, and then sends back the return values. In *rpc* style, input can be read from the context only and the output parameter is set in the context only. This style does not generate a stream. A variable is created in the context with name as specified in the Output Parameter Name field and then output is set into that variable. In the current example, the specified Web Service is of *document* style.

18. Click **Next** button. The Web Service Consumer Screen is displayed (see Figure 9.8).

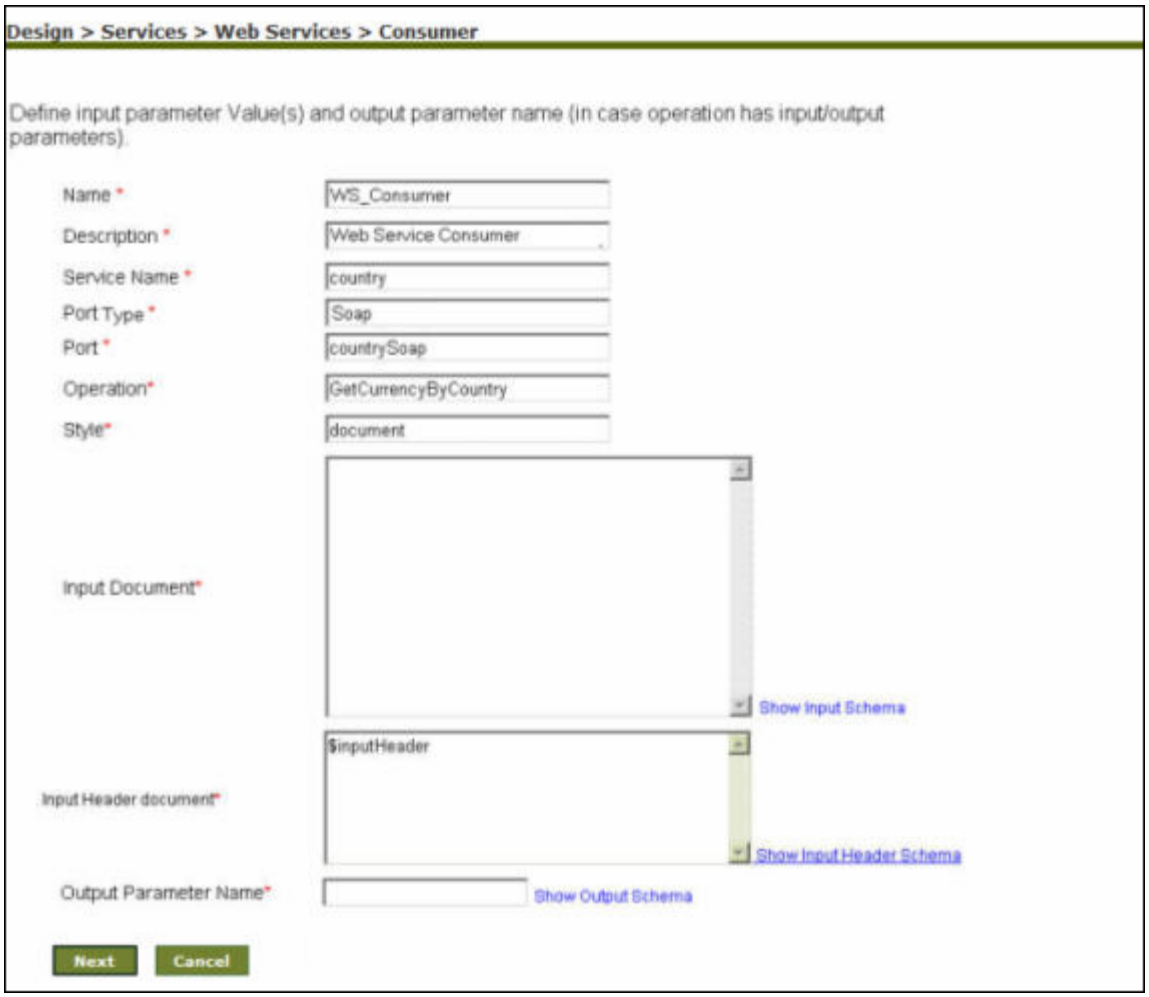


Figure 9.8: Enter Input XML

19. Enter the Input XML in the *Input Document* field. Input document is used to define the input data for WS Consumer activity. The actual value could be given in three ways, explained in the table below.

Table 9.3: Value for WS Consumer Activity

Method	Description
Definition Time	Here the complete XML document, which is compliant with the Input XSD, has to be copied in the *Input Document* field. To view sample XSD, click **Show Input Schema** link. The sample input schema is displayed. Do not paste this schema in the *Input Document* field.
Run Time	Here a variable prefixed with $ is defined in the *Input Document* field. For example $variable1. The value of variable1 will be searched in process flow context during process flow execution as the control reaches web service consumer activity. The value must be an XML string.
Streaming	WS Consumer activity can also consume stream as input data from another activity. This behavior is controlled at the time of the process flow creation not at the time of WS Consumer activity creation. At creation time user must give either an XML document or variable as defined above.

20. At execution time WS consumer activity will search for its input in following order:

1. Take input from Stream
2. Search variable in process flow context
3. Get XML document

The way input is passed to the WS Consumer activity depends upon the style *document* or *rpc* not on WSDL locator UDDI or URI.

21. Enter Header XML in the "*Input Header Document*" field to define this at design time. You can also enter a variable with $ prefix if you want read this value dynamically from context. This value cannot be read from stream.

- The WS Header element supports multiple headers.
- Header is optional it may be there in some operations or may not be there in other.
- If there is any fault as an output of web service call, it sets as "Output Parameter Name" + "Fault" in the process context.
- Web service response may also have header. The header is set into context with variable "output Parameter Name" +"Header".

22. Enter relevant context variable in the *Output Parameter Name* field. Web Service stores the output in this context variable.

23. Click **Next** button. The Select Security Policy screen is displayed (see Figure 9.9).

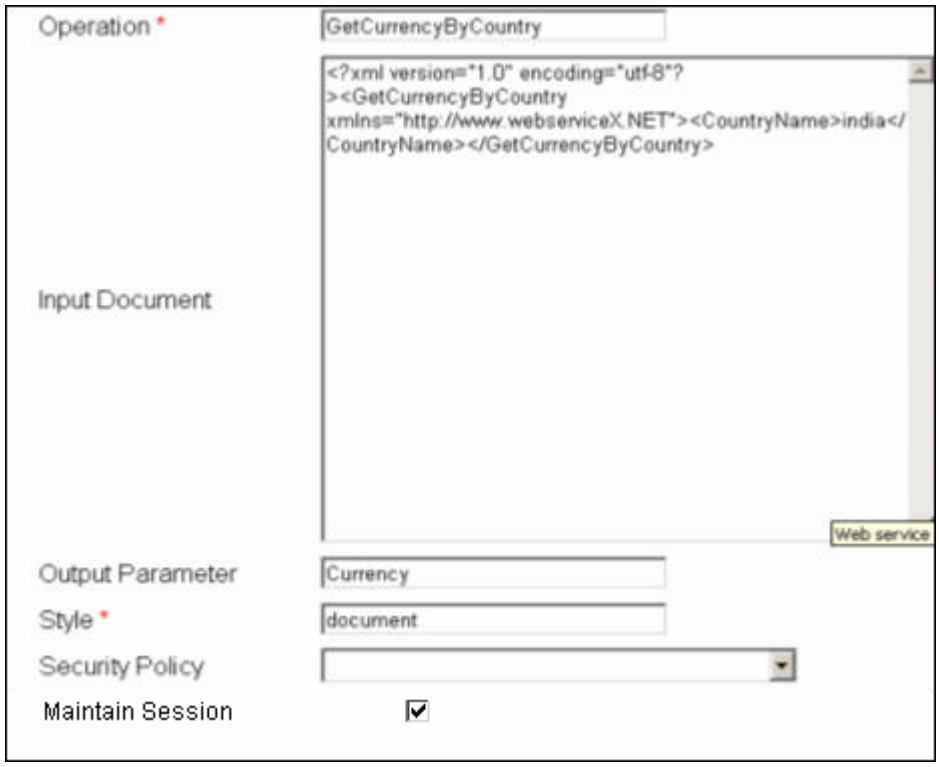


Figure 9.9: Select Security Policy

24. Select the Security Policy activity from the *Security Policy* drop-down list.

25. Enable *Maintain Session* checkbox if the web service, which you are accessing, maintains session through cookie. This field is applicable for only *Document* style web services.

	To change timeout duration of the web service, click **[+]** to expand Advanced Properties and enter the time in *Timeout(in seconds)*field. The *Timeout* duration controls how long the connectivity should be there with the web service in case it takes longer time to execute a web service. The *Timeout* duration should be higher than the actual execution time. This field is applicable for only *Document* style web services. To learn about Advanced Properties refer to **Changing Advanced Properties** section

26. Click **Save** button. This displays a screen confirming that the web service consumer has been created successfully. If the Comments property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the web service consumer (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

28. Click **OK** to save the comments. This displays a screen confirming that the web service consumer has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Creating Web Services Consumer Activity using URI

Steps to create a Web Service Consumer Activity using URI

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Web Services** to expand the tree, and then click **Consumer**. The Manage Consumer screen is displayed (refer to Figure 9.3).
4. Click the **New** link. The Create Web Service Consumer screen is displayed (refer to Figure 9.4).
5. Enter the name of the new Web Service Consumer activity in the *Name* field. Then, enter the description for the Web Service Consumer activity in the *Description* field.
6. Select URI in *Web Service Definition Location*. The Web Service Consumer screen is displayed (see Figure 9.10).

	WSDL (Web Service Definition Language) is a document written in XML. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes.

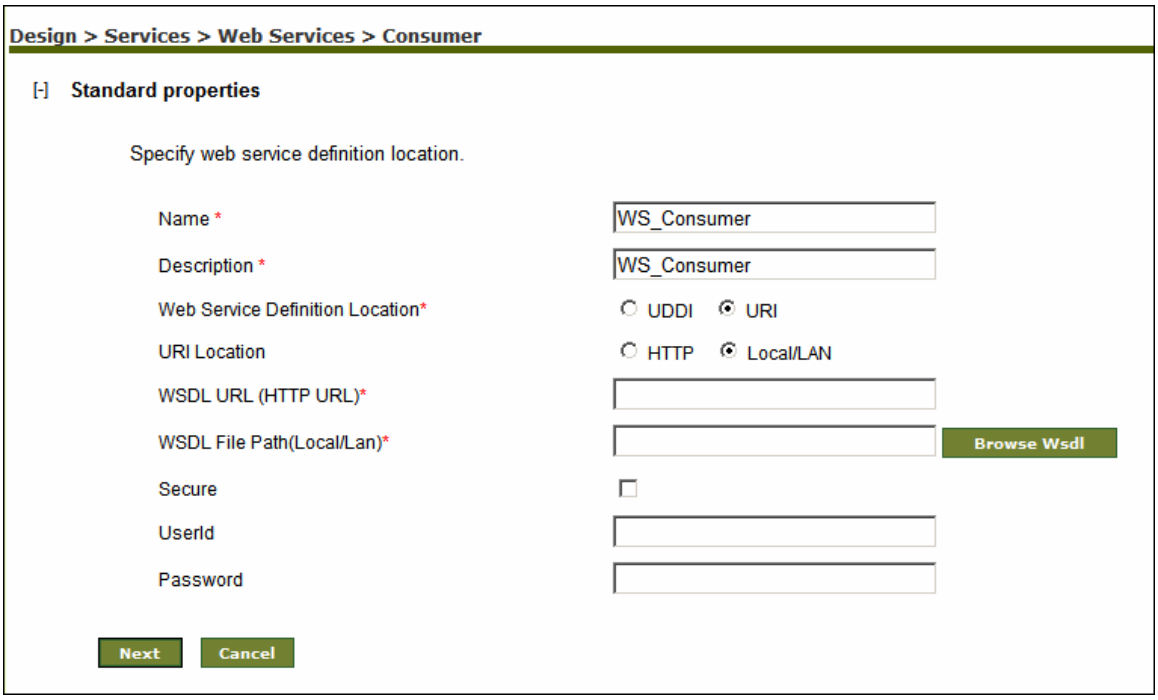


Figure 9.10: Locate WSDL

7. Select the location of URI.

	In the URI Location, select: ▪ HTTP, if the WSDL file is located on an HTTP Site. ▪ LocalLAN, if the WSDL file is located Local LAN.

8. Click **Browse Wsdl** button. This displays the Upload Wsdl file screen (see Figure 9.11).

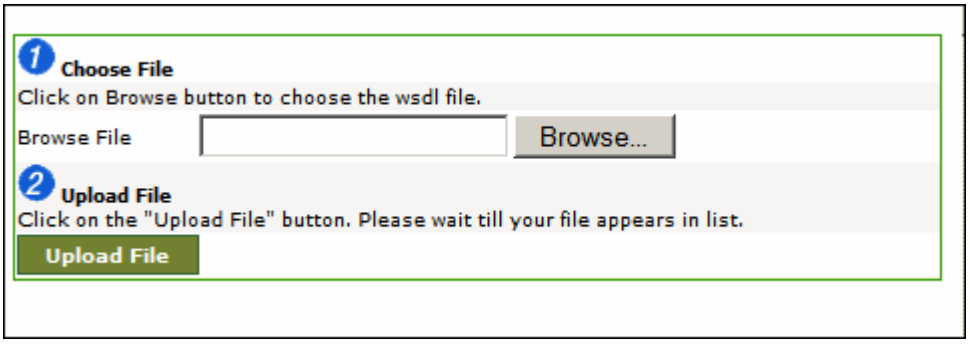


Figure 9.11: Upload WSDL File

9. Click Browse to select the wsdl file. Then click Upload File button. This uploads the file and displays it in the *WSDL File Path* field in the Web Consumer screen (see Figure 9.12).

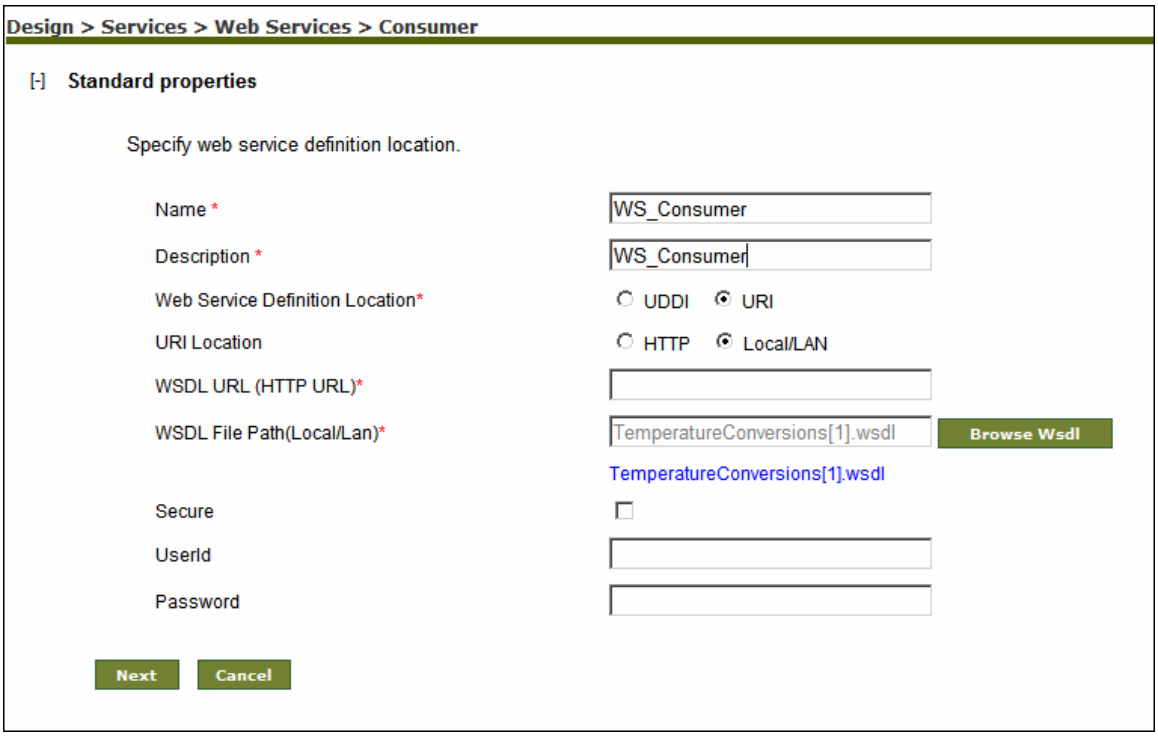


Figure 9.12: Uploaded WSDL File

10. Check the *Secure* checkbox if the path specified in *WSDL URL* field is secured. Then enter User ID in the *User ID* field and password in the *Password* field.
11. Click **Next** button. The Web Consumer screen is displayed (see Figure 9.13).

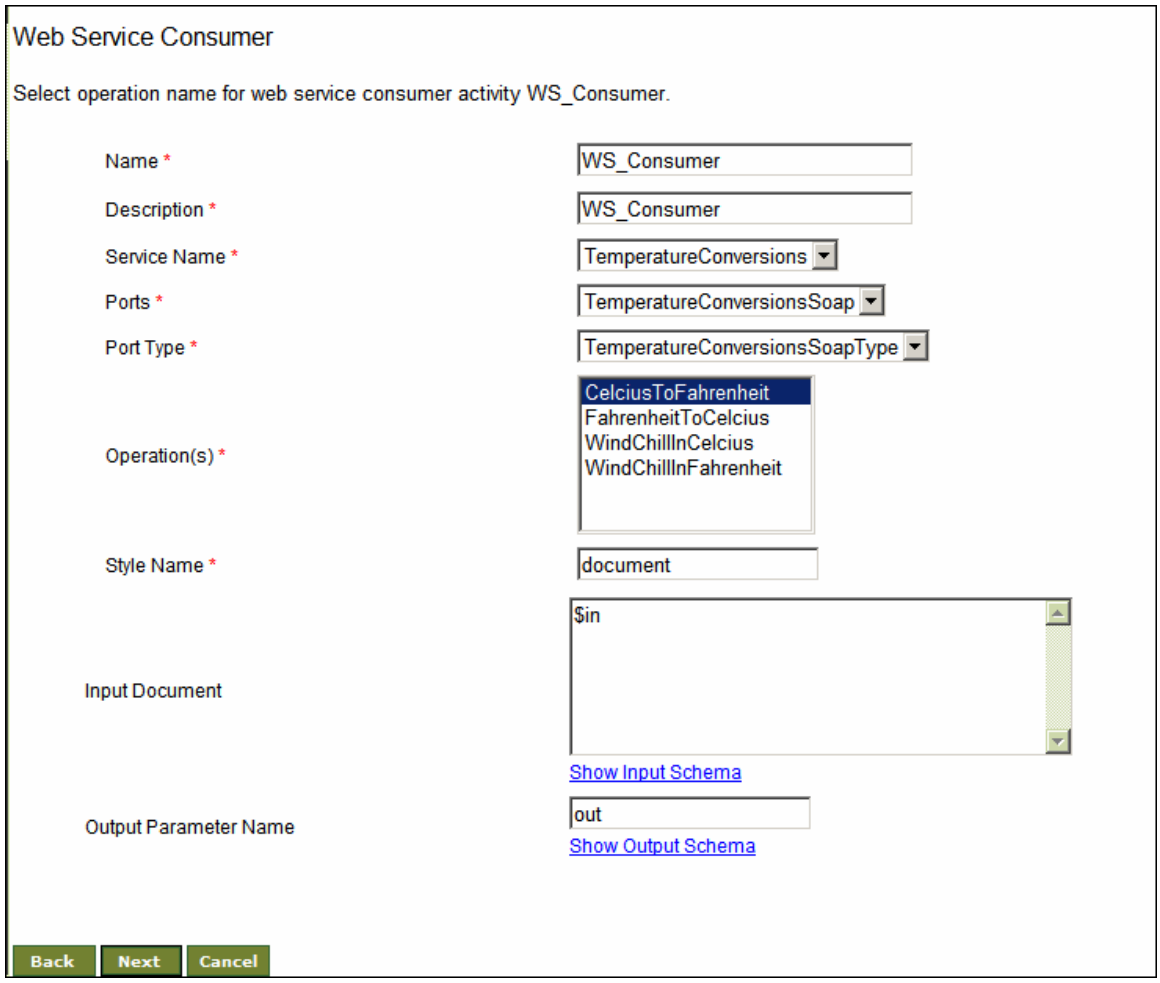


Figure 9.13: Select Web Service Operation

12. Select the name of Service from the *Service Name* drop-down list.
13. Select the port type from the *Port Type* drop-down list. A port type can support multiple ports. This selection populates the options in the *Ports* drop-down list.
14. Select the port for the selected port type from the *Ports* drop-down list.

	Currently, Soap12 port is not supported by Adeptia Suite.

15. Select the required operation from the *Operation(s)* field. This selection automatically displays the style name in the *Style Name* field.

	The *Style Name* field is automatically populated according to the specified operation. There are two types of Style: **document** and **rpc**. **Document**: When a WS Consumer invokes a *document* style Web service, the consumer typically sends it an entire document, such as a purchase order, rather than a discrete set of parameters. The Web service accepts the entire document, processes it, and may or may not return a result document. In a *document* style, the input can be read from context or a

	stream coming from another activity. Similarly, output can be set to context or can be generated as stream to other activities. In case the output is set to context, a variable is created in the context with name as specified in the *Output Parameter Name* field and then output is set into that variable. Additionally, the XSD of the Web Service Consumer can be used to create XML schema. **rpc**: In *rpc* style when WS Consumer invokes the Web service, the consumer sends parameter values to the Web service, which executes the required methods, and then sends back the return values. In *rpc* style, input can be read from the context only and the output parameter is set in the context only. This style does not generate a stream. A variable is created in the context with name as specified in the Output Parameter Name field and then output is set into that variable. In the current example, the specified Web Service is of *rpc* style.

16. Select required operation in the *Operation* field.
17. Click **Next** button. The Web Service Consumer screen is displayed (see Figure 9.14).

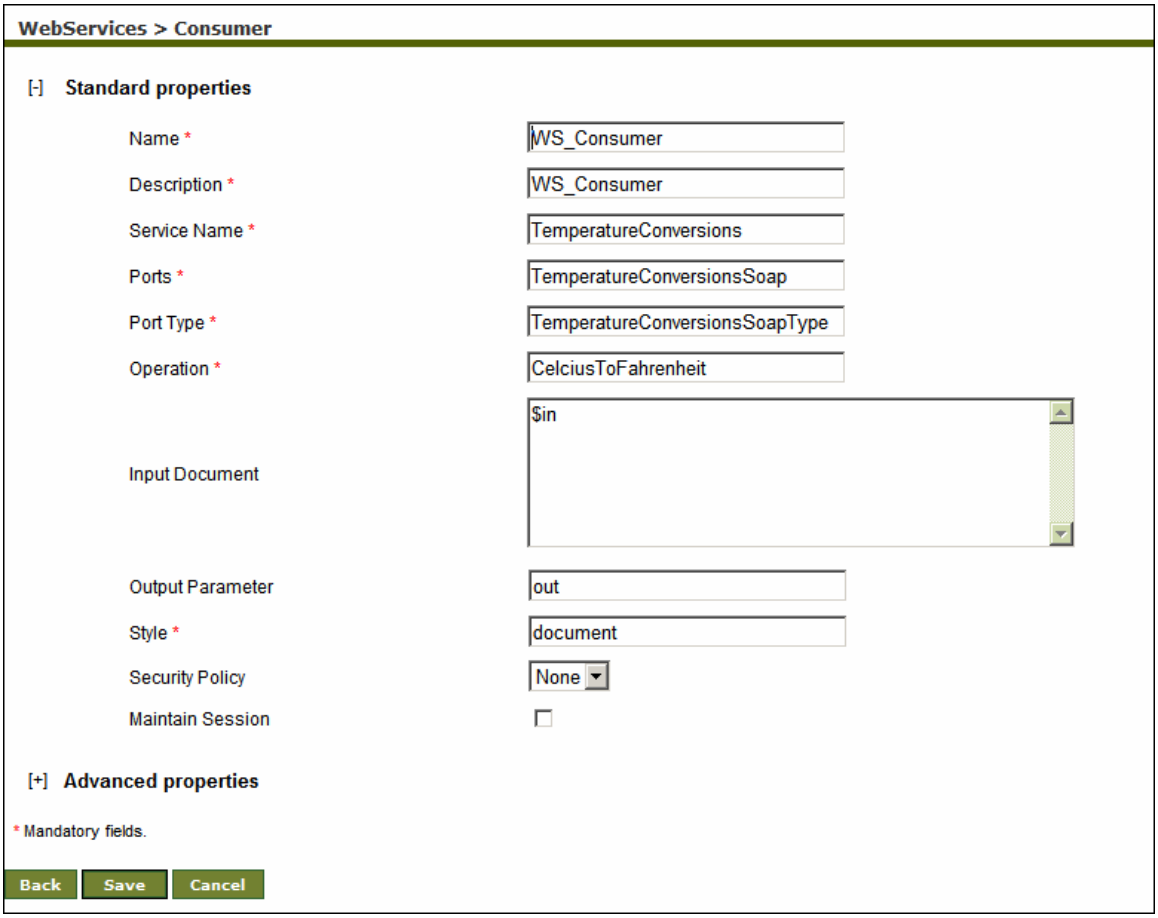


Figure 9.14: Enter Input and Output Parameters

18. Enter the input value in the *Input Parameter Value* field. Input Parameter is used to define the input data for WS Consumer activity. The actual value could be given in two ways, as shown in the table below.

Table 9.4: Input Parameter Value for WS Consumer Activity

Input Value	Description
Definition Time	Here the actual value is given based on the data type of the parameter.
Run Time	Here a variable prefixed with $ is defined in the input document field. For example $variable1. The value of variable1 will be searched in process flow context during process flow execution as the control reaches web service consumer activity. The value will be of Java primitive type.

19. At execution time Ws consumer activity will search for its input in following order:

    1. Search variable in process flow context
    2. Get XML document

The way input is passed to the WS Consumer activity depends upon the style *document* and *rpc* not on WSDL locator UDDI or URI.

20. Enter relevant variable in the *Output Parameter Name* field.

Do not prefix $ with the variable name defined in *Output Parameter Name* field.

21. Select the required Security Policy activity from the *Security Policy* drop-down list.

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

22. Click **Save** button. This displays a screen confirming that the web service consumer has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the web service consumer (refer to Figure 3.6).
23. Enter the comments in the *Add Comments* 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 web service consumer has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING WEB SERVICE PROVIDER ACTIVITY

Web Service Provider is used to publish process flows that are accessed by Web Service consumers. Once a Web Service is published, it creates a WSDL and makes it available to the Adeptia Suite users. The user can use this WSDL to invoke the Web Service. The Web Service can be published in two modes:

- **Synchronous:** In case the Web Service is published in synchronous mode the consumer waits for the completion of the process flow and hence for the output of the process flow.

- **Asynchronous:** In case the Web Service is published in asynchronous mode the consumer does not wait for the process flow to be completed. Thus consumer gets only a Co-relation ID not the output of the process flow. Later on, using this Co-relation ID, consumer can get the output.

Prerequisites:

- To use Adeptia Suite as Web Service Provider, you must change the value of following property:

  abpm.webservice.host

  Replace its value from localhost to machine name, where Adeptia Suite is running. For more details refer to the section *Adeptia Suite Properties* in *Administrator Guide.*

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	

Steps to create a Web Service Provider Activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Web Services** to expand the tree, and then click **Provider**. The Manage Provider screen is displayed (see Figure 9.15).

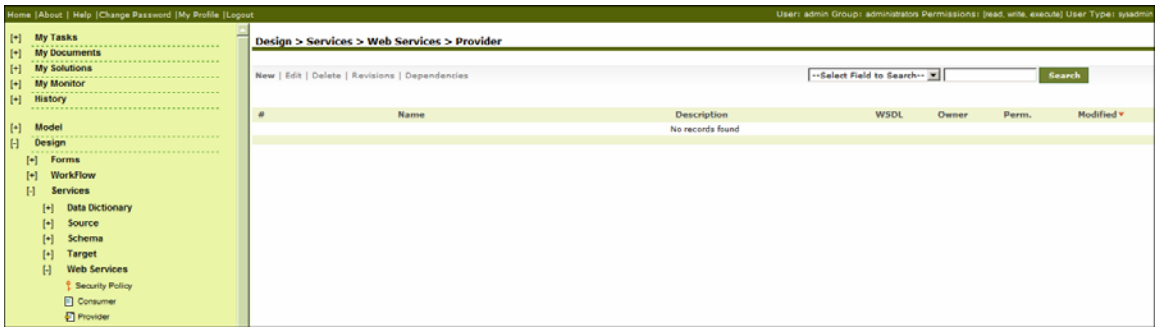


Figure 9.15: Manage Web Service Provider Activity

4. Click the **New** link. The Create Web Service Provider screen is displayed (see Figure 9.16).

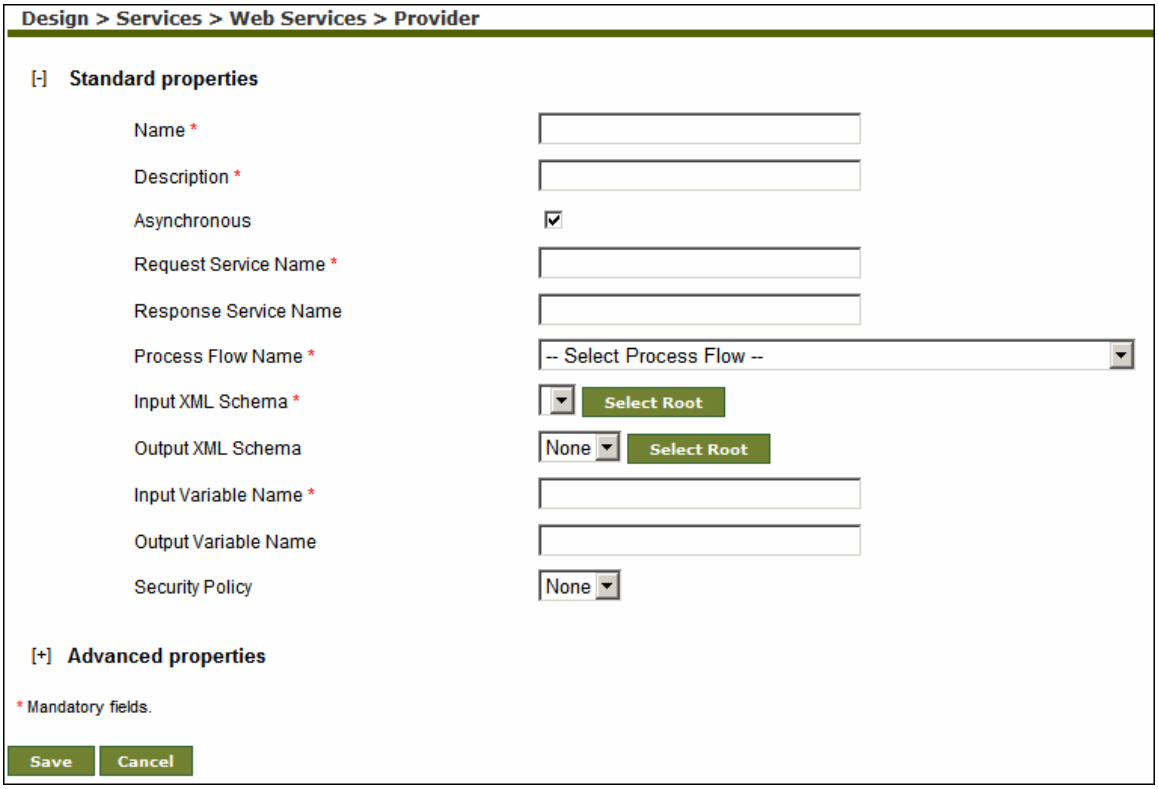


Figure 9.16: Create Web Service Provider Activity

5. Enter the name and description of the new Web Service Provider activity in the *Name* and *Description* fields respectively.

6. Uncheck the *Asynchronous* checkbox if you want to publish Web Service in *Synchronous* mode or else the Web Service will be published in asynchronous mode.

	In case the Web Service is published in *synchronous* mode, one WSDL is available which provides an operation. The operation takes an XML document as an input passes it to the process flow, which processes the data and generates an output XML document. This output XML document is returned back to WS Consumer.

	In case the Web Service is published in *asynchronous* mode, two WSDL's are available. First WSDL provides a web service to post the data and other to retrieve the data. First the WS Consumer posts the data (XML Document) to the Web Service and gets the correlation Id in form of an XML document. The actual data is not returned as the Web Service may take hours or days to process the input data. The second WSDL provides a web service to retrieve the data processed by the first web service. The consumer passes input as the correlation Id XML document obtained from output of first web service and get the processed data In *asynchronous* case the same data exchange took two steps where as in *synchronous* only one. The main advantage of the first one is loose coupling.

7. Enter the request service name and response service name in the *Request Service Name* and *Response Service Name* respectively. The Web Service will be published with the respective service name given in *Request/Response Service Name* field.

	In case the *Asynchronous* checkbox is unchecked, the *Response Service Name* field will be disabled.

8. Select the process flow, which you want to publish as Web Service from the *Process Flow Name* drop-down list.
9. Select Input XML Schema from the *Input XML Schema* drop-down list. This XML Schema corresponds to the XML Input provided by Web Service consumer activity.
10. Select the Output XML Schema from the *Output XML Schema* drop-down list.

	If case, selected XML Schema is having multiple roots, click the **Select Root** button and select the required root.

11. Enter the Input and Output Variables in the *Input Variable* and *Output Variable* fields respectively.
12. Select the Security Policy activity from the *Security Policy* drop-down list.

	If any security policy is not selected, then the web service is published in anonymous mode. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

13. Click **Save** button. This displays a screen confirming that the Web Service Provider activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the web service provider activity (refer to Figure 3.6).
14. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

15. Click **OK** to save the comments. This displays a screen confirming that the Web Service Provider activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	In case the process flow is published as *asynchronous* web service, the Manage Web Services Provider screen looks like as displayed in Figure 9.17.

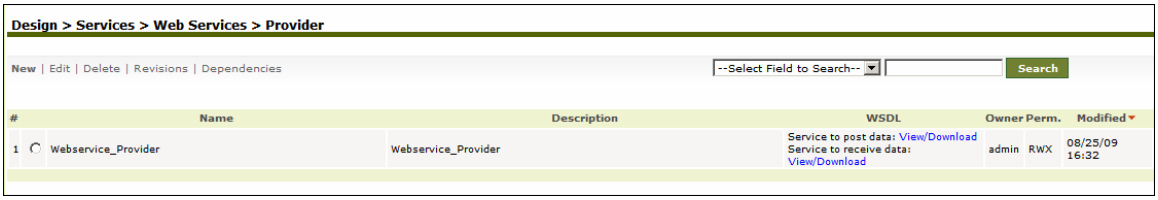


Figure 9.17: Manage Web Service Provider (asynchronous web service)

10 TRANSFORMING DATA

Adeptia Suite can transform any type of data into another type i.e. it facilitates any-to-any transformation. For example, the Adeptia Suite can transform a text file into an XML file. Data transformation can be done in two ways. These are outlined as:

- **Using Data Mapper**
- **Using Record to Record Service**

USING DATA MAPPER

Data Mapper is a visual data-mapping tool used for mapping elements in advanced data integration projects. This tool allows you to specify source and target schemas and to map data fields from source schema to target schema.

The Data Mapper tool is used to map source schema elements to target schema elements. You can map one source schema element to a target schema element directly using the drag and drop approach. Additionally, this tool also offers certain mapping functions using which you can map source and target schema elements.

Furthermore, the Data Mapper supports multiple source and target schemas. This implies that you can select more than one schema at a time, both at the source and the target end. This facilitates mapping of multiple source and target schema elements.

In the Adeptia Suite, this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Accessing Data Mapper

Pre- Requisites

- JRE 1.5 needs to be installed on your system to open the Data Mapper applet.
- The *Pop-up Blocker* needs to be disabled in the web browser, to open the Data Mapper applet. By default, the *Pop-up Blocker* is enabled.

Steps to access Data Mapper

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed (see Figure 10.1).

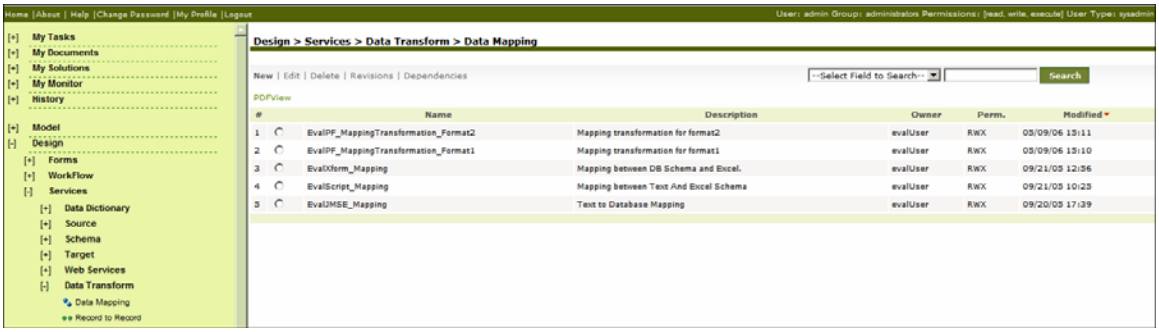


Figure 10.1: Manage Data Mapping

4. Click **New** link. The Create Data Mapping screen is displayed (see Figure 10.2).

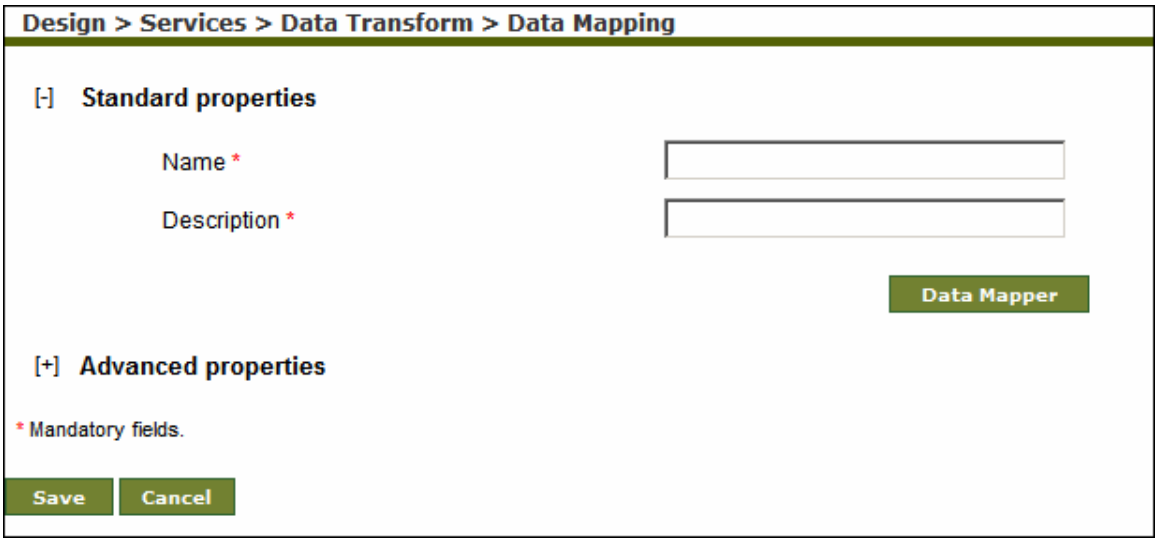


Figure 10.2: Create Data Mapping

5. Enter the name and description of the new mapping activity in the *Name* and *Description* fields respectively.
6. Click **Data Mapper** button. This displays the Data Mapper screen (refer to
7.
8.
9. Figure 10.3).

	If you are starting the Data Mapper on your system for the first time, then a warning message is displayed that prevents you from starting this application. Ignore this message and click **Start** to continue.

	If you have *read-only* permissions, you can view a mapping activity in read-only mode. You can view the applied mapping, create new mapping rules, edit existing mapping rules and even run the simulation. For details, refer to the **View Mapping in Read-Only Mode** section.

10. Click **Advanced Properties** if you want to set options for splitting data. You can split the source data into different chunks. You can specify the size of chunks in terms of the number of records. In the output however, only one file is generated at the location specified in the target activity. To know more about splitting data, refer to the **Splitting Source Data** section.

	You can set the **Advanced Properties** at any time of the mapping activity.

11. Once you have set the Advanced Properties, click **Save** to save the splitting options. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the mapping transformation. (refer to Figure 10.35).
12. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

13. Click **OK** to save the comments.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Understanding Data Mapper Applet

The Data Mapper screen is displayed below (see

Figure 10.3).

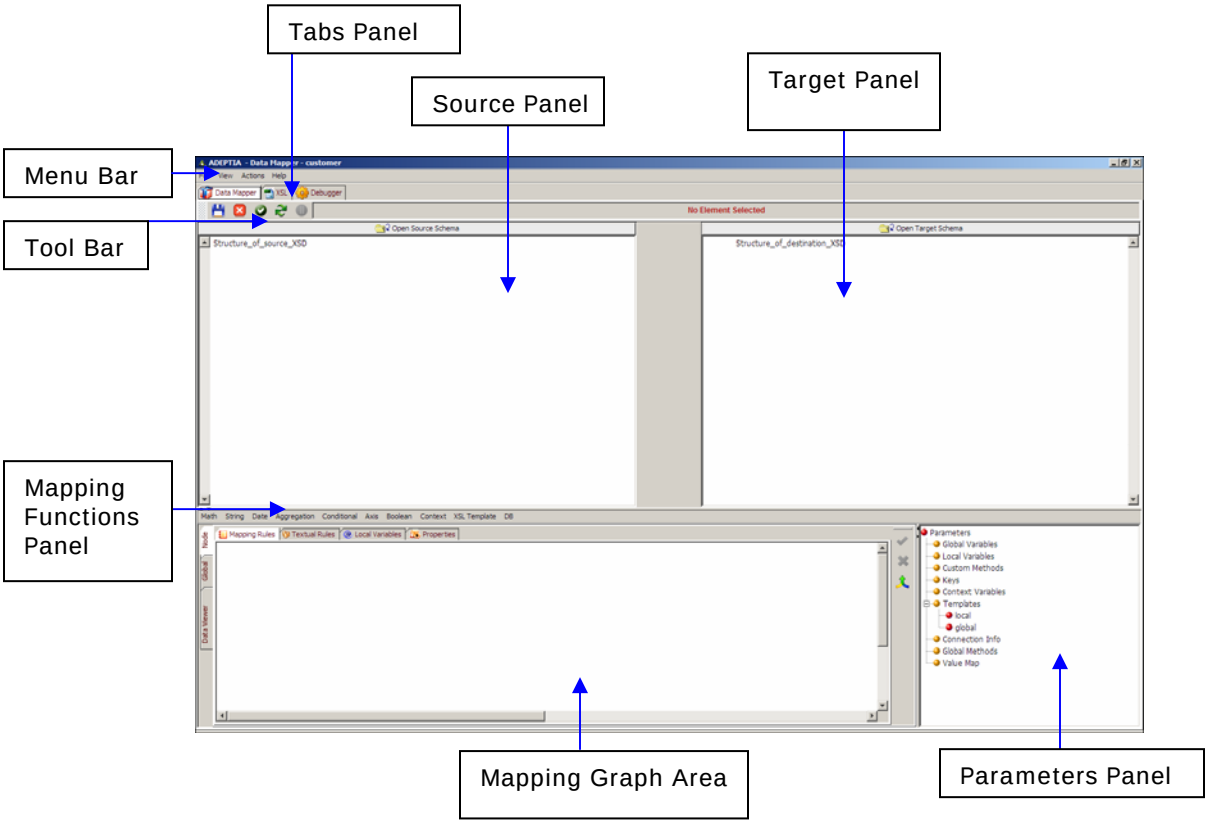


Figure 10.3: Data Mapper Applet

The Data Mapper screen is divided into eight sections. These are outlined as:

- **Menu Bar**
- **Tabs Panel**
- **Tool Bar**
- **Source Panel**
- **Target Panel**
- **Mapping Functions Panel**
- **Mapping Graph Area**
- **Parameters Panel**

Menu Bar

Options of the Menu Bar are explained in the table below.

Table 10.1: Options of Menu Bar

Menu Option	Sub-Option	Function
File	Save	Save mapping activity
	Exit	Exit Data Mapper
View	Enable Tool Tips For Source/Target Tree	Activate or deactivate Tool Tips to be displayed at source or target leaf elements. Tool Tips are always displayed as active for root elements.
	Move Connection Ends	Set the width of connecting lines. By default, it is set to 50px.
	View Mapping XML	Display generated XML file with line numbers. It is displayed in read-only mode.
Actions	Validate XSL	Validate the generated XSL file.
	Global Custom XSL Before	Add custom XSL code at the top of the mapping XSL.
	Global Custom XSL After	Add custom XSL code at the bottom of the mapping XSL.
	On Demand (optimized) loading of XML Schema tree	If the XSD is complex, then this option allows you to process only the root level element and not the complete XSD. Only when you expand the tree elements, it processes the child elements and loads them into the memory. The tree elements expansion is now displayed in the applet. This option is applicable only on XML schemas. It optimizes the processing of XML schemas and loads the XSD elements only up to 2 levels in the source and target panels. You need to click a tree element to load its further child elements.
	Namespace	Manage Namespace
	Enable DBQuery caching	Enable DBQuery cache. This implies that if the same DBquery is fired on another element, it will generate the result from the existing query, instead of creating a new connection. By default, the DBQuery cache is disabled.

	Search Element	Search an element in the Source or Target schema tree.
	Set Data Viewer Record Count	Set maximum number of records that can be shown in data viewer
	Add Custom Auto Map Options	Add custom options for Auto Mapper
	Add Value Map Options	Add options for value map
	Enable Quick Splitting	Splits data at source end and merges at target end automatically
Help	Data Mapper Help	Display online help for the Data Mapper section.
	Mapping Functions Help	Display online help for mapping functions.
	About Data Mapper	Display the About box for the Data Mapper tool.

Tabs Panel

Options of the Tabs Panel are explained in the table below.

Table 10.2: Options of Tabs Panel

Button	Name	Function
	Data Mapper	Display the Data Mapper screen for mapping source and target elements.
	XSL	Display the generated XSL code with line numbers for the mapping activity.
	Debugger	Validate and view output of the mapping activity.

Tool Bar

Options of the Tool Bar are explained in the table below.

Table 10.3: Options of Tool Bar

Button	Name	Function
	Save Mapping	Save mapping activity
	Remove All Mappings	Remove all the mapping between the source and the target elements

	Validate XSL	Validate generated XSL
	Auto Mapper	Map the source and the target elements if the structure of the source and the target schemas and the names of the source and the target elements are similar
	One to One Mapping	Map all the source and the target elements under a parent element if the number of elements in the source and the target schemas are same
	Open Source Schema	Open source schema
	Open Target Schema	Open target schema

Source Panel

The Source Panel is used to display a source schema. All the elements of the source schema are listed in the Source Panel. When multiple source schemas are loaded, then all elements of each schema are listed under their respective source nodes.

Target Panel

The Target Panel is used to display a target schema. All the elements of the target schema are listed in the Target Panel. When multiple target schemas are loaded, then all elements of each schema are listed under their respective target nodes.

Mapping Functions Panel

The Mapping Functions Panel displays all mapping functions that can be used to map source and target schema elements. Mapping functions are supported by XSLT. However, the Data Mapper also includes some customized functions, which are not standard of XSLT. Refer to **Using Mapping Functions** for details on mapping elements using these functions.

In addition to the mapping functions, an option of XSL Template is also displayed on the Mapping Functions Panel. This option enables you to **Add XSL Template** and further manage XSL templates from the Manage XSL Template screen.

Mapping Graph Area

The Mapping Graph Area is used to map the source and the target elements. You can apply the mapping function between source and target elements in this section. Mapping of the selected

target element is displayed in the Mapping Graph Area. This mapping is displayed in the form of Mapping Rules (graphical representation) or as Textual Rules (code representation). Additionally, you can view and **set target element properties** such as **adding comments** for target elements and also repeat occurrences of a target element based on the occurrences of a source element, by applying the **For Each property**. Further, you can **apply sorting rules** for elements and also set the Disable-Output-Escaping property.

Additionally, you can create **Local**, **Global** and **Context Variables** in Mapping Graph Area. You can also define **Custom Methods** and **Key Functions** in this section.

The Mapping Graph Area comprises of two tabs. These tabs further list sub tabs using which you can perform various functions. These are outlined in the table below.

Table 10.4: Tabs of Mapping Graph Area

Tab	Sub Tabs	Function
Node	Mapping Rules	Display graphical representation of the mapping of source and target elements.
	Textual Rule	Display the textual code representation of the mapping of source and target elements.
	Local Variables	Create and define local variables for mapping source and target elements.
	Properties	Display and set properties for source and target elements. Enables you to add comments for elements and applies sorting rules for target elements. It allows you to set the *Disable-Output-Escaping* property.
Global	Global Variables	Create and define global variables for mapping source and target elements.
	Custom Methods	Create global methods for mapping source and target elements.
	Key Functions	Create and define keys for mapping source and target elements.
	Context Variables	Create and define context variables for mapping source and target elements.
	Connection Info	Create and define connection info variables for mapping source and target elements.
Data Viewer	Source	Preview Data of the uploaded file at the source end.
	Target	Preview Data at the target end according the uploaded source file and mapping rule applied.

The Mapping Graph Area also comprises of buttons. These buttons are displayed based on the selected tabs. All buttons of the Mapping Graph Area are explained in the table below.

Table 10.5: Buttons of Mapping Graph Area

Button	Name	Function
	Apply Mapping	Map the selected source elements to the selected target elements.
	Remove Mapping	Remove the mapping between the selected source and the target elements
	Pop Out Parameters Panel	Pop out the Parameters Panel and display it as maximized.
	Add Global/Local/Context Variable/Key	Add a new global, local, context variable or a key.
	Save Global / Local /Context Variable/Key	Save a new local, global, context variable or a key.
	Remove Selected Global/Local /Context Variable/Key	Remove the selected global variable, local variable, context variable or key.
	Remove all Global/Local/Context Variables/Keys	Remove all existing global variables, local variables, context variables or keys.
	Clears Global / Local variable name and value text fields	Clear all data entry fields while adding a local or global variable.
	Add New Method	Add a new custom method.
	Remove Selected Method	Remove the selected custom method.

You can expand the Mapping Graph Area by clicking **Maximize** () button displayed on the Split Bar below the Source and Target Panels. Similarly, by clicking **Minimize** () button, you can restore the Mapping Graph Area to its original size. Alternately, you can drag the Split Bar to resize the Mapping Graph Area and Source and Target Panels.

Parameters Panel

The parameters listed in the Parameters Panel are explained in the table below.

Table 10.6: Parameters Listed in the Parameters Panel

Parameters	Description
Global Variables	Display a list of all the global variables defined for the mapping of source and target elements in the current mapping object.
Local Variables	Display a list of all the local variables defined for the mapping of source and target elements in the current mapping object.

Custom Methods	Display a list of all the methods defined for the mapping of source and target elements in the current mapping object.
Keys	Display a list of all the keys defined for the mapping of source and target elements in the current mapping object.
Context Variables	Display a list of all the context variables defined for the mapping of source and target elements in the current mapping object. All context variables created in the Data Mapper are transferred to the Process Flow Designer.
Templates	Display a list of all the XSL templates created for the mapping of source and target elements in the current mapping object.
Connection Info	Display a list of all connection info variables.
Global Methods	Display a list of all class files present in the Custom Classes folder.
Value Map	Display a list of value maps created for the elements

All parameters are displayed in this section as and when they are created in the mapping process.

You can remove a parameter by right-clicking the parameter and selecting the Remove option.

You can expand the Parameters Panel horizontally by clicking **Maximize** () button displayed on the Split Bar left to the Parameters Panel. Similarly, by clicking **Minimize** () button, you can restore the Parameters Panel to its original size. Alternately, you can drag the Split Bar to resize the Mapping Graph Area and Parameters Panel.

Mapping Elements

The process of mapping elements comprises of various steps. These are outlined as:

- **Load Source and Target Schemas**
- **Map Source and Target Elements**
- **Save Mapping and Exit Data Mapper**

Load Source and Target Schemas

Steps to load Source and Target Schemas

1. Click **Open Source Schema** ( ) button displayed on the Tool Bar. The Select Schema screen with a list of existing source schemas is displayed (see Figure 10.4).

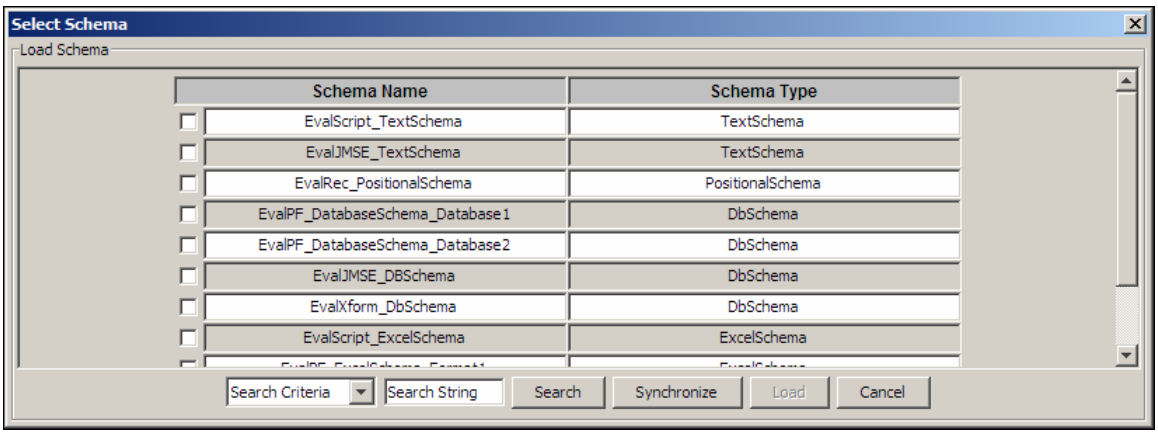


Figure 10.4: Select Schema

	If you have created a new schema, then it will not be displayed in this list. To refresh the list of schemas and add the new schema to this list, click **Synchronize** button.

2. Select the checkbox of the source schema that you want to load. This will enable the **Load** button.

	You can select more than one schema at a time to load multiple schemas. At times, the list of schemas is very long. In such a case, you can search for schemas to be loaded, on the basis of the *schema name* or *type* by selecting the appropriate search criteria from the drop-down list and entering the search criteria in the text box. Clicking **Search** button will display only those schemas that conform to the entered criteria. You can also enter wildcard characters like '*', '?' and '[]' in the search criteria. You need to click the **Synchronize** button to synchronize the schema list before using the *Search* feature.

3. Click **Load** button. This loads the selected source schemas with all their elements in the Source Panel.

4. Click **Open Target Schema** ( ) button displayed on the Tool Bar. This displays the Select Schema screen with a list of existing target schemas (refer to Figure 10.4).

5. Select the checkbox(s) of the target schema(s) that you want to load.

6. Click **Load** button. This loads the selected target schemas with all their elements in the Target Panel.

	The schema name is displayed against the root node of the source and target trees. This is useful in case of loading multiple source and target schemas.

7. Click the **Expand** (+) button to expand and display all elements of the selected schemas in their respective panels (see Figure 10.5).

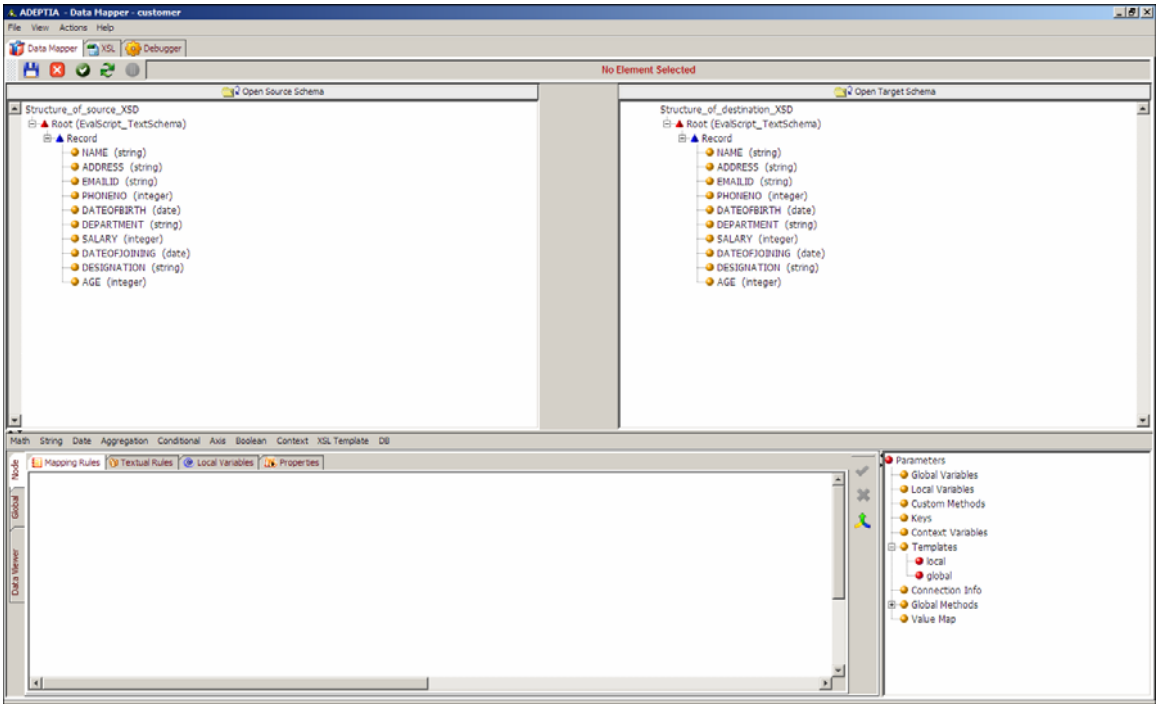


Figure 10.5: Source and Target Schema Elements

	The Data Mapper tool allows you remove a loaded schema. For details, refer to the section **Removing a Schema**.

Elements in the Source and Target Panels are identified by the legends displayed before the name of the element. These legends are explained in the table below.

Table 10.7: Legends of Elements

Legend	Description
	Root Element
	Parent/Complex Element
	Leaf Element
	Attribute Element

	If no schema is loaded, then all tabs of the Data Mapper appear as disabled.

Search Element in Source or Target Schema Tree

At times, when the source or target schemas are quite large, then searching an element can be very cumbersome. The Data Mapper applet eases this task by allowing you to search for an element in a source or target schema.

Steps to search an element in a source or target schema tree

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click the **Actions** menu and select **Search Element** option (see Figure 10.6). Alternately you can press **<Ctrl> + <F>** on the keyboard.

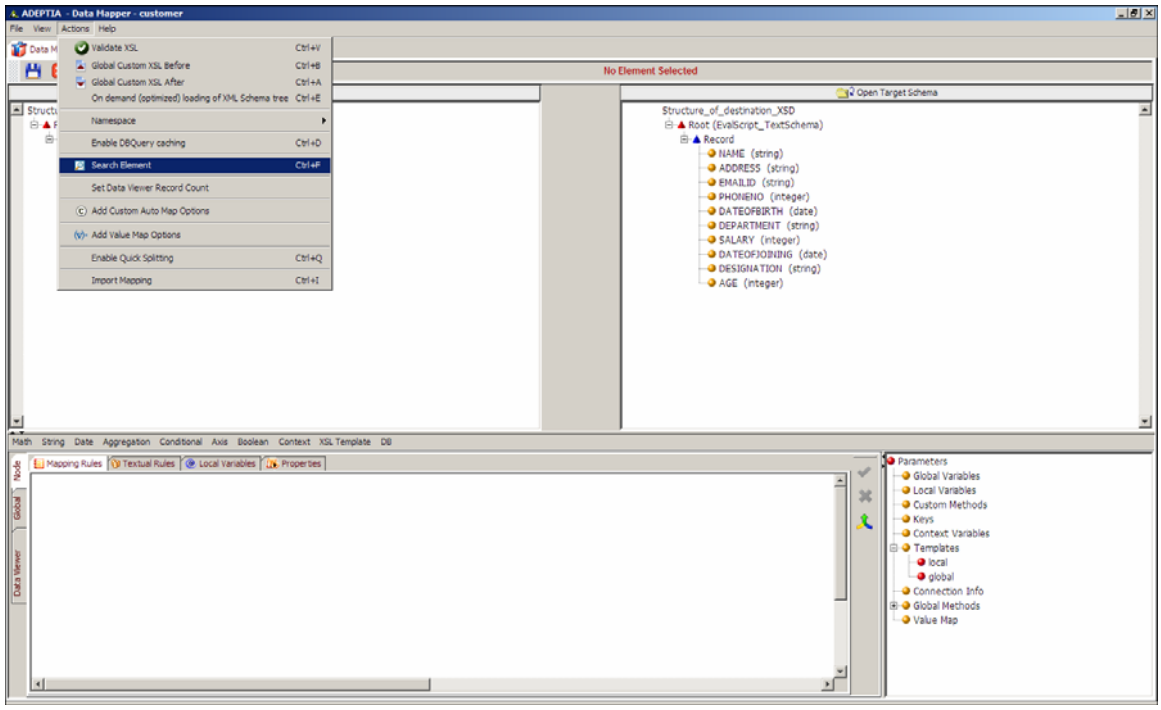


Figure 10.6: Select Search Element from Actions Menu

3. The Element Search dialog box is displayed (see Figure 10.7).

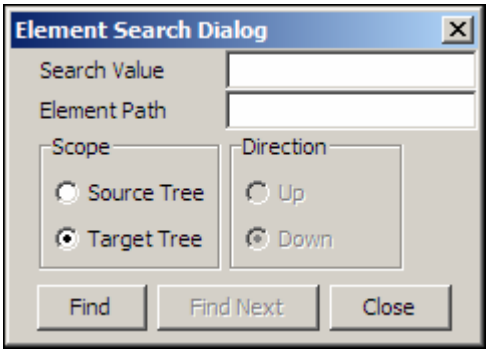


Figure 10.7: Element Search Dialog Box

4. Enter the name of the element that you want to search for, in the *Search Value* field. You can also use Wildcard character(s) to search for an element. The Xpath of the element is automatically displayed in the *Element Path* field.

5. Select the appropriate option in the *Scope Panel*, to search the element in the Source or Target schema tree. For example, if you want to search in the Source schema tree, then select *Source Tree* option. Only one option can be selected at a time. By default, *Target Tree* is selected (see Figure 10.8).

Figure 10.8: Enter Parameters in Element Search Dialog Box

6. Click **Find**. This searches for the first occurrence of the element in the selected tree. If the element match is found, then that element is selected and highlighted. The Xpath of that node is displayed in the *Element Path* field (see Figure 10.9).

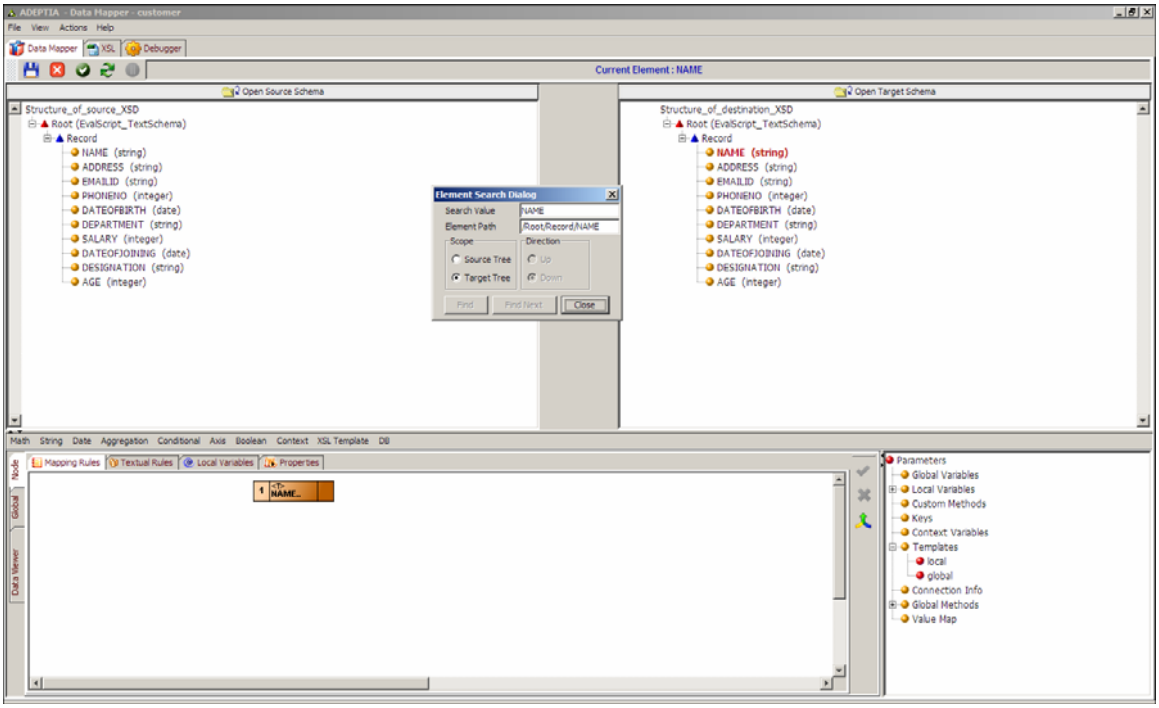


Figure 10.9: Element Match Found

7. If multiple matches are found for the search criteria, then the *Direction Panel* and the **Find Next** button is activated (see Figure 10.10). You can select the direction in which you want to search in the selected tree. For example, if you want to search upwards in the tree, select *Up* option. Only one option can be selected at a time. By default, *Down* is selected.

Figure 10.10: Multiple Matches Found

8. Click **Find Next** to search for the next element in the tree, based on the selected direction. Once the element match is found, click **Close** to close the Search Element dialog box. In case no element match is found, then a warning message is displayed (see Figure 10.11).

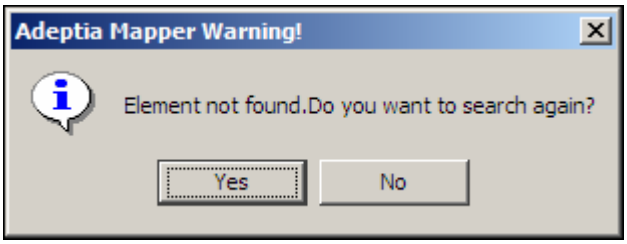


Figure 10.11: Warning Message

9. Click **Yes** to search again, else click **No** to close this screen.

Map Source and Target Elements

You can map source and target elements using any of the listed methods:

- **Drag and Drop Approach**
- **Buttons on the Tool Bar**
- **Copy/Paste Mapping**
- **Mapping Functions**

Map Elements Using Drag and Drop Approach

This is the default and most commonly used method for mapping source and target elements.

Steps to map elements using Drag and Drop approach

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click a source element and drag it to the desired target element. A line is displayed, indicating the mapping between the selected source and target element. The graphical representation of the above mapping is displayed in the Mapping Graph Area (see Figure 10.12).

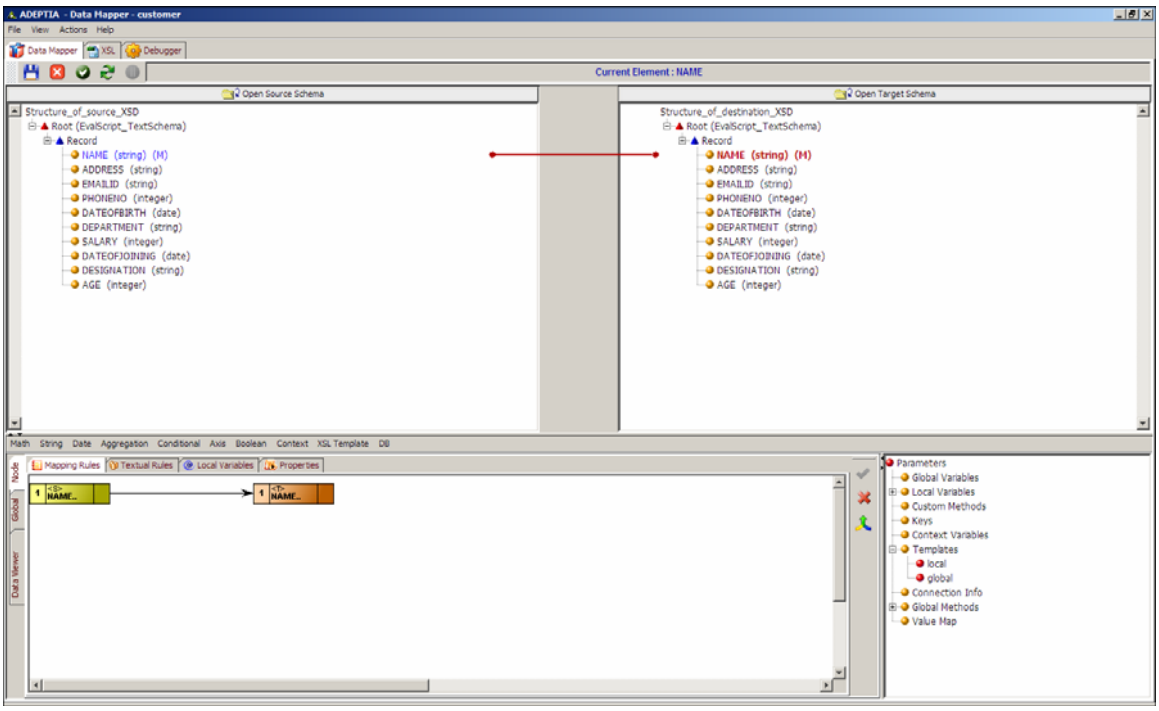


Figure 10.12: Map Source and Target Element using Drag and Drop Approach

	Once a source and target element is mapped, the letter **(M)** is displayed next to each mapped source and target element. This signifies that the target element has been mapped to a source element. Refer to **Table of Suffixes** for details on suffixes displayed next to an element.

3. Similarly, drag and drop each source and target element that you want to map. The mapping between all source and target elements will be displayed (see Figure 10.13).

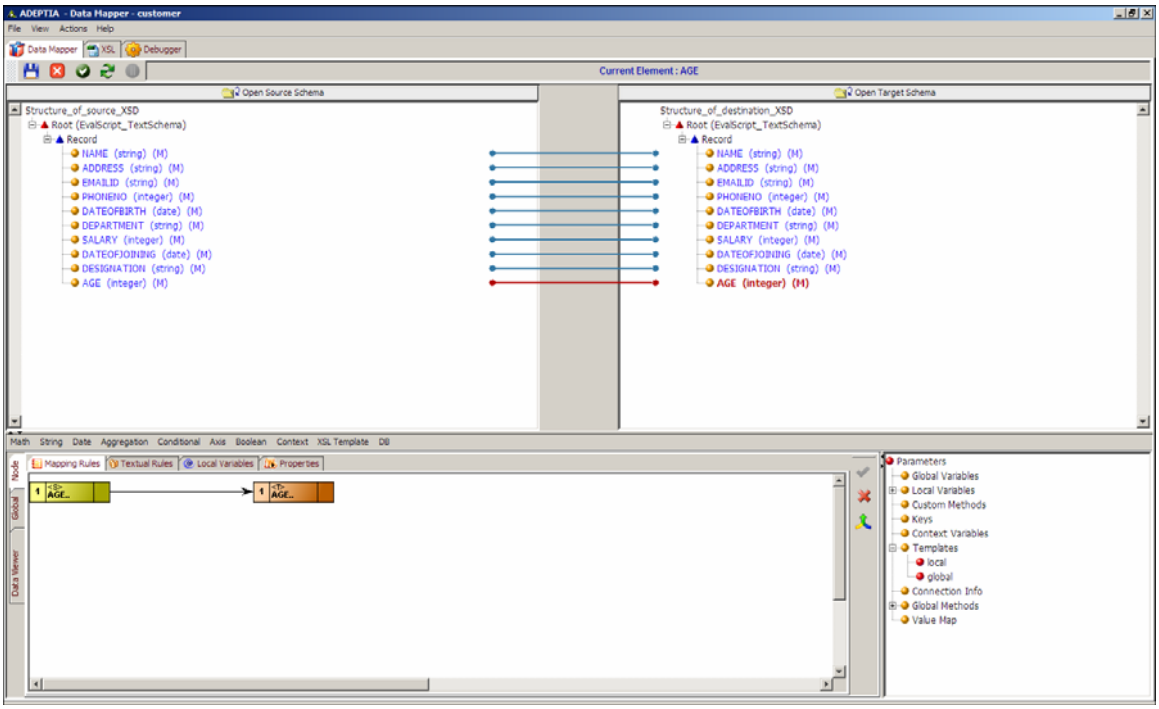


Figure 10.13: Map Source and Target Elements

4. If you have loaded multiple source and target schemas, then drag and drop the source and target elements you want to map. The mapping between these elements will be displayed (see Figure 10.14).

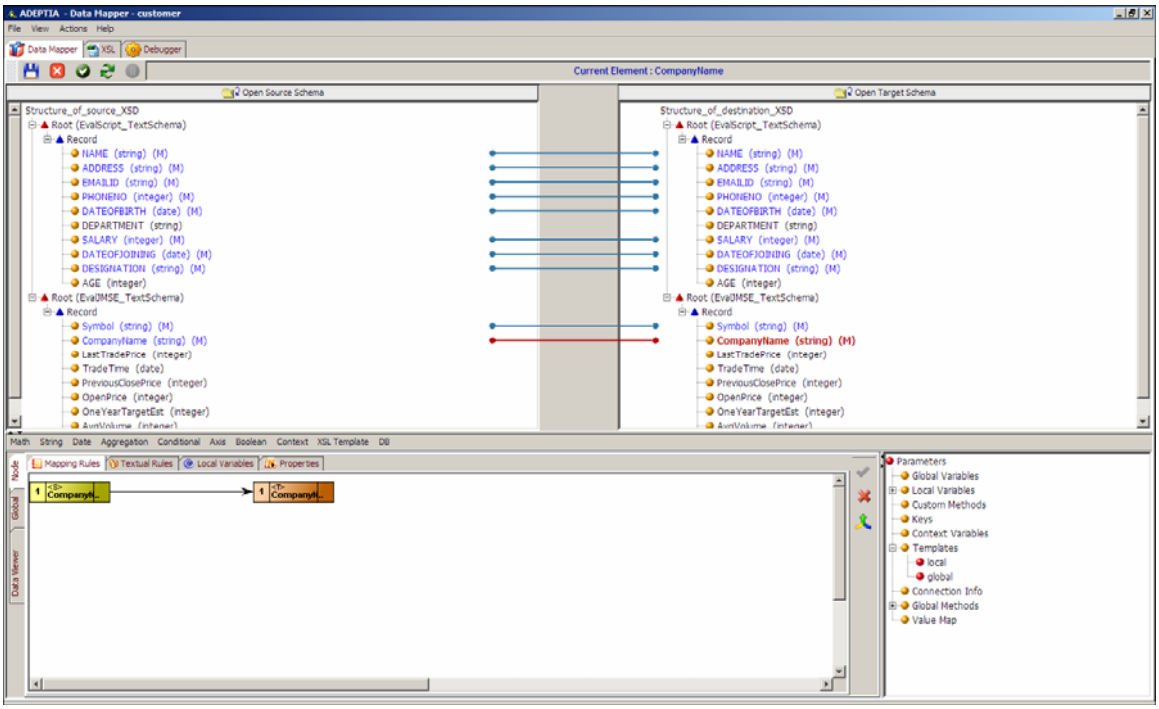


Figure 10.14: Map Multiple Source and Target Schema Elements

5. **Save** the mapping activity and exit the Data Mapper.

	If you are mapping multiple source and target schema elements, then you need to **assign data streams**, before saving the mapping activity.

	Once you have mapped source and target elements, you can **view and validate the generated mapping XSL**, from the Data Mapper screen. You can also **view the target XML** and **view and validate mapping output** from this screen.

Map Elements using Buttons on the Tool Bar

You can map source and target elements using certain buttons on the Tool Bar. These buttons are explicated as:

- **One To One Mapping** ()
- **Auto Mapper** ()
- **Remove All Mappings** ()

One to One Mapping

If the number of leaf elements of the parent element in the source and target schema is same, then you can use the One to One mapping button to map all the source elements to the corresponding target elements.

Steps to use One to One mapping

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click and drag the parent element of the source schema to the parent element of the target schema. This will apply template of the source parent element on the target parent element, and activate the **One to One Mapping** button.
3. Click **One to One Mapping** () button. All the leaf elements of the parent element in the source schema will be mapped to their respective leaf elements of the parent element in the target schema.

	Only leaf elements are mapped using this button. It does not map complex elements. To map complex elements, you need to use the **drag and drop** method.

	If multiple schemas are loaded, then dragging a source parent element will map all leaf elements of schemas displayed previous to the current schema. Moreover, in multiple schemas, mapping can be performed between parents at any level, if the number of their leaf elements is the same.

Auto Mapper

If the hierarchy and the names of the complex elements are similar, then you can use the **Auto Mapper** option to map the source elements to the corresponding target elements. In case of complex elements, the hierarchy and the name of all elements must be the same.

Steps to use the Auto Mapper

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes. Additionally, the hierarchy and name of the all elements is the same (see Figure 10.15).

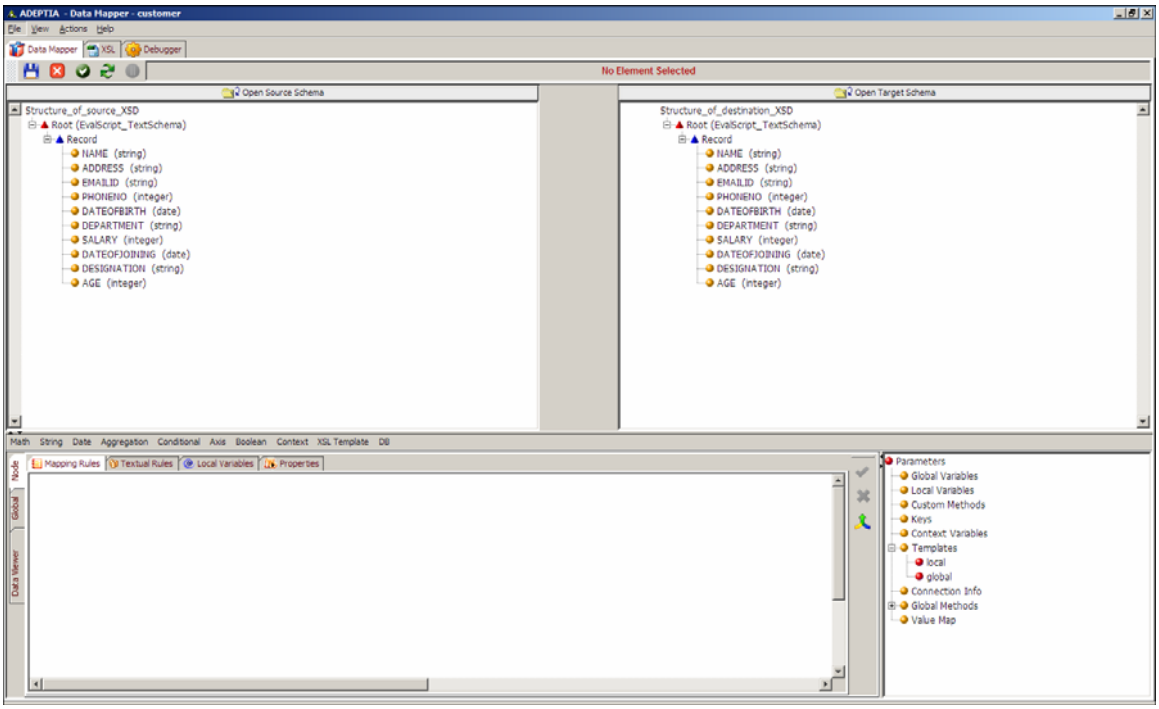


Figure 10.15: Same Hierarchy and Element Names

2. Click **Auto Mapper** ( ) button. The Auto Mapper Options dialog box is displayed (see Figure 10.16).

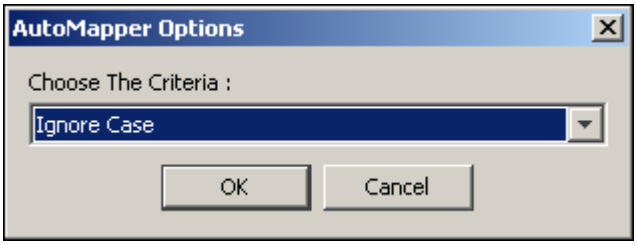


Figure 10.16: Auto Mapper Options

3. Select the mapping criteria from the *Choose the Criteria* drop-down list. Its options are listed as:

   - **Ignore Case:** The source and the target elements are mapped even if they are of different case. Their attributes are not mapped.
   - **Case Sensitive:** The source and target elements are mapped only if they are of the same case. Their attributes are not mapped.
   - **Ignore Case and Include Attributes:** The source and target elements and their attributes are mapped even if they are of different case.
   - **Case Sensitive and Include Attributes:** The source and target elements and their attributes are mapped, only if they are of the same case.
   - **Consider Only Leaf Element Count:** The source and target elements are mapped only if the number of leaf elements in the source schema and target schema are the same. It

is not necessary that names of source and target elements are similar. Their attributes are not mapped.

- **Consider Only Leaf Element Count and Include Attributes:** The source and target elements and their attributes are mapped only if the number of leaf elements in the source schema and target schema are the same. It is not necessary that names of source and target elements are similar.

4. Click **OK** button. All elements of the source and target schema will automatically be mapped based on the selected criteria (see Figure 10.17).

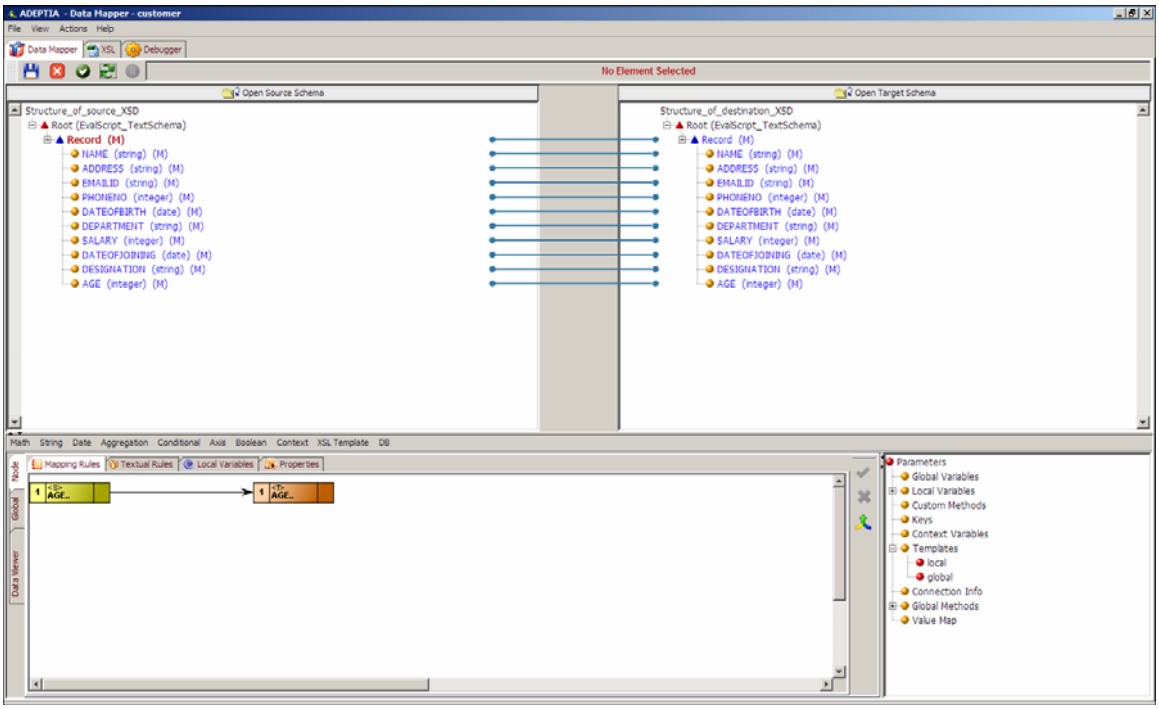


Figure 10.17: Mapped Elements using Auto Mapper

Remove All Mappings

You can use the Remove All Mappings option to remove all mappings from the Data Mapper.

Steps to remove all mappings between source and target elements

1. Click **Remove all Mapping** ( ) button. A Remove Options screen is displayed (see Figure 10.18).

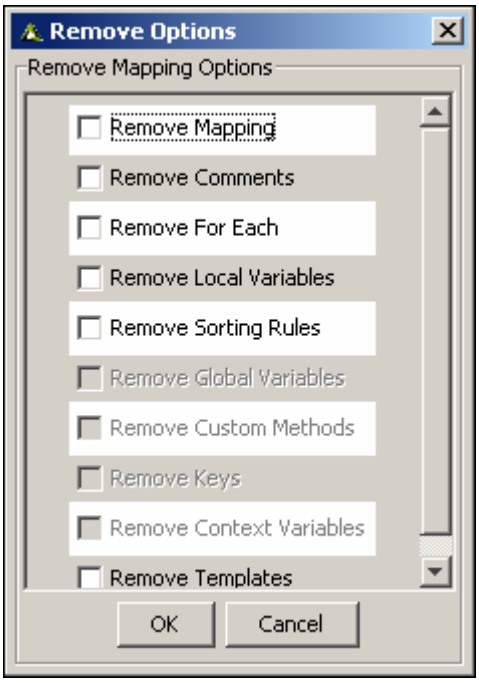


Figure 10.18: Remove Options

2. This screen displays a list of the properties, variables, custom methods, keys and templates associated with a mapped element, which can be removed with the mapping.

	The *Remove Mapping*, *Remove Comments*, *Remove For Each* and *Remove Local Variables* options are always enabled. The other options are enabled only if they have been defined for the mapped element.

3. Select the checkbox(s) of the options that you want to remove and click **OK** button. All the selected options and all mappings between source and target elements will be removed.

	You can remove mapping associated with each element individually. For details, refer to the section **Remove Mapping of an Element**.

Using Custom Auto Map Options

You can use Auto Mapper feature only when the elements of source and target schema are in same hierarchy and have the same names. Now, if you want to automatically map elements of source and target schemas where hierarchy and elements are different, you can use *Custom Auto Map Options.* This option allows you to define custom options for Auto Mapper. For example you can define, if the source element is *EmPName,* it should map to *Name.*

All the Custom Auto Map Options that you create are saved globally and can be used in any mapping activity.

Steps to use Custom Auto Map Options

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes (see Figure 10.19).

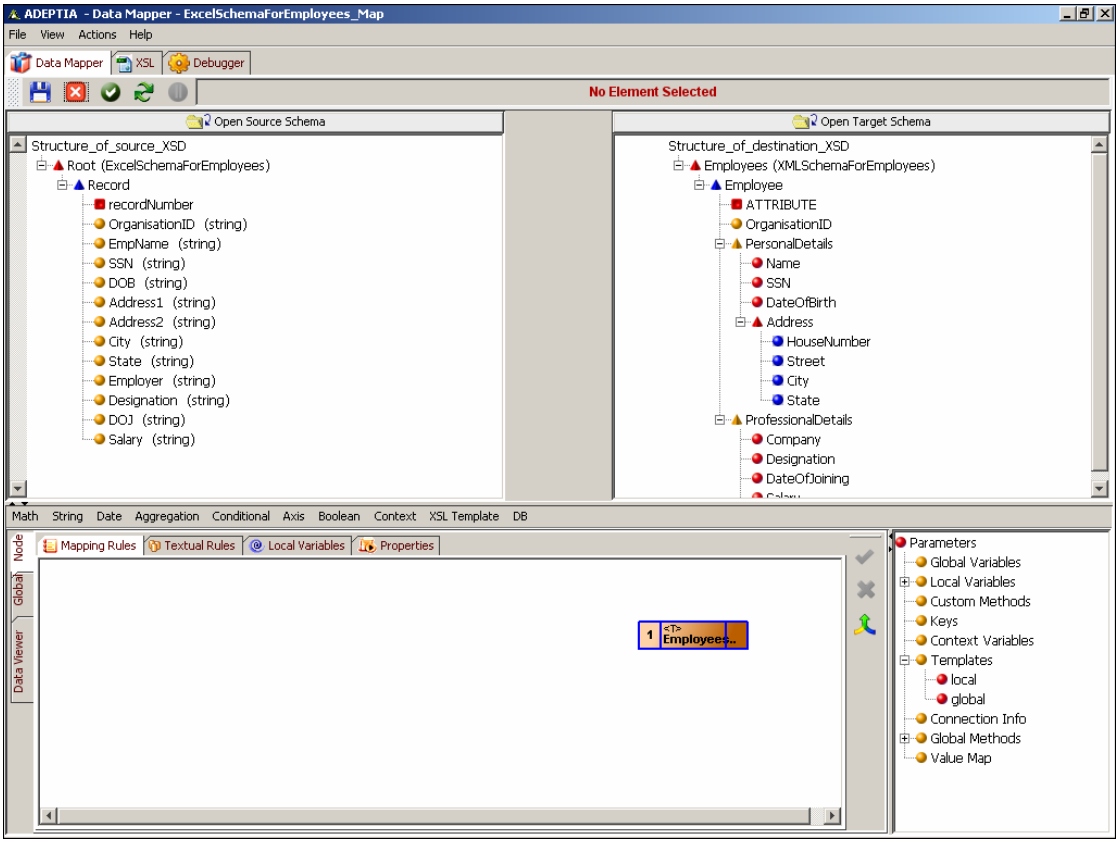


Figure 10.19: Same Hierarchy and Element Names

2. Click **Actions** menu and select **Add Custom Map Options** (see Figure 10.20).

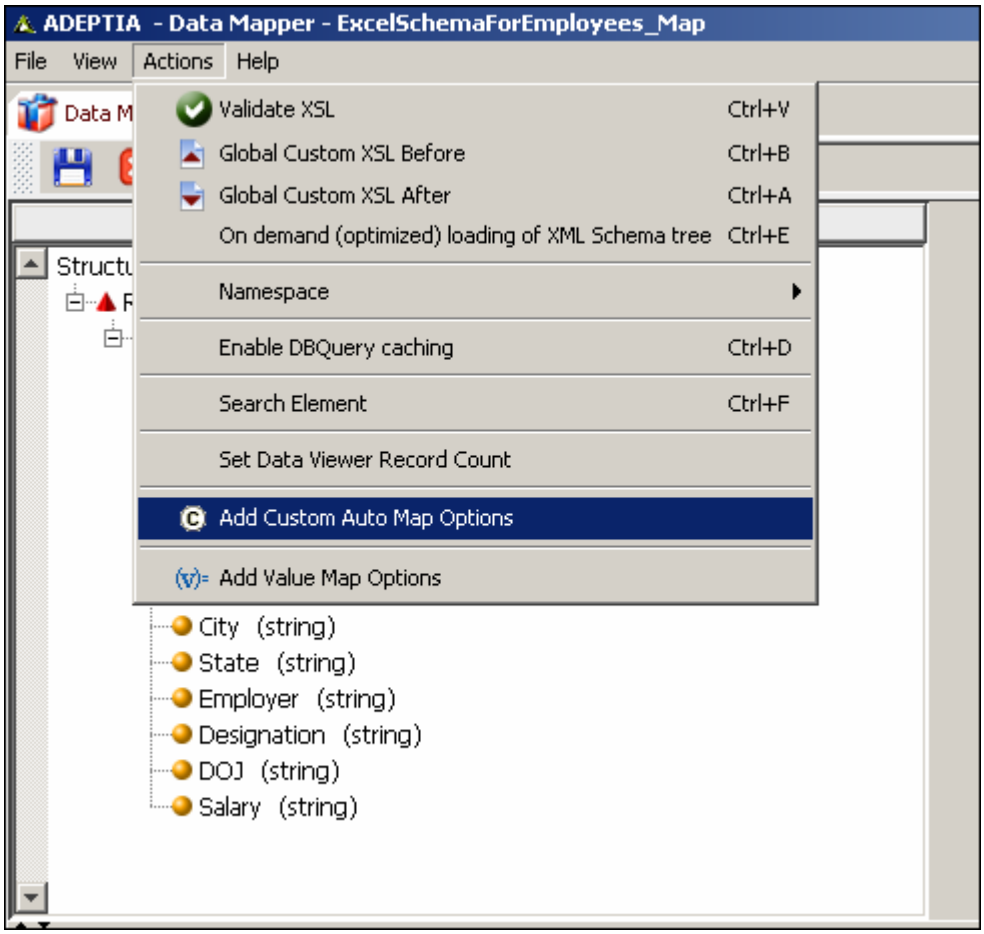


Figure 10.20: Select Add Custom Map Options

3. The Custom Auto Map Options dialog box is displayed (see Figure 10.21).

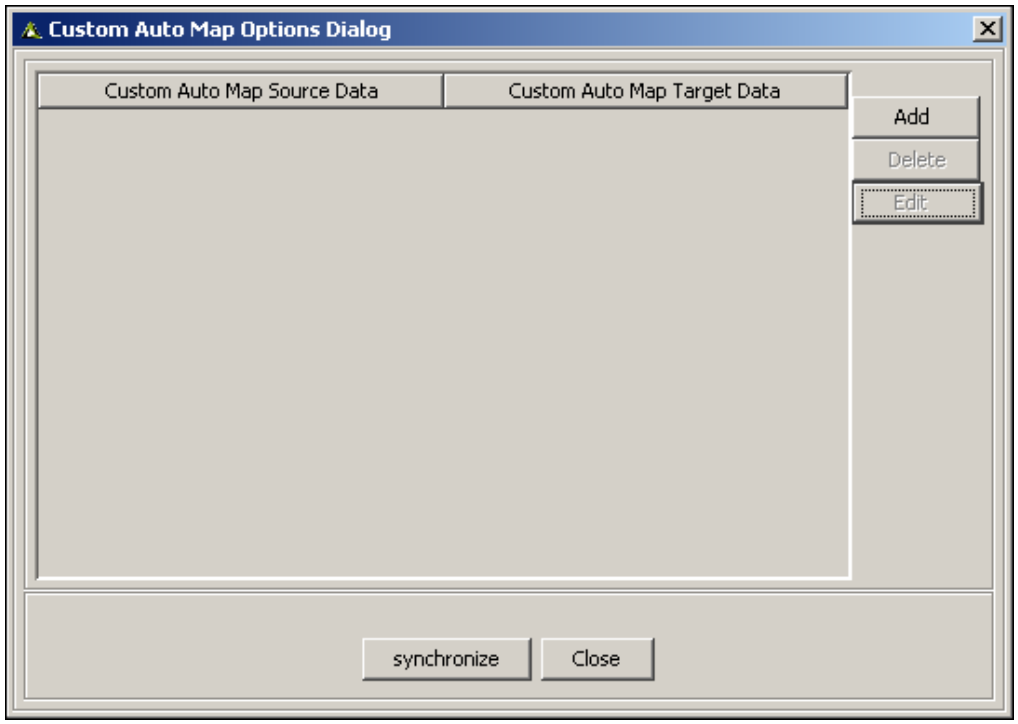


Figure 10.21: Custom Auto Map Options Dialog box

4. Click **Add** button. The Add Data Dialog box is displayed (see Figure 10.22).

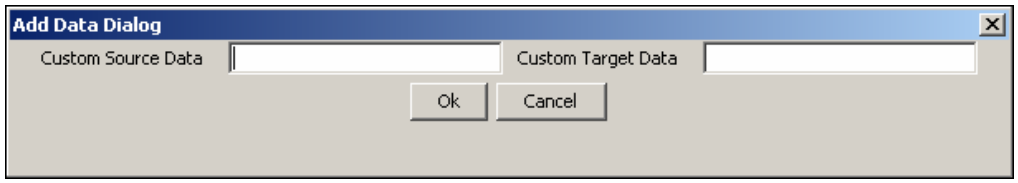


Figure 10.22: Add Data Dialog box

5. Enter the name of the source element for which you want to define custom auto map option in the *Custom Source Data* field.
6. Enter the name of the target element to which you want to map the source element specified in previous step, in the *Custom Target Data* field.
7. Click **OK** to close the Add Data Dialog box. The added entry is shown in Custom Auto Map options dialog box (see Figure 10.23).

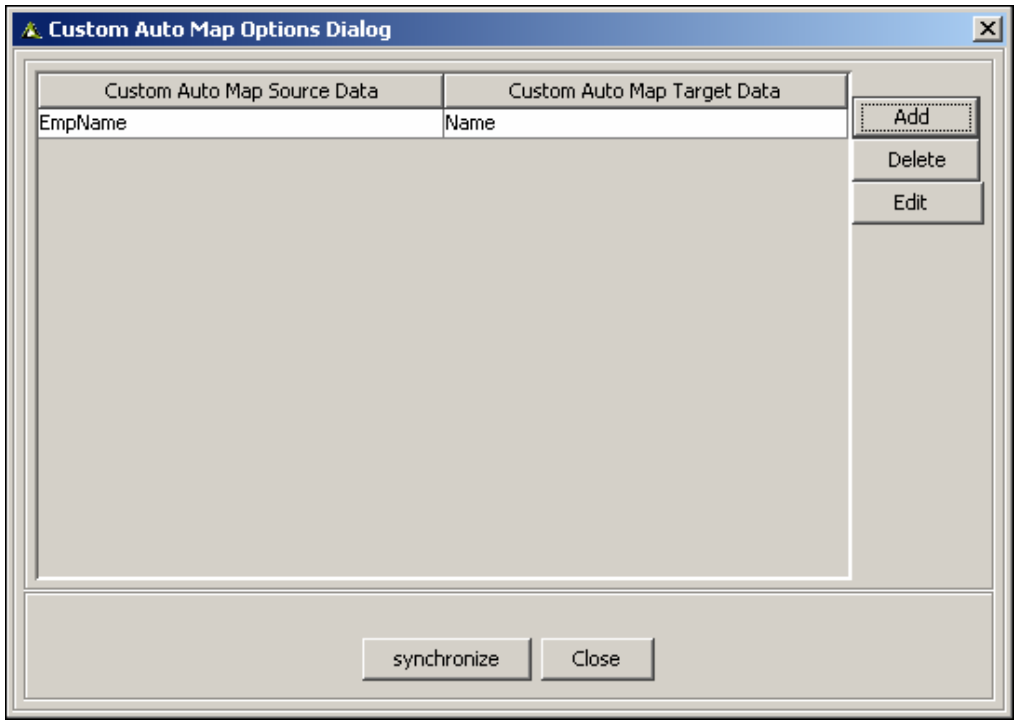


Figure 10.23: Added Entry for Custom Auto Map

8. Similarly you can add other entries.
9. Click **Close** to close this dialog box.
10. Map the parent record of Source and Target Schema.
11. Right-click the parent record of the target schema and select **Connect Matching Record** options (see Figure 10.24).

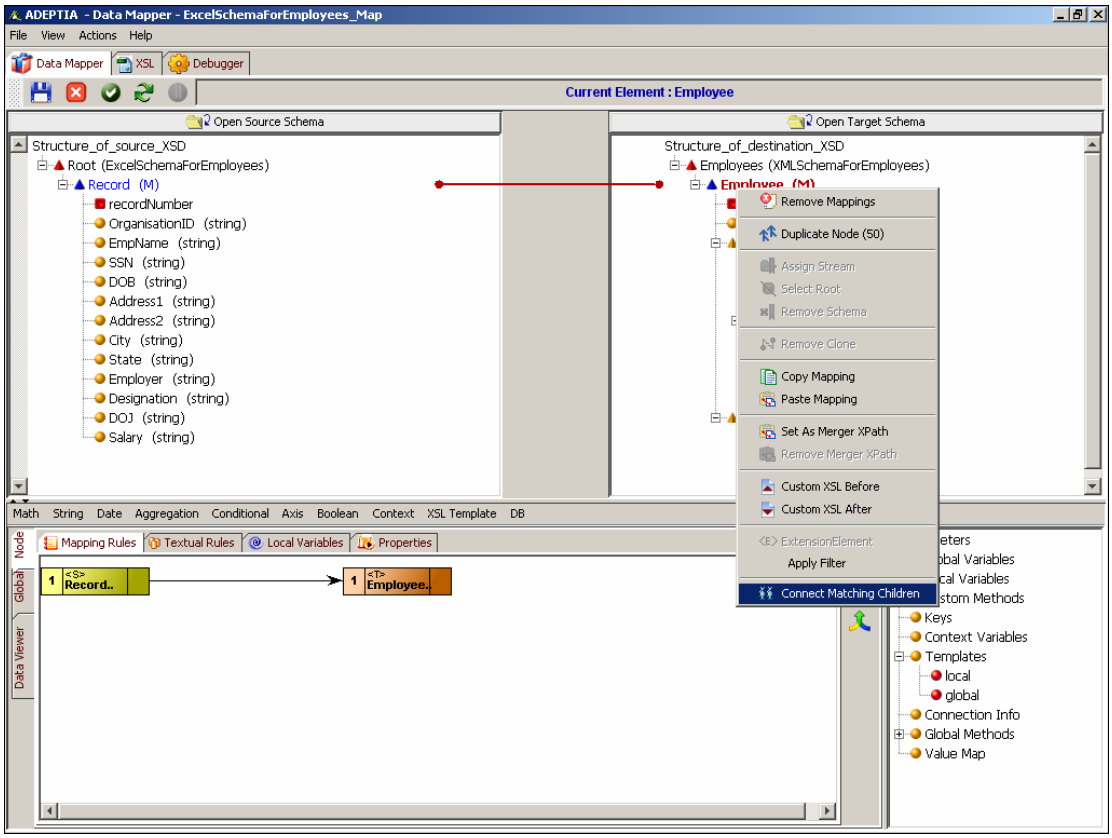


Figure 10.24: Select Connect Matching Children

12. The Connect Matching Children Dialog box is displayed (see Figure 10.25).

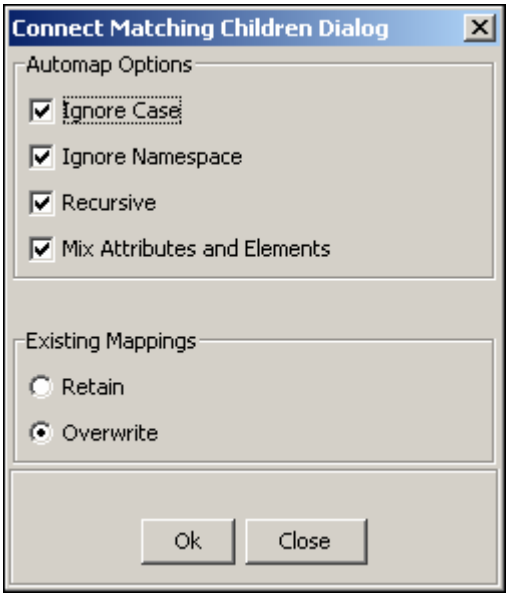


Figure 10.25: Connect Matching Children Options

13. Select the mapping criteria from the *Automap options* list. Its options are listed as:
    - **Ignore Case:** The source and the target elements are mapped even if they are of different case. Their attributes are not mapped.
    - **Ignore Namespace:** The source and target elements are mapped, when their names are same but source schema has any namespace prefix.
    - **Recursive:** Elements of parent level as well as all child level with the corresponding target elements.
    - **Mix Attributes:** Attributes of source and target schemas are mapped.

14. Select the mapping criteria from the *Existing Options* list. These options are :
    - **Retain:** If any source and target elements are already mapped, those mapping will not be overwritten.
    - **Overrite:** Overrites all the existing mappings.

15. Click **OK**. This will map source and target schema as per the criteria selected (see Figure 10.26).

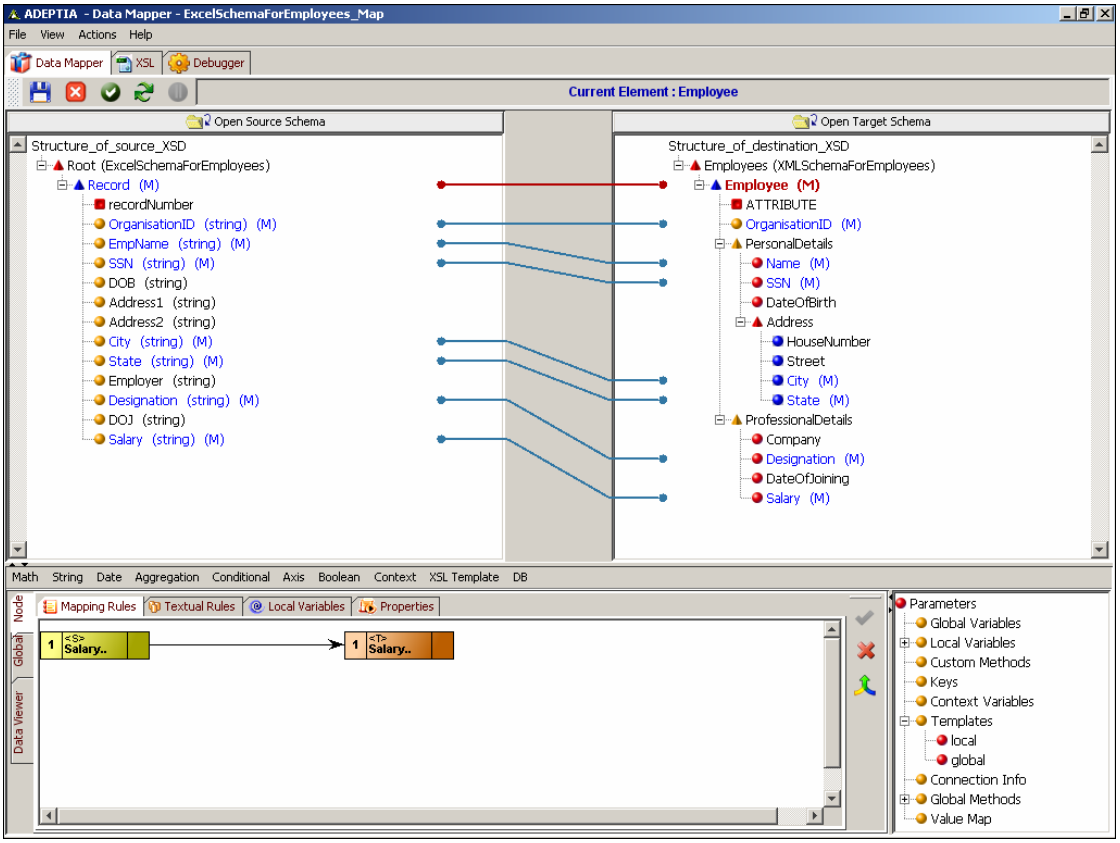


Figure 10.26: Source and Target elements are mapped

Using Add Value Map Options

You can set the Map for a particular value of an element using the Add Value Map feature. For example, if the value of an element, (for example, *EmpGender*) from source schema is coming as F and M, then you can set the map for these values as Female and Male respectively and pass it to the target element.

All the Value Map Options that you create are saved globally and can be used in any mapping activity.

Steps to use Value Map Options

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes (see Figure 10.27).

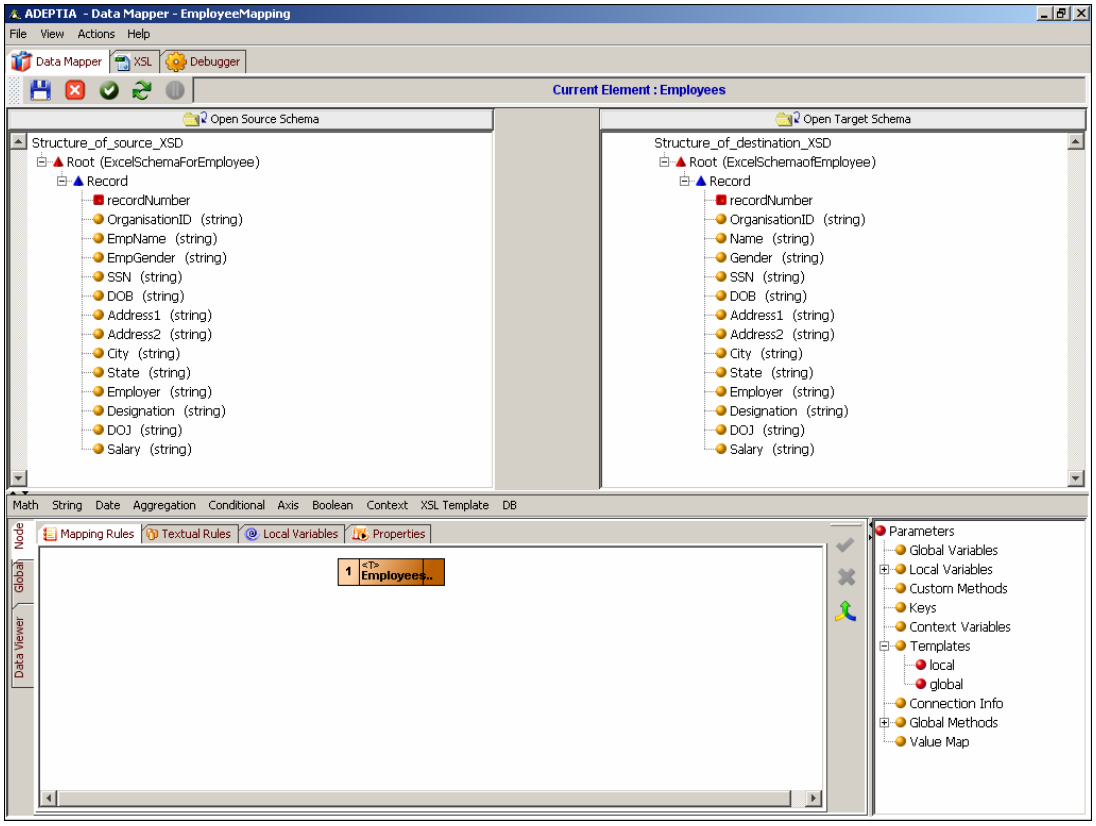


Figure 10.27: Elements of Source and Target Schema

2. Click **Actions** menu and select **Add Value Map Options** (see Figure 10.28).

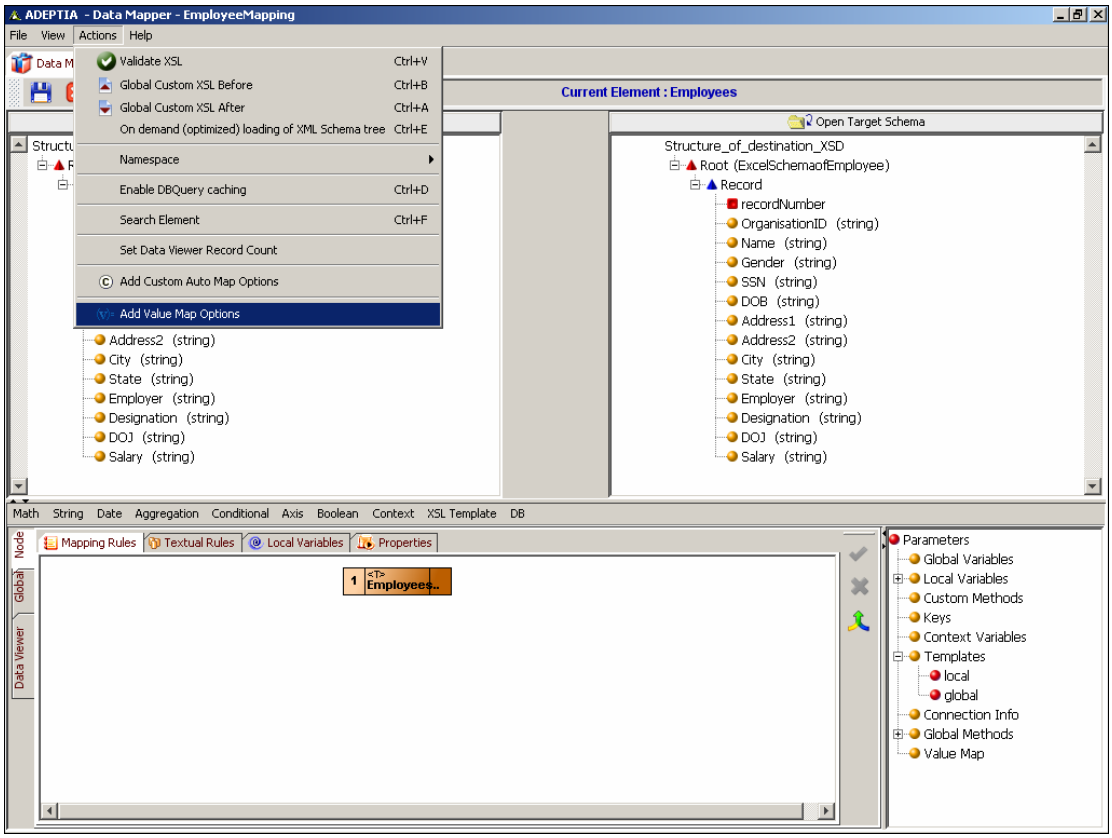


Figure 10.28: Select Add Value Map Option

3. The Value Map Options dialog box is displayed (see Figure 10.29).

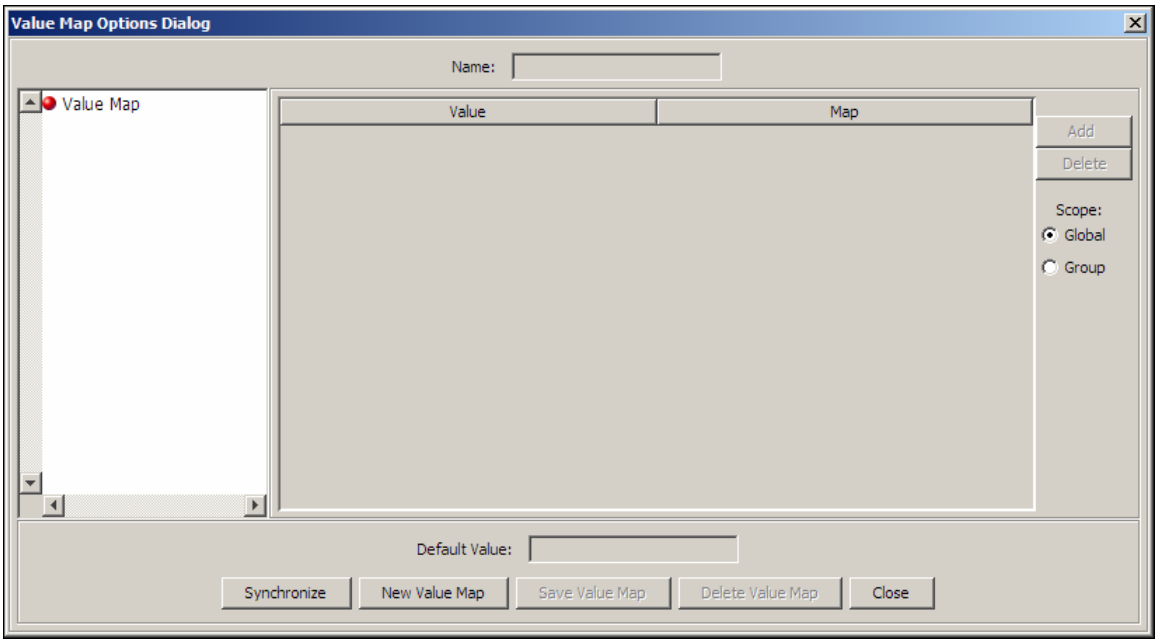


Figure 10.29: Value Map Options Dialog box

4. Click **New Value Map** button and enter the name of value map (for example, *GenderMap*) in the *Name* field.
5. Click **Add** button and enter the value (for example, *F*) in *Value* field and enter its map value(for example, *Female*) in *Map* field.
6. Similarly you can add more *Value* and *Map* fields.
7. You can provide the default Map value in the *Default Value* field (for example, *Not Provided*). Incase at source side the value is neither *F* nor *M* then *Not Provided* value is mapped set to the target element (see Figure 10.30).

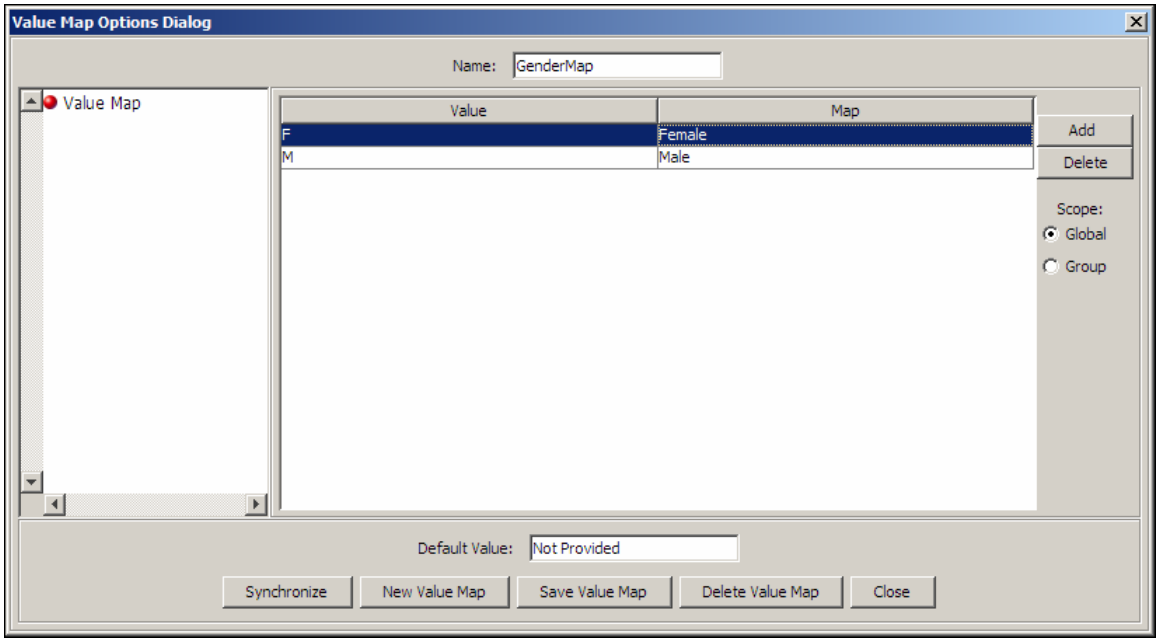


Figure 10.30: Add Value Map

8. Click the **Save Value Map** button to save the value map. This will add the newly saved value map in *Value Map* tree of *Add Value Map Options* dialog box and *Parameter Panel*. Similarly you can add more Value Map.
9. To synchronize the list of value map, click **Synchronize** button. This will list value maps created by other users.
10. Click **Close** to close the Value Map Options Dialog box.
11. Now to map it, select a target element, for example *Gender*. The selected target element is shown in Mapping Graph Area.
12. Double click on a value map you want to use, from the *Value Map* tree of Parameter Panel. For example, *GenderMap*.
13. Now double click the source element. For example, *EmpGender*.
14. Connect the output of the source element to input of value map and then connect the output of value map to the input of the target element.
15. Once you have performed the mapping activity, click **Apply Mapping** ( ) button to save the mapping.

Copy/Paste Mapping

This is an additional feature of mapping elements. You can copy the mapping associated with a target element and paste it on other target elements.

Steps to copy the mapping of a target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes. Additionally, a source and target element should be mapped.
2. Right-click the target element whose mapping you want to copy and select the **Copy Mapping** option. This copies the mapping of the selected target element.

Steps to paste the mapping on a target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes. Additionally, the mapping of a target element should be copied.
2. Right-click the target element on which you want to paste the mapping and select the **Paste Mapping** option. The Paste Mapping Options screen is displayed with a list of Paste options (see Figure 10.31).

Figure 10.31: Paste Mapping Options

3. This screen displays a list of options that can be pasted with the mapping. It includes the *For Each* and *Comments* properties and local variables.

	The **Paste Mapping** option is always enabled. The other options are enabled only if they have been defined for the copied element.

4. Select the checkbox(s) of options you want to paste with the mapping, and click **OK** button. The mapping with the selected options is pasted on the target element.

	If local variables are pasted for a target element, they are not overwritten, but are added to the existing local variables of that target element.

	You can also copy a mapping instance and paste it onto another instance. This is possible only if the schemas are same in both the mapping instances.

Mapping Functions

You can use the mapping functions built-in the Data Mapper to map source schema elements to the target schema elements. These mapping functions are listed as:

- **Math**
- **String**
- **Date**
- **Aggregation**
- **Conditional**
- **Axis**
- **Boolean**
- **Context**
- **DB**

All these mapping functions comprise of sub-functions using which you can map elements.

Steps to map elements using Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click a target element. The target element node is displayed in the Mapping Graph Area (see Figure 10.32).

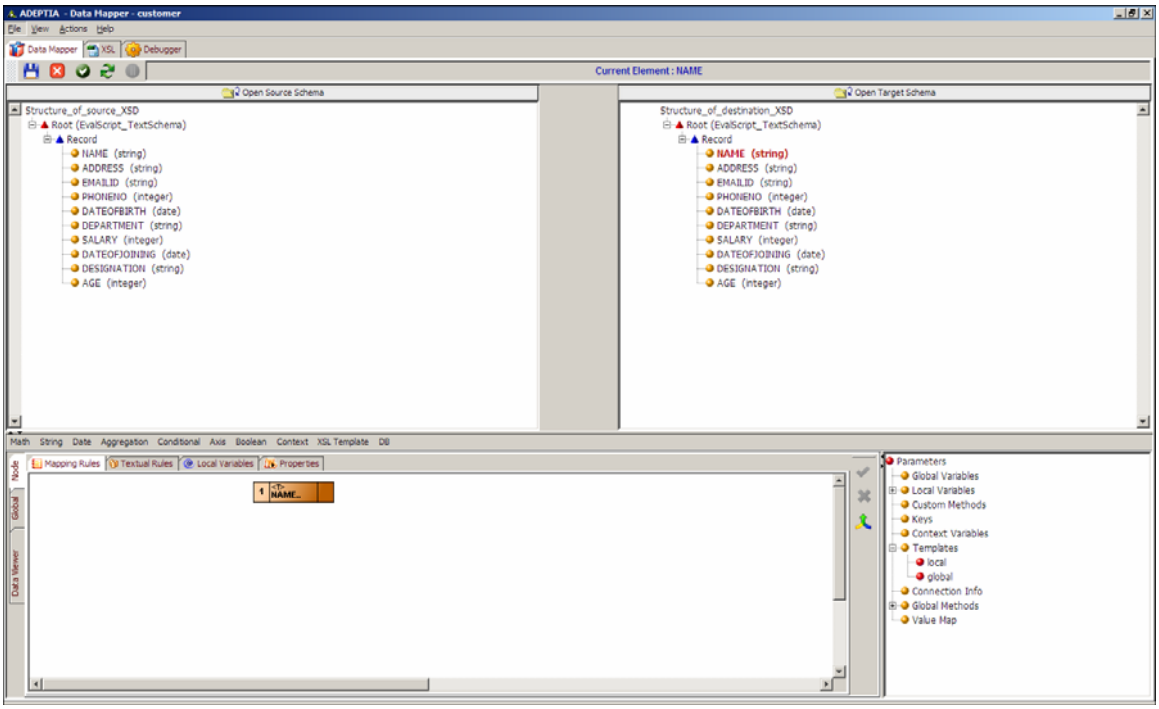


Figure 10.32: Select Target Element

3. Double-click a source element. The source element is displayed in the Mapping Graph Area (see Figure 10.33).

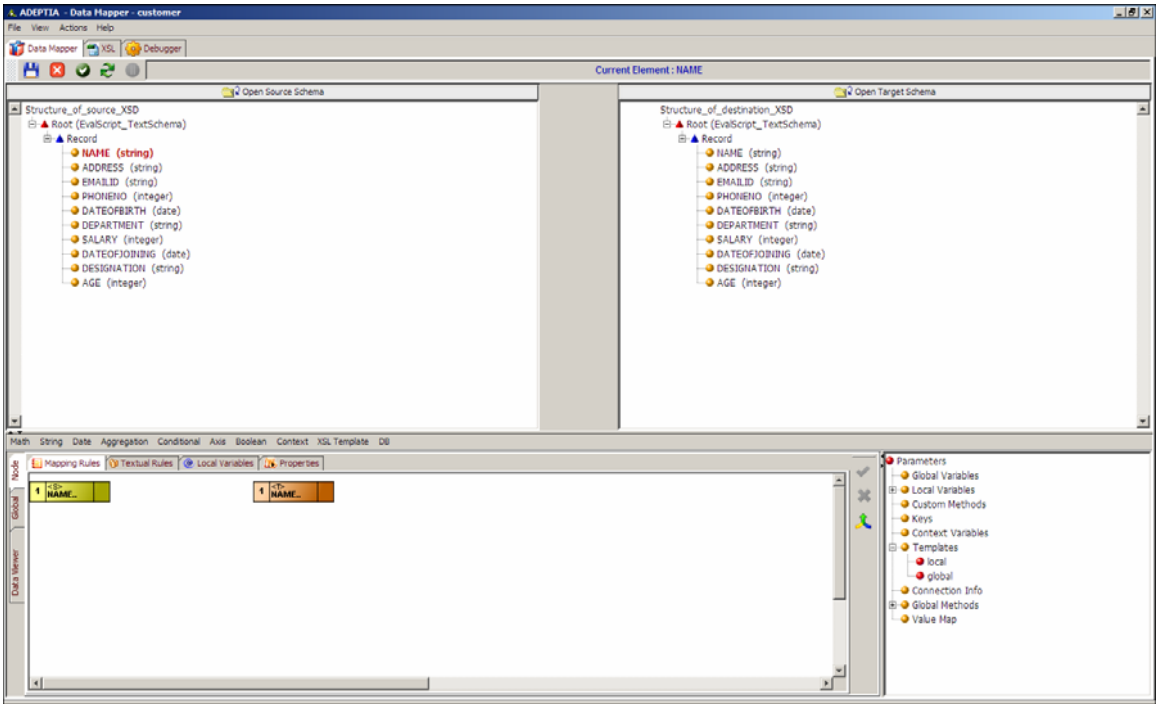


Figure 10.33: Select Source Element

	You can double-click more source elements if required. This will display the selected source elements in the Mapping Graph Area (see Figure 10.34).

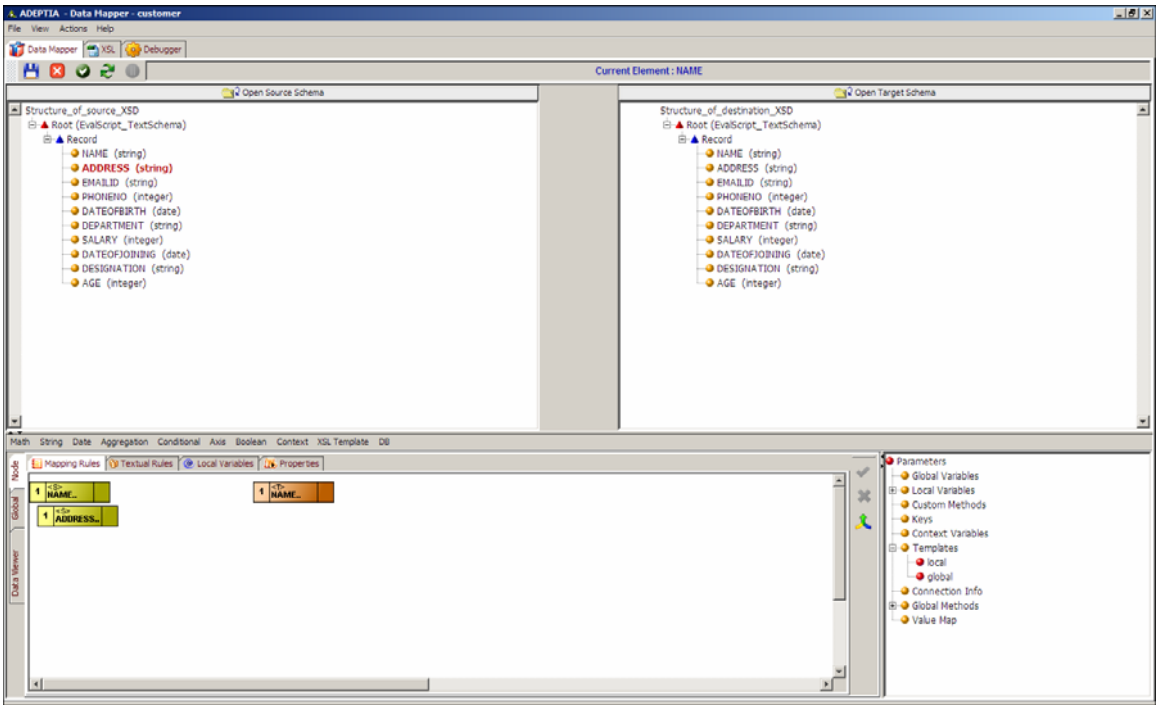


Figure 10.34: Select Second Source Element

4. Click the desired mapping function. This displays a list of sub-functions associated with the selected mapping function.
5. Select the desired sub-function and use it to map the displayed source and target elements.
6. Once you have performed the mapping activity, click **Apply Mapping** ( ) button to save the mapping.

	Once you have mapped source and target elements, you can **view and validate the generated mapping XSL**, from the Data Mapper screen. You can also **view the target XML** and **view and validate mapping output** from this screen.

For details on using these mapping functions, refer to the section **Using Mapping Functions**.

While mapping elements, certain suffixes are displayed next to source and target elements in the Source and Target Panels. These suffixes are explained in the table below.

Table 10.8: Suffixes of Source and Target Elements

Suffix	Description

(C)	Target element can be cloned to create another target element. Only a parent element can be cloned in case of a text schema.
(CM)	Comments have been added for the target element.
(F)	For Each property has been applied for the target element.
(LV)	A local variable has been declared to the target element.
(M)	Element has been mapped.
(SR)	Target Element has been sorted.
(S)	Source Element has been set as Splitter XPath. Target element has been set as Merger XPath.
(FL)	Filter has been applied on the target element.

More than one suffix can be displayed next to a source and target element.

Save Mapping and Exit Data Mapper

Once you have mapped source and target elements, you can save the mapping and exit the Data Mapper.

Steps to save the mapping and exit the Data Mapper

1. Click **File** menu and select **Save** to save the mapping. Alternately, you can click the **Save** () button on the toolbar. The server first validates the mapping activity. If successful, it displays a dialog box confirming that the mapping has been saved successfully. If the C*omments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to the mapping. (see Figure 10.35).

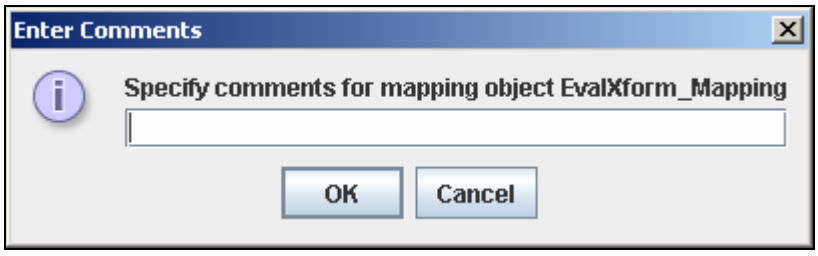


Figure 10.35: Add Comments (Mapping)

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

	The comment should be at least 1 character in length. If you enable/disable the *Comments* property in the middle of a mapping activity, you need to restart the mapping applet.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

4. Click **File** menu and select **Exit** to close the Data Mapper applet. A confirmation dialog box is displayed (see Figure 10.36).

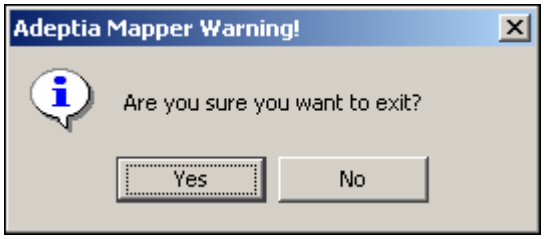


Figure 10.36: Exit from Data Mapper

5. Click **Yes** button to exit the Data Mapper screen and return to the Manage Data Mapping screen. If the mapping object has not been saved, then the Save Mapping Object dialog box is displayed (see Figure 10.37).

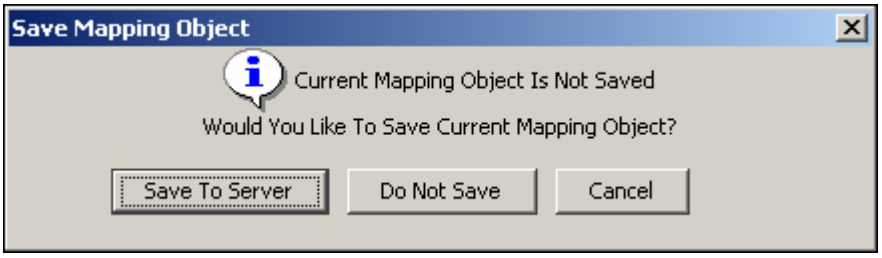


Figure 10.37: Save Mapping Object

6. Click **Save To Server** button to save the mapping object to the Adeptia Suite. A screen is displayed where you need to enter comments related to the mapping. (refer to Figure 10.35).
7. Enter the comments in *Specify comments for mapping object (object name)* field.
8. Click **OK** to save the comments. This displays a confirmation dialog box stating that the mapping has been saved successfully. The control returns to the Manage Data Mapping screen, where a list of mapping activities is displayed.
9. Refresh the Internet browser to view the saved mapping activity.

	You can **view and validate the generated mapping XSL**, **view the mapping XML** and **view and validate mapping output**, before saving the mapping activity.

	At times, on saving a mapping activity, the memory usage may exceed its actual capacity depending on the JVM and the operating system.

View and Validate Generated Mapping XSL

This is a very useful feature of the Data Mapper. It enables you to validate and test the generated XSL before saving the mapping activity.

Steps to view and validate the generated mapping XSL

1. Click **XSL** tab on the Tabs Panel to view the generated mapping XSL. The generated XSL code with line numbers is displayed (see Figure 10.38).

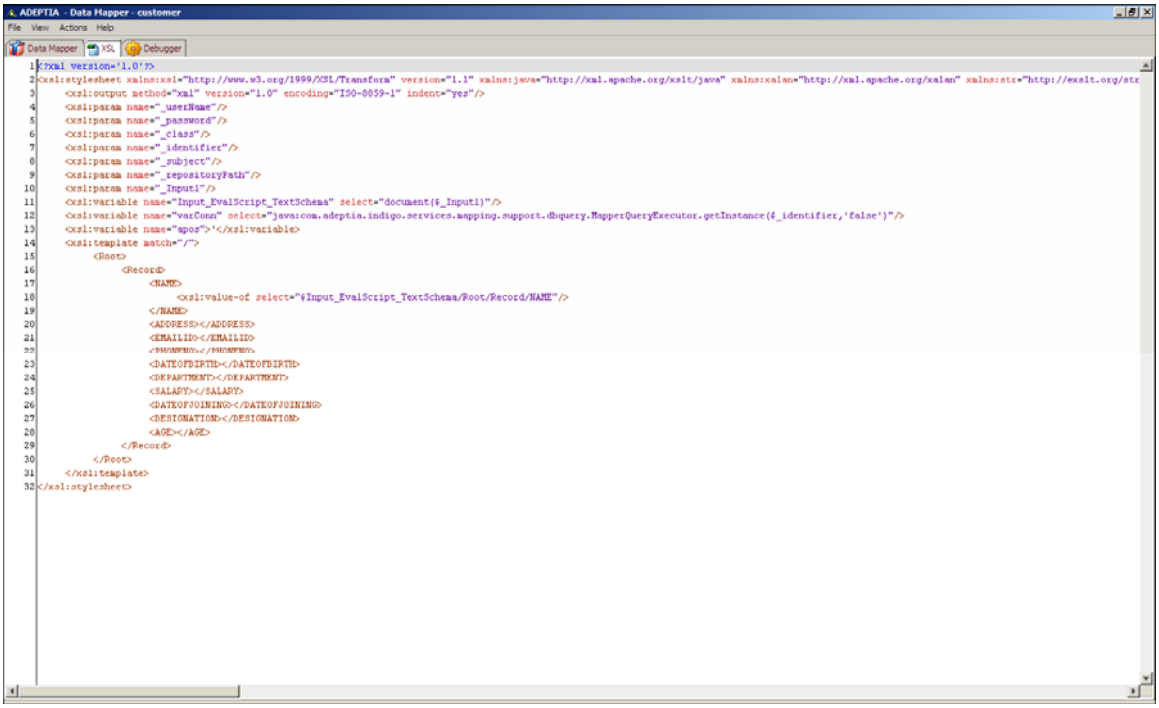


Figure 10.38: Mapping XSL

2. Click the **Action** menu and select **Validate XSL** option to validate the generated XSL. A dialog box is displayed confirming that the XSL is valid.

	If generated XSL is invalid, then an error is displayed as a selected line. Error details are displayed in the XSL Errors Pane. It displays one error at a time.

3. Click **OK** button to close the above dialog box.

	You can also validate the generated XSL from the Create Data Mapping screen. For more details, refer to the section **Splitting Source Data.**

View Mapping XML

The Data Mapper allows you to view the generated XML code, before saving the mapping activity.

Steps to view Mapping XML

1. Click the **View** menu and select **View Mapping XML** option. This displays the generated XML code with line numbers, in read-only mode (see Figure 10.39).

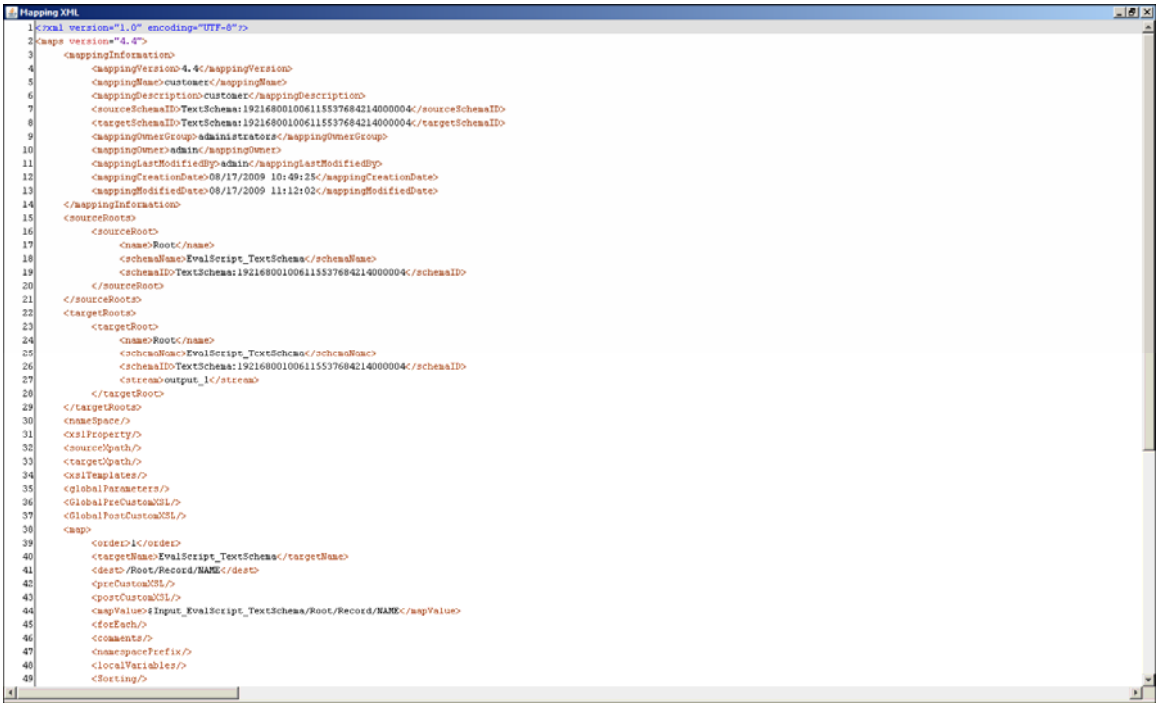


Figure 10.39: View Mapping XML

View and Validate Mapping Output

Once you have mapped source and target schema elements, you can view and validate the output of the mapping activity. This feature enables you to test the data and make modifications to the mapping activity, before saving it.

Steps to view and validate mapping output

1. Click **Debugger** tab on the tabs panel to view and validate the output of the mapping activity. The Debugger is displayed on the Data Mapper screen (see Figure 10.40).

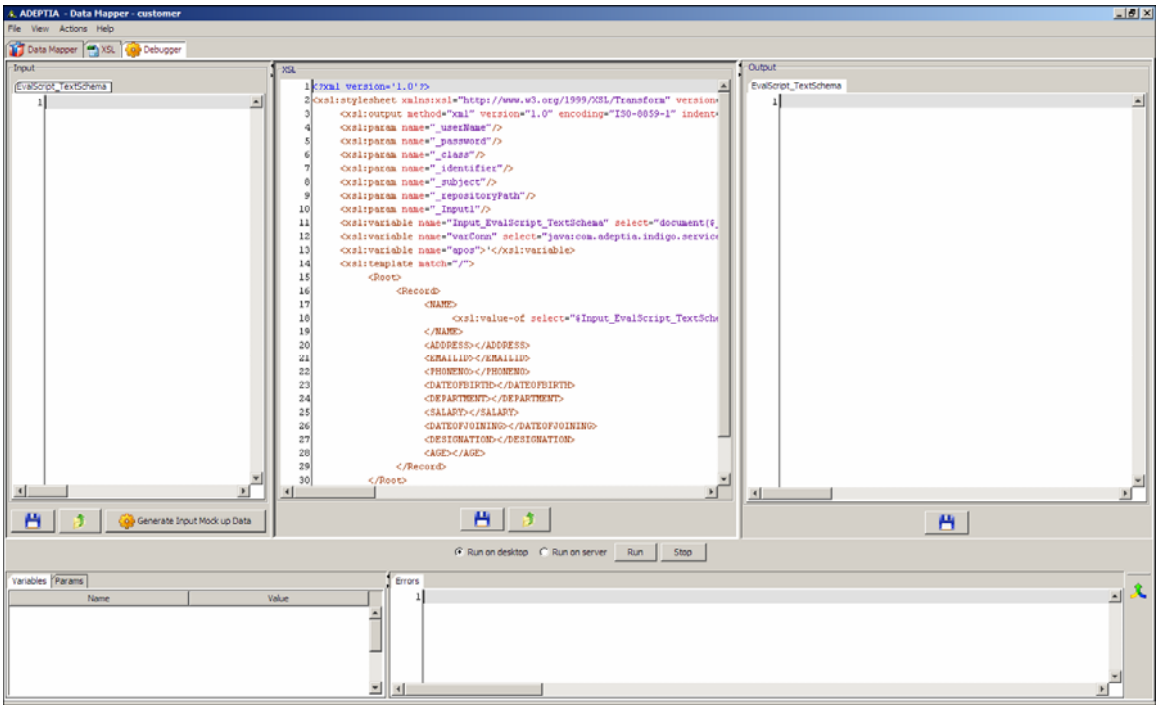


Figure 10.40: Debugger screen

This screen is divided into five sections as listed in the table below.

Table 10.9 : Options of the Debugger Screen

Section	Description
Input	Displays data for all input schemas in XML format. This data can be mockup data or a selected XML file.
XSL	Displays the generated XSL. You can generate sample output data based on the input data.
Output	Displays the generated output data for all schemas.
Variables/ Params	This section comprises of two tabs: Variables Params The Variables tab displays the local variables that are encountered in the generated XSL. The Params tab displays all parameters that are defined for a XSL template. These parameters are displayed only if that XSL template is used in the mapping activity.
Errors	Displays all errors encountered during debugging.

2. Click **Generate Input Mock up Data** button displayed in the *Input* section, to generate the sample input data in XML format. The sample input data contains name of the element as XML Tags and some randomly generated values as their data. The sample input XML file is displayed in the *Input* section (see Figure 10.41).

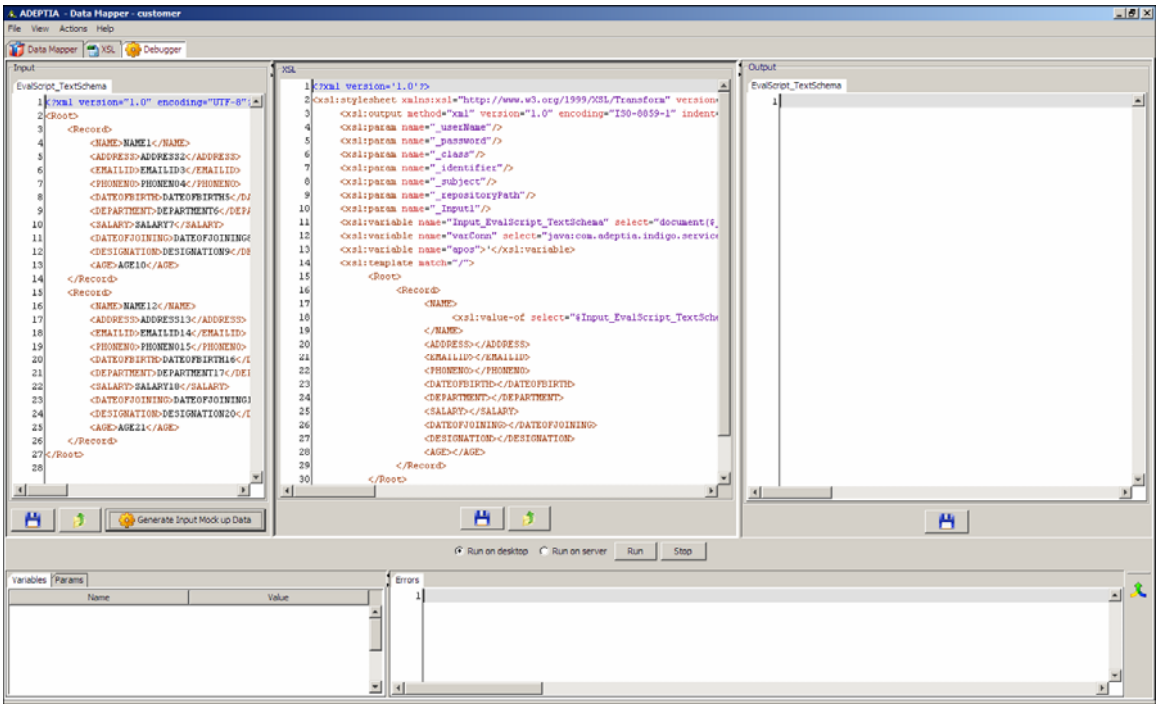


Figure 10.41: Generate Mockup Data

	Data will be generated for all input schemas.

3. Alternately, you can attach a source file to the root element of the source schema. Right-click the root element of the source schema and select **Attach Source** from the drop-down list. Select the source file to attach it. This automatically converts the actual data of the source file into XML input data, which can be validated.

4. Alternately, you can display input data from an existing XML file. Click **Load Input XML** button ( ) displayed in the Input section to select and upload the existing XML file.

	Another way of uploading the XML file is right-clicking the *Input* section and clicking the **Load** option. If you load an XML file, then the existing XML code is overwritten. It is not appended.

5. Click **Save** button ( ) displayed in the Input section to save the generated input data.

	You can simulate the mapping either on the desktop or on the server. By default, simulation is done on the desktop. Simulating on the server supports global functions, custom methods, Context Variable, Set Context, Get Context

	and the DBlookup () functions. However, it can be used only in case of mapping of single source and target elements. While simulating on the server using context variables, if you assign a value to the context variable using the Set Context function and retrieve its value using the Get Context function, it will return the default value of the context variable, instead of the new value assigned. For example, a context variable 'Var1' has the default value as 10 and is later assigned the value of 20 by Set Context function. When the Get Context function is used to retrieve the value of the variable, it returns 10 instead of 20.

6. Click the *Run on Server* radio button if you want to simulate the mapping activity on the server.

7. Click **Run** button displayed in the XSL section to generate sample output data based on input data. The generated sample output data is displayed in the Output section (see Figure 10.42).

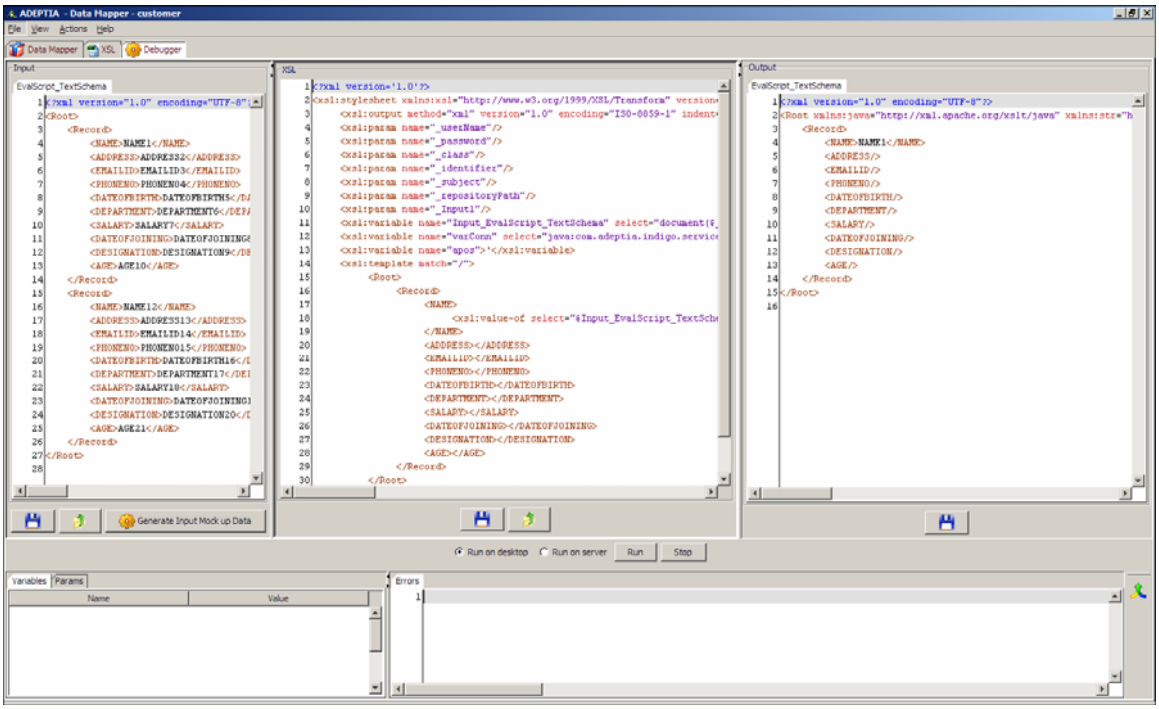


Figure 10.42: Output XSL

	You can stop the generation of output data at any time by clicking the **Stop** button displayed in the XSL section. In case XML Schema is used at target end, then unmapped elements are filtered out automatically.

8. Click **Save** button () displayed in the Output section to save the generated mapping output.
9. A dialog box is displayed using which you can save the XML file at the desired location.

Using Data Viewer

Data Viewer is an advanced feature of Data Mapper which is used to view actual input and output data after applying mapping rules. In data viewer, you can upload actual input file and view the actual output. It also displays error records (if any). Additionally, it also validates the enumerated values and displays the list of enumerated literals for the target element incase XML Schema is used at target.

Data Viewer is helpful if you want to know what will be the output of the mapper. In this case you need not to execute the process flow. You can simply upload the source file and view input and output record.

In data viewer, records are displayed in Grid View as well as Tree View. Table 10.10 lists the schemas, whose records can be displayed in data viewer.

Table 10.10: List of schema whose records can be shown in Data Viewer

Schema	Tree View	Grid View
Adv. Text Schema	√	X
Excel Schema	√	√
Hierarchical Excel Schema	√	X
Text Schema	√	√
XML Schema	√	X
Adv. Positional Schema	√	X
Positional Schema	√	√

Steps to use Data Viewer

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.

2. Map source and target elements and apply the mapping rule as per your need.

3. To use Data Viewer**,** click **Data Viewer** tab. The **Data Viewer** panel is displayed (see Figure 10.43).

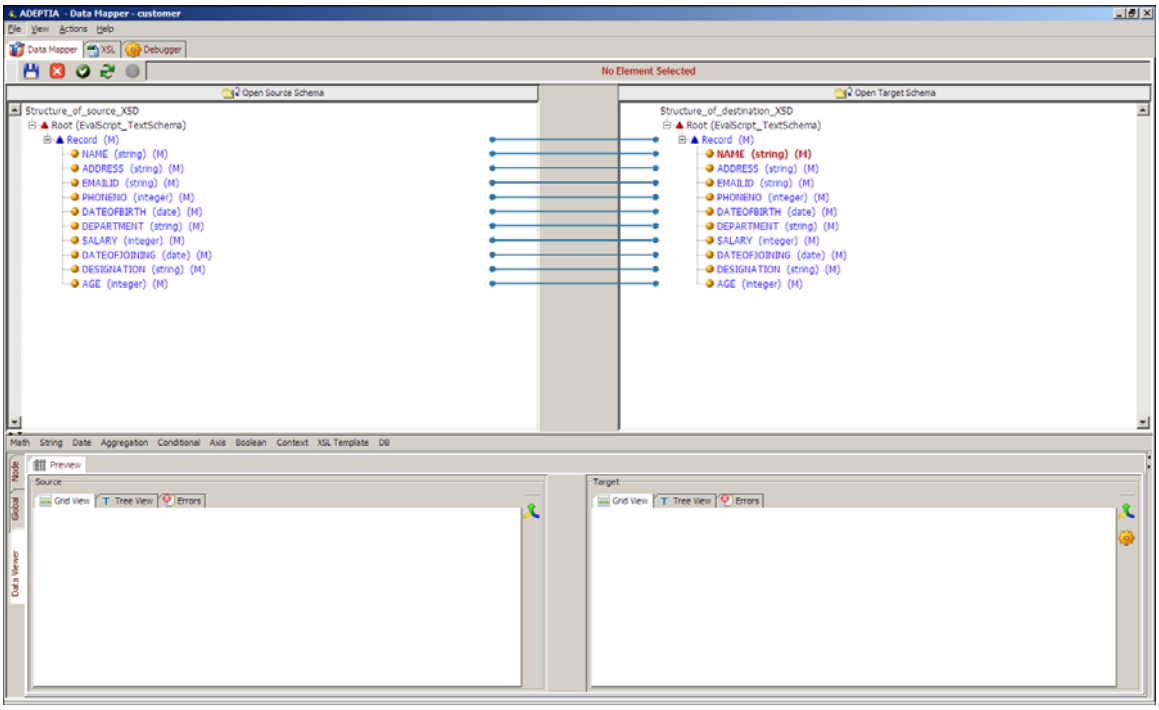


Figure 10.43: Data Viewer

4. To attach the source file, right click the **Root** element in the Source Panel and select **Attach Source** option and select the source file (see Figure 10.44).

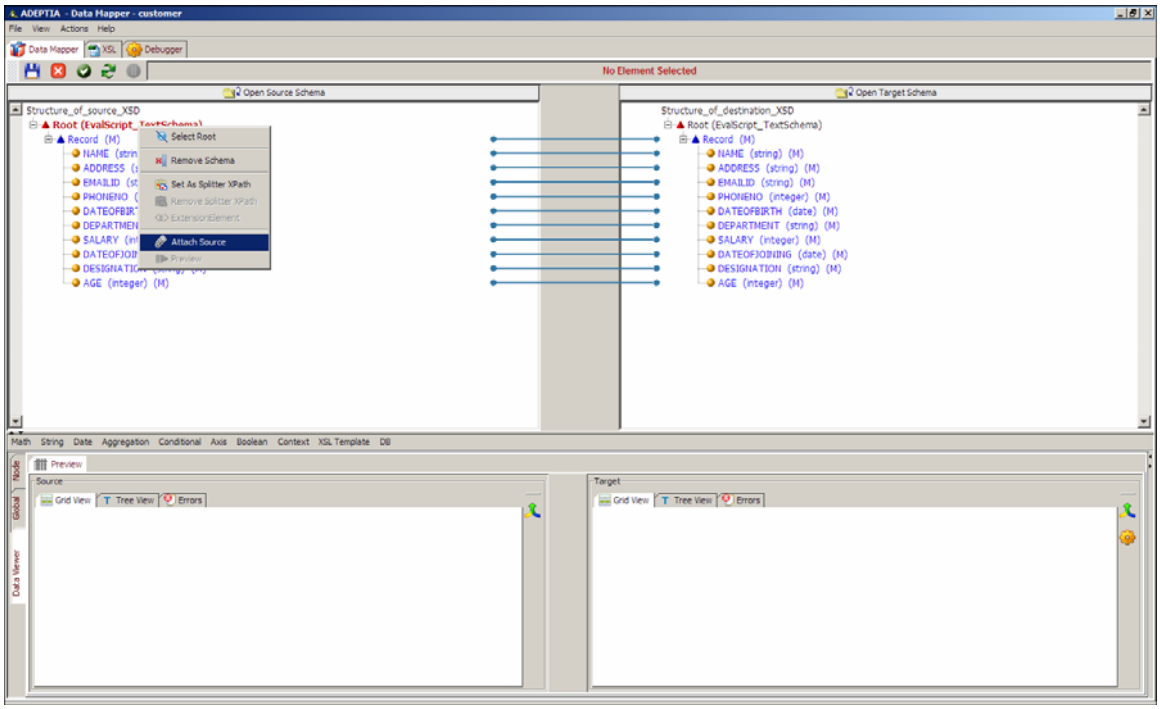


Figure 10.44: Attach Source

5. After selecting the source file, again right click the **Root** element in source panel and select **Preview**. Source Data is displayed in the *Grid View* panel (see Figure 10.45).

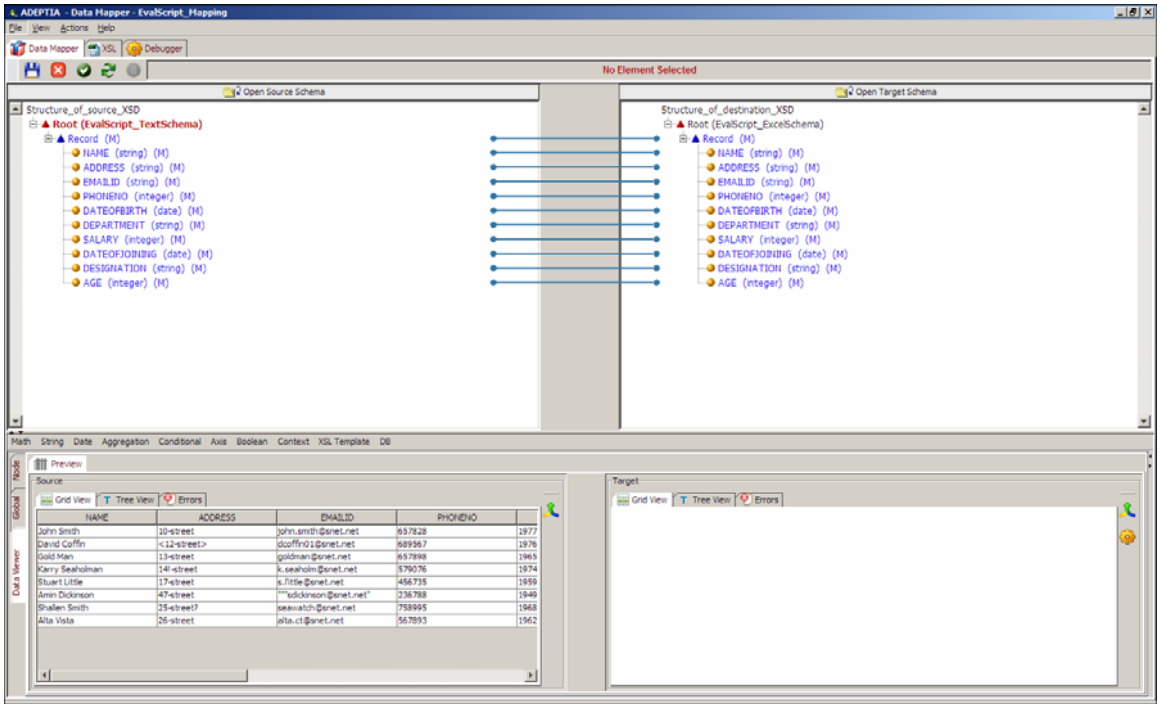


Figure 10.45: Source records in Grid View

	By default maximum 50 records are displayed in the Data Viewer. To view more records, click **Action** menu and select **Set Data Viewer Record Count** and enter the desired value. When you change this value, you need re-attach the source.

6. If there is any error record in the source file, a pop-up message is displayed that *"Error Records found in the source data."* (see Figure 10.46).

Figure 10.46: Pop-up message

7. Click **OK** to close this pop-up message.

8. To view the source data in *Tree View*, click the **Tree View** tab. The source data is selected in the tree view (see Figure 10.47).

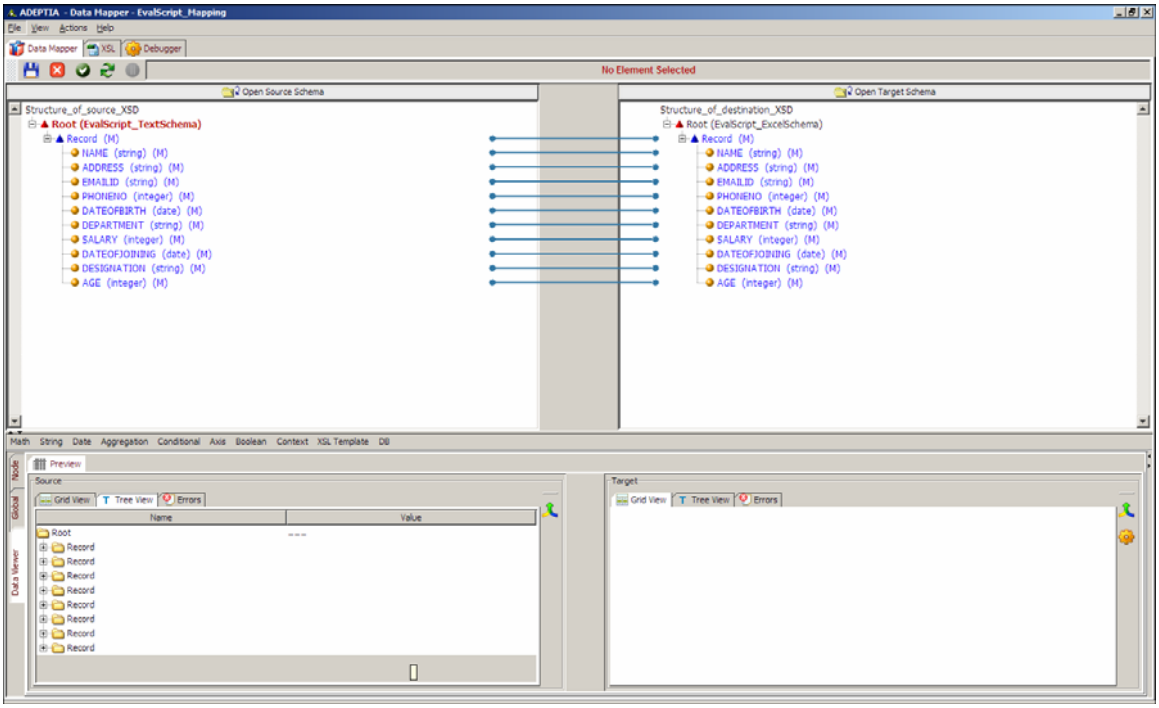


Figure 10.47: Tree View

9. To view the values of the record, click **[+]** to expand the *Record.* Values of the expanded record are shown (see Figure 10.48).

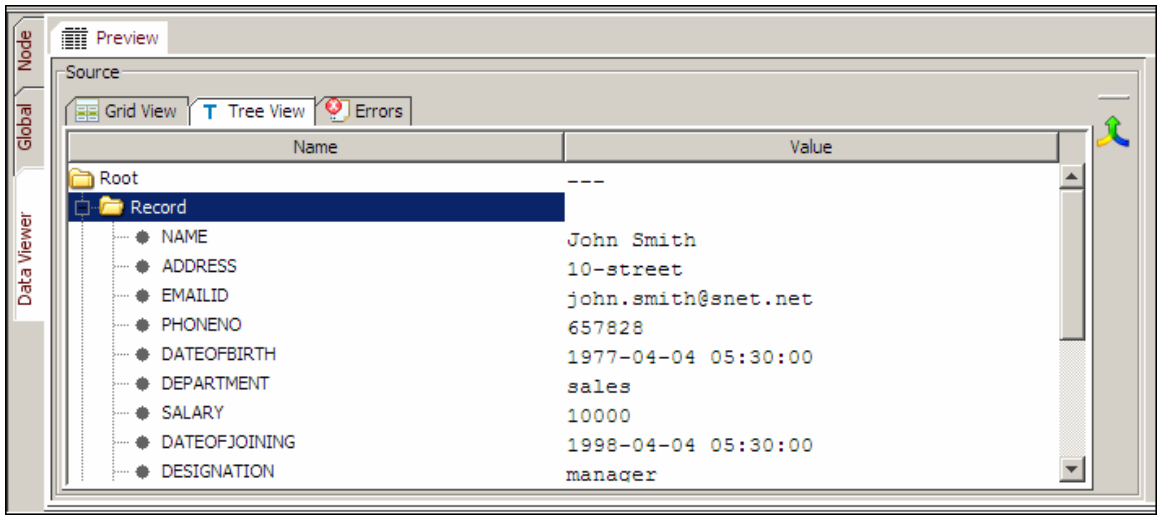


Figure 10.48: Records in Tree View

10. Similarly to view values of other records, expand other records.

11. If there is any error encountered in the source data, those error records are shown in the *Errors* tab. To view the error records, click **Errors** tab (see Figure 10.49).

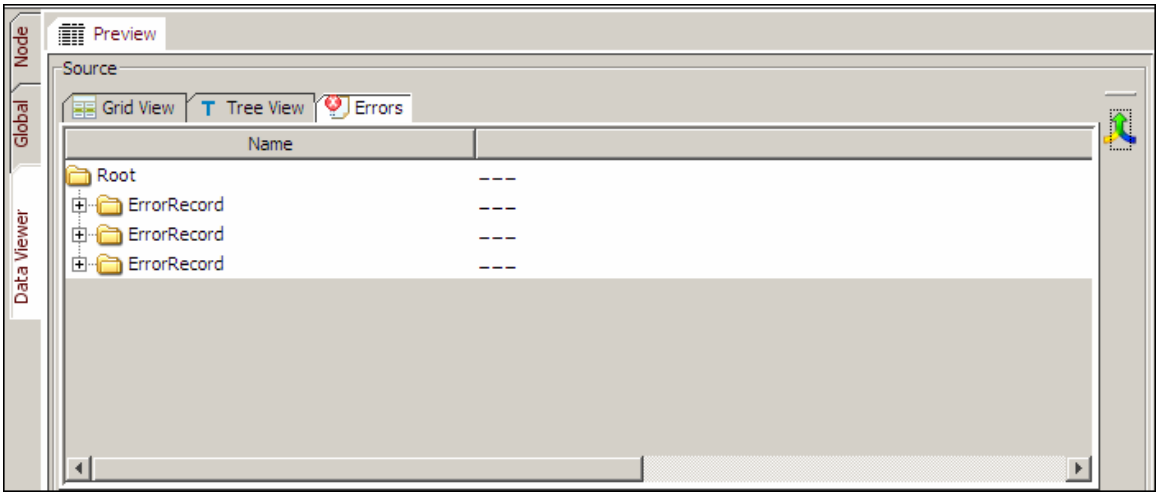


Figure 10.49: Error Record

12. To view the output record, right click the **Root** element of the target panel and select **Preview** option. The generated output records are displayed in the *Target* panel of *Preview* pane (see Figure 10.50).

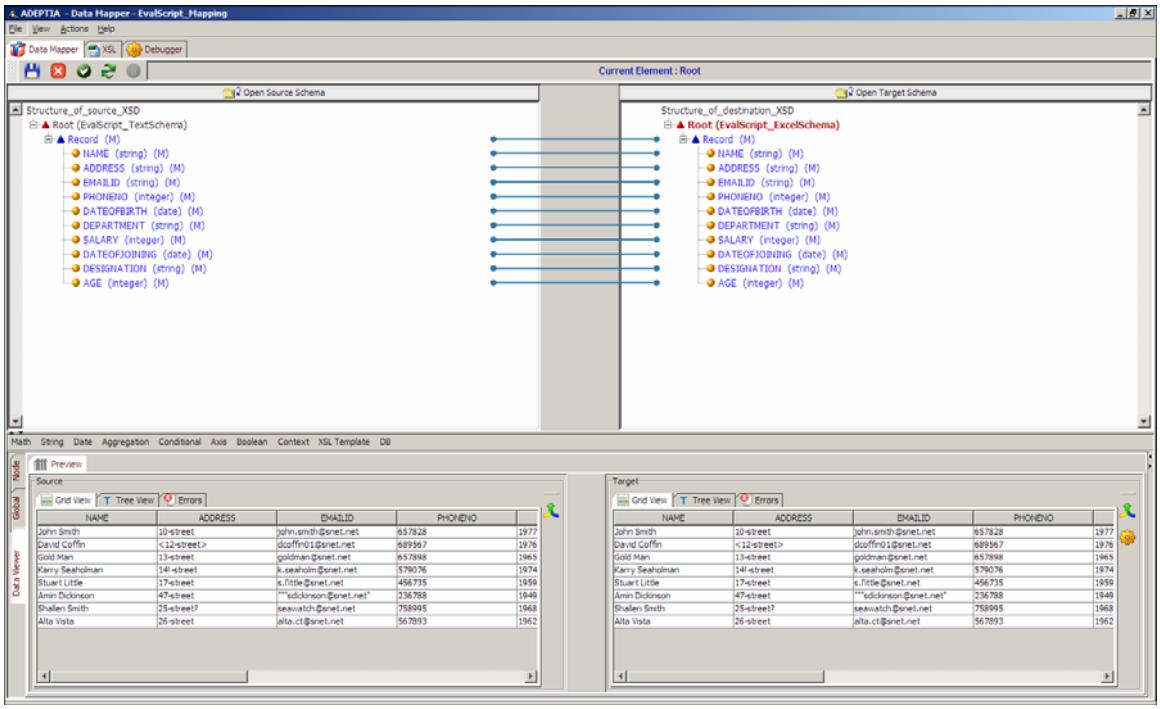
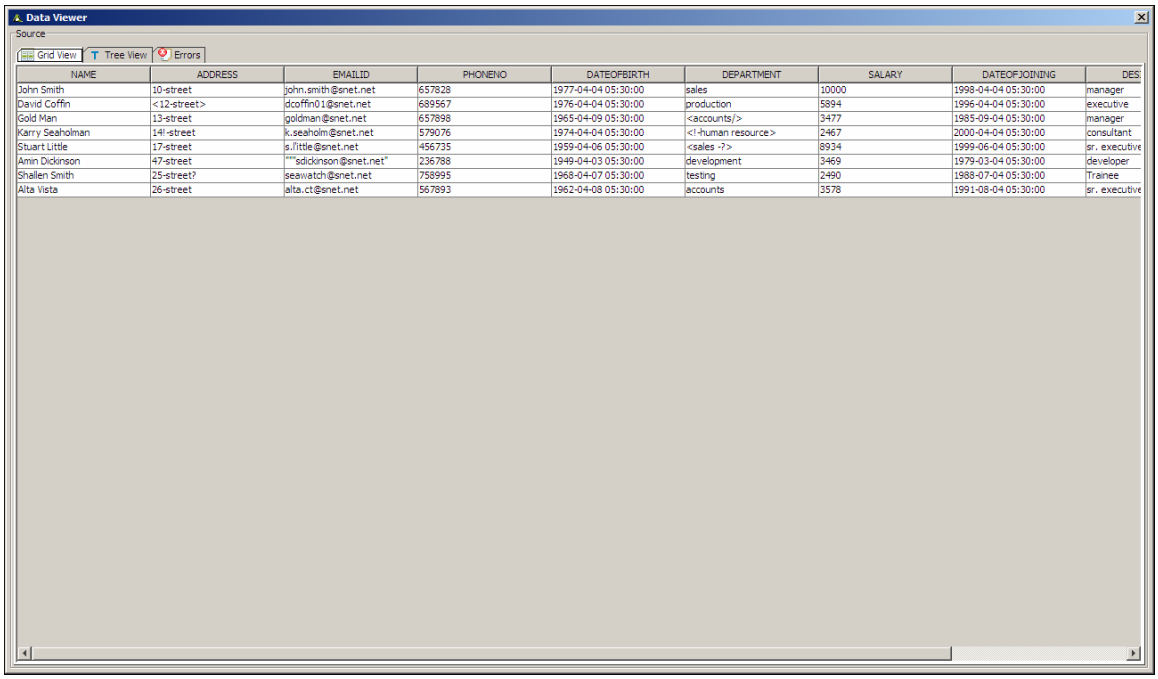


Figure 10.50: Target Records in Grid View

	In case XML Schema is used at target end, then unmapped elements are filtered out automatically.

13. To view the error records at the target end, click **Errors** tab of target panel of Data Viewer. It displays:
    - The number of errors occurred during transformation
    - The list of errors occurred during transformation
    - List of enumerated literals for the target element if invalid data is mapped to the target element which is restricted to the enumerated values
14. In case you perform any changes in mapping rules, you need to click  to refresh the target data according to mapping rules applied.
15. You can also view the output data in tree view, by clicking the *Tree View* tab.
16. To expand the **Source** or **Target** panel, click  . The **Source** or **Target** panel is shown in the expanded (see Figure 10.51).



NAME	ADDRESS	EMAILID	PHONENO	DATEOFBIRTH	DEPARTMENT	SALARY	DATEOFJOINING	DES
John Smith	10-street	john.smith@snet.net	657828	1977-04-04 05:30:00	sales	10000	1998-04-04 05:30:00	manager
David Coffin	<12-street>	dcoffin01@snet.net	689567	1976-04-04 05:30:00	production	5894	1996-04-04 05:30:00	executive
Gold Man	13-street	goldman@snet.net	657898	1965-04-09 05:30:00	<accounts/>	3477	1985-09-04 05:30:00	manager
Karry Seaholman	14!-street	k.seaholm@snet.net	579076	1974-04-04 05:30:00	<!-human resource>	2467	2000-04-04 05:30:00	consultant
Stuart Little	17-street	s.little@snet.net	456735	1959-04-06 05:30:00	<sales -?>	8934	1999-06-04 05:30:00	sr. executive
Amin Dickinson	47-street	"""sdickinson@snet.net"	236788	1949-04-03 05:30:00	development	3469	1979-03-04 05:30:00	developer
Shallen Smith	25-street?	seawatch@snet.net	758995	1968-04-07 05:30:00	testing	2490	1988-07-04 05:30:00	Trainee
Alta Vista	26-street	alta.ct@snet.net	567893	1962-04-08 05:30:00	accounts	3578	1991-08-04 05:30:00	sr. executive

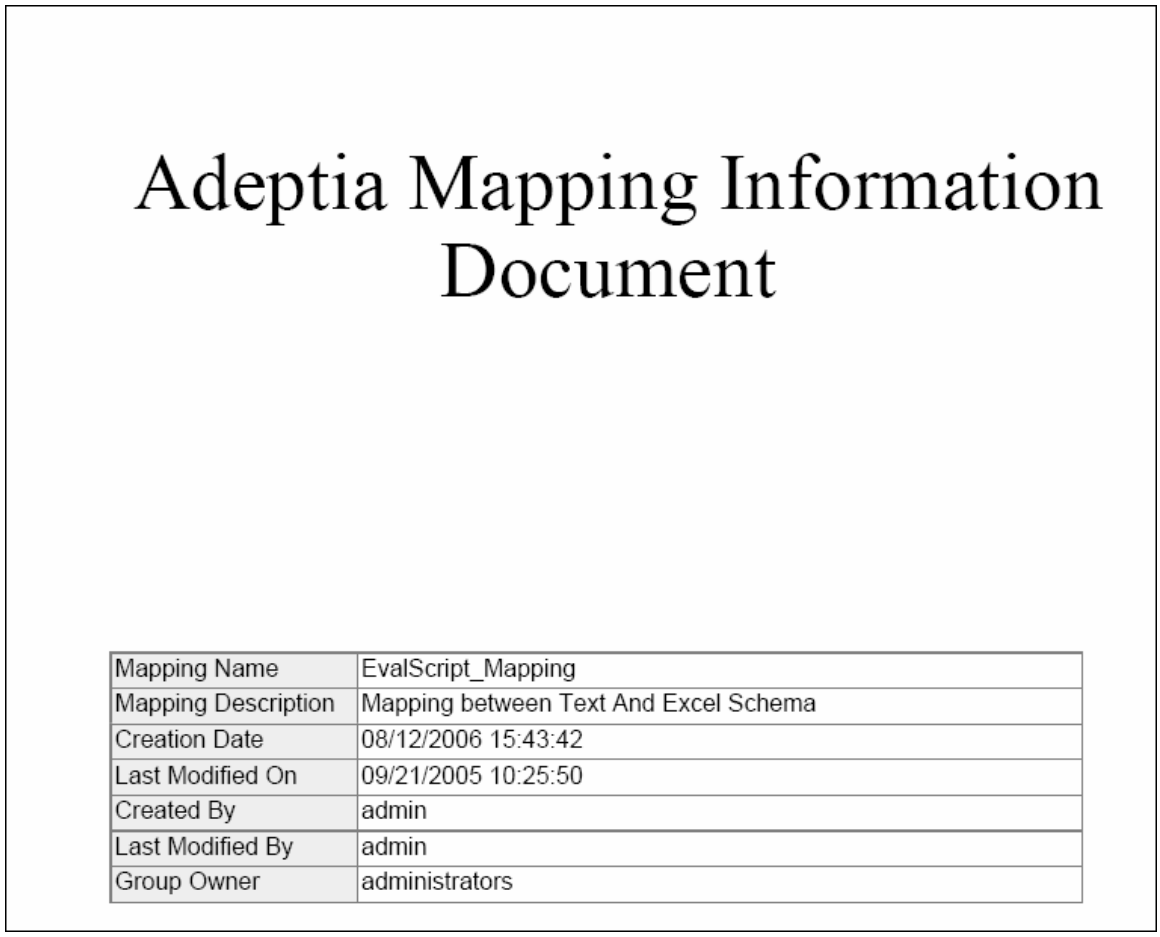
Figure 10.51: Target Panel in expanded mode

17. To close the expanded **Source** or **Target** panel, click  .

Viewing Mapping in PDF format

Steps to view mapping activity in PDF format

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed (refer to Figure 10.1).
4. Click the radio button against the mapping activity whose PDF details you want to view. This selects the mapping activity and activates the *PDF View* link. Clicking this link displays a PDF file with all information associated with the mapping activity.
5. This file displays divides the mapping information into various pages. The first page contains mapping information such as mapping name, description; creation details and the group owner (see Figure 10.52).

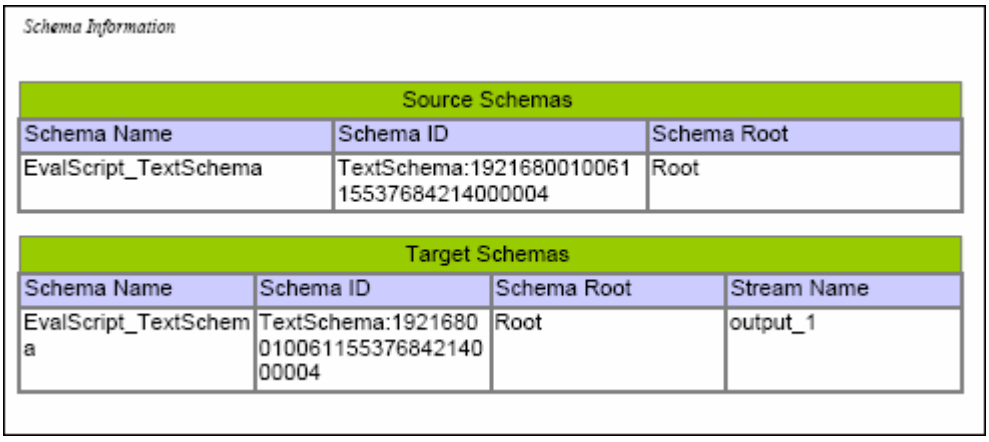


Adeptia Mapping Information Document

Mapping Name	EvalScript_Mapping
Mapping Description	Mapping between Text And Excel Schema
Creation Date	08/12/2006 15:43:42
Last Modified On	09/21/2005 10:25:50
Created By	admin
Last Modified By	admin
Group Owner	administrators

Figure 10.52: Mapping Information Document

6. The next page displays all source and target schema information (see Figure 10.53).



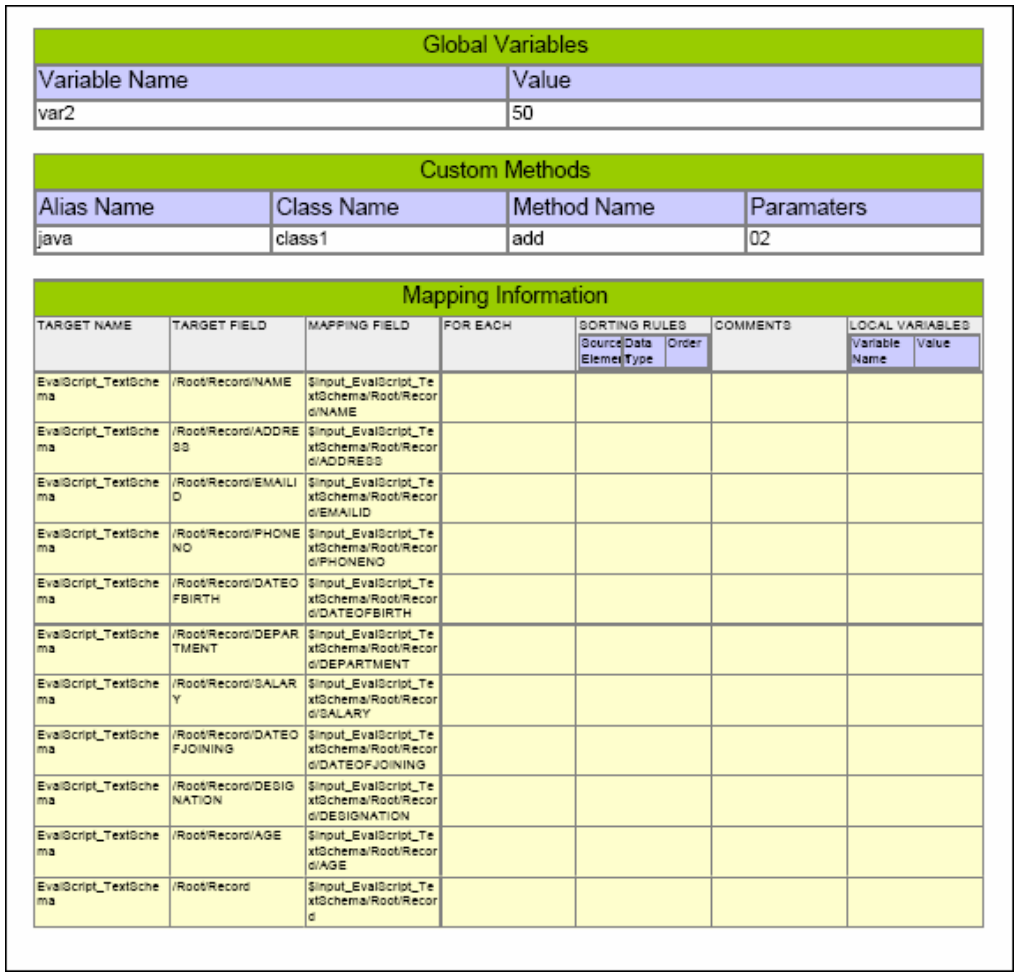
Schema Information

Source Schemas		
Schema Name	Schema ID	Schema Root
EvalScript_TextSchema	TextSchema:1921680010061 15537684214000004	Root

Target Schemas			
Schema Name	Schema ID	Schema Root	Stream Name
EvalScript_TextSchema	TextSchema:1921680 010061155376842140 00004	Root	output_1

Figure 10.53: Schema Information

7. The next page displays all other mapping information such as XSL templates used, variables defined, properties, keys, and sorting rules (see Figure 10.54).



Global Variables

Variable Name	Value
var2	50

Custom Methods

Alias Name	Class Name	Method Name	Paramaters
java	class1	add	02

Mapping Information

TARGET NAME	TARGET FIELD	MAPPING FIELD	FOR EACH	SORTING RULES			COMMENTS	LOCAL VARIABLES	
				Source Elemer	Data Type	Order		Variable Name	Value
EvalScript_TextSchema	/Root/Record/NAME	$input_EvalScript_TextSchema/Root/Record/NAME							
EvalScript_TextSchema	/Root/Record/ADDRESS	$input_EvalScript_TextSchema/Root/Record/ADDRESS							
EvalScript_TextSchema	/Root/Record/EMAILID	$input_EvalScript_TextSchema/Root/Record/EMAILID							
EvalScript_TextSchema	/Root/Record/PHONENO	$input_EvalScript_TextSchema/Root/Record/PHONENO							
EvalScript_TextSchema	/Root/Record/DATEOFBIRTH	$input_EvalScript_TextSchema/Root/Record/DATEOFBIRTH							
EvalScript_TextSchema	/Root/Record/DEPARTMENT	$input_EvalScript_TextSchema/Root/Record/DEPARTMENT							
EvalScript_TextSchema	/Root/Record/SALARY	$input_EvalScript_TextSchema/Root/Record/SALARY							
EvalScript_TextSchema	/Root/Record/DATEOFJOINING	$input_EvalScript_TextSchema/Root/Record/DATEOFJOINING							
EvalScript_TextSchema	/Root/Record/DESIGNATION	$input_EvalScript_TextSchema/Root/Record/DESIGNATION							
EvalScript_TextSchema	/Root/Record/AGE	$input_EvalScript_TextSchema/Root/Record/AGE							
EvalScript_TextSchema	/Root/Record	$input_EvalScript_TextSchema/Root/Record							

Figure 10.54: Detailed Mapping Information

View Mapping in Read-Only Mode

If you have read-only rights, you can still view a mapping activity. You can view the applied mapping, create new mapping rules, edit existing mapping rules and run simulation. However, you cannot save the mapping activity.

Steps to view mapping in read-only mode

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Data Mapping**. The Manage Data Mapping screen is displayed (refer to Figure 10.1).

4. Click the mapping activity that you want to view in read-only mode. The View Data Mapping screen is displayed (see Figure 10.55).

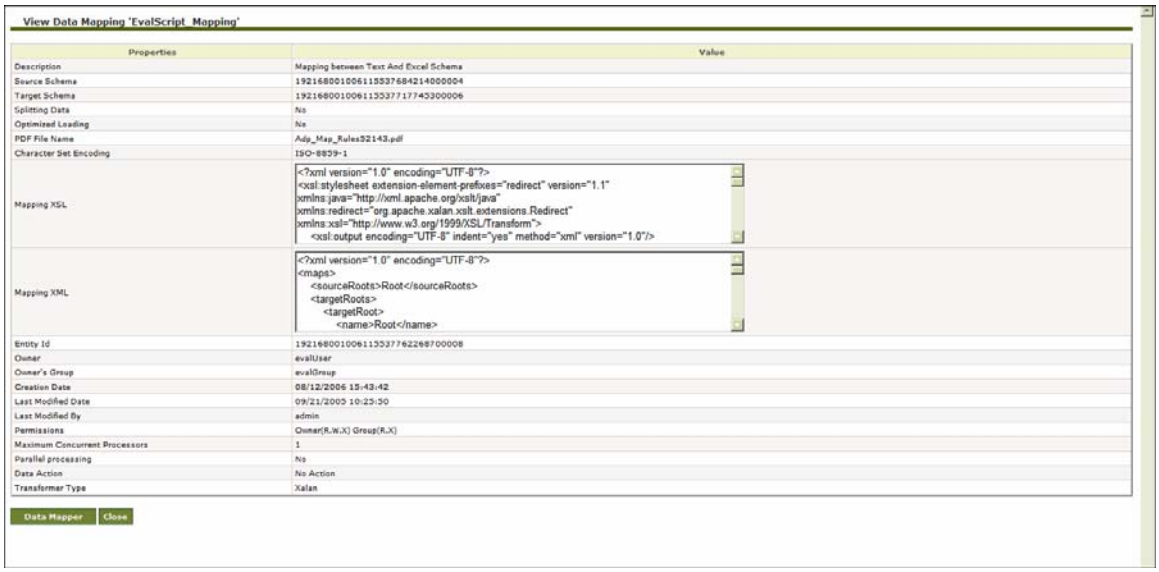


Figure 10.55: View Data Mapping

5. Click **Data Mapper** button. This displays the selected mapping activity in the Data Mapper applet in read-only mode (see Figure 10.56).

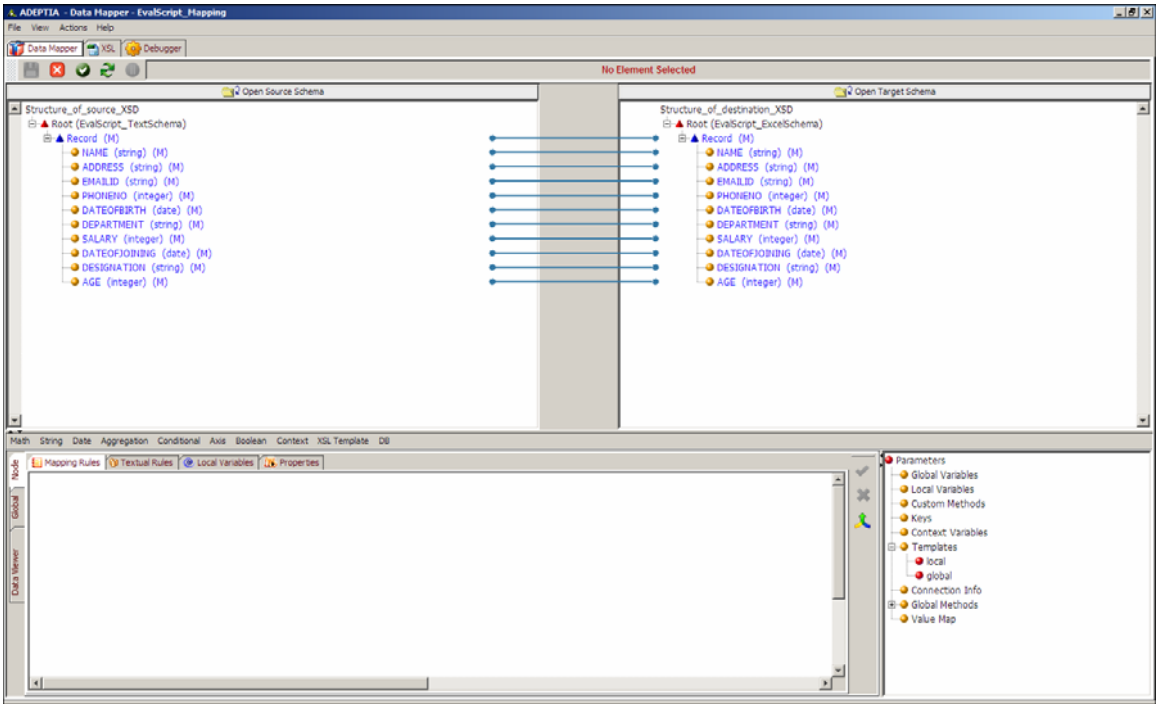


Figure 10.56: Data Mapper in Read-Only Mode

	In the read-only mode all save options in the Data Mapper are disabled.

6. You can edit this mapping activity, by adding new mapping rules, or modifying existing ones. Additionally, you can also run simulation on this mapping.
7. Once you have made all changes, if you try and save the modified mapping, you will not be able to save it, as all save options are disabled in read-only mode. The **Save** () icon and the **Save** option in the **File** menu are disabled. Additionally, after making the changes, if you close the Data Mapper applet, then the pop-up to save the mapping activity before exiting, will not appear.

Splitting Source Data

Splitting feature enables you to process large files. You must use splitting if your source file is huge. If you do not use splitting in this case, then Data Mapper may not be able to map it. If the data in the source file is huge, you can split the source data into different chunks. You can specify the size of the chunks in terms of the number of records or data streams. In the output, however, only one file is generated at the location specified in the target activity.

Steps to split the source data

1. Click **[+] Advanced Properties** on the Create Data Mapping screen to expand the advanced properties of the new Mapping activity (see Figure 10.57).

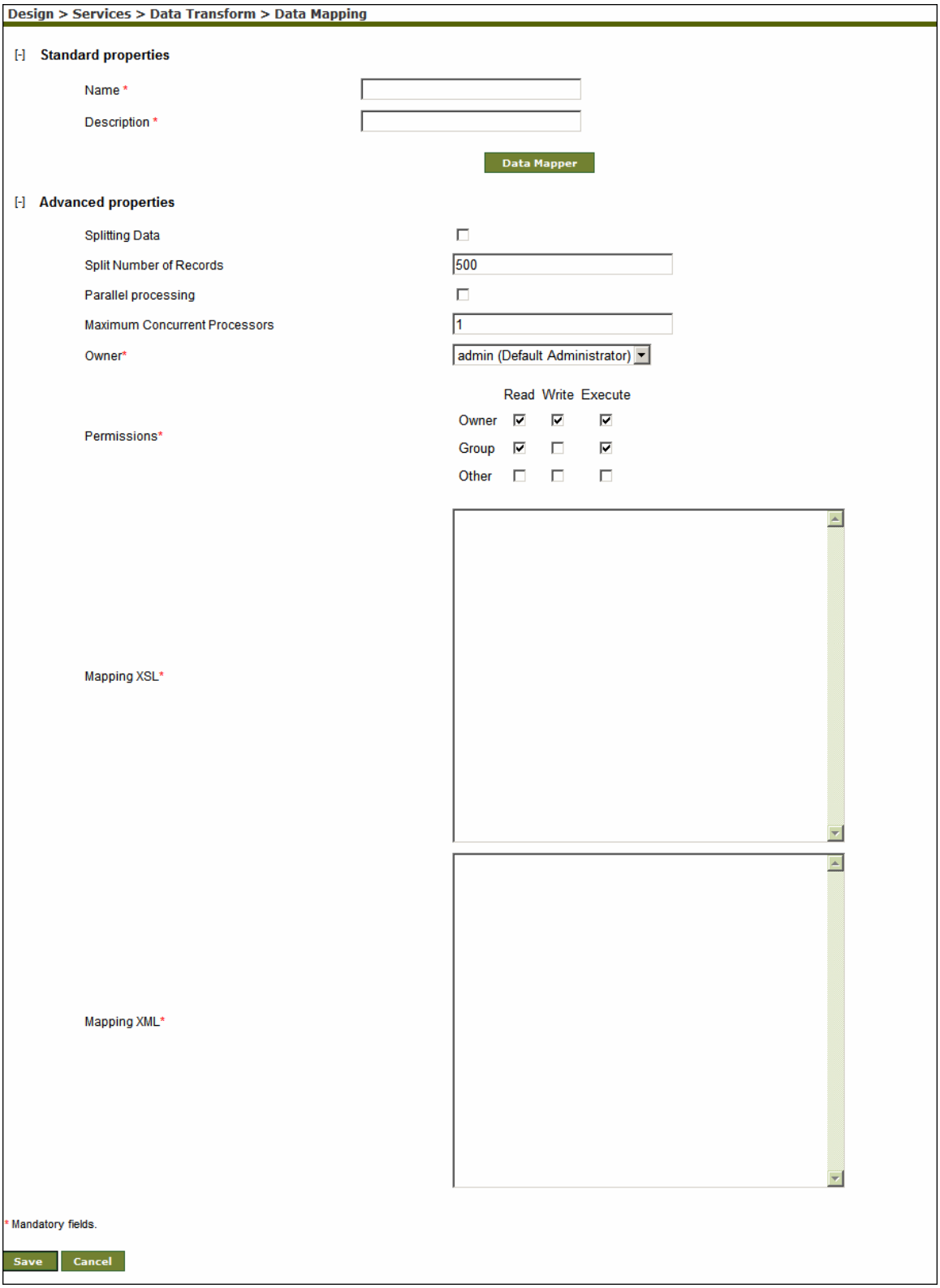

Design > Services > Data Transform > Data Mapping
[-] Standard properties
Name *
Description *
Data Mapper
[-] Advanced properties
Splitting Data
Split Number of Records
500
Parallel processing
Maximum Concurrent Processors
1
Owner*
admin (Default Administrator)
Permissions*
Read Write Execute
Owner
Group
Other
Mapping XSL*
Mapping XML*
* Mandatory fields.
Save
Cancel

Figure 10.57: Advanced Properties

2. Select the *Splitting Data* checkbox to enable the splitting of data.

	If you select this checkbox, then you have to split the data. You can do this by **setting the Splitter XPath** in the Data Mapper applet. Alternately, you can split the data by ***Enabling Quick Splitting*** feature.

3. Enter the number of records into which you want to split the data, in the *Split Number of Records* field. For example, if you have a source file of 100,000 records and you want split this file into records of 10,000 each, then you need to enter 10,000 in this field.
4. Select the *Parallel Processing* checkbox to parallel process these records. By default, records are processed sequentially.
5. Enter the number of records you want to process concurrently, in the *Maximum Concurrent Processes* field. By default, value of 1 is entered.

	It is advised to change this value, if *Parallel Processing* is selected. Else, the records will be processed at one time only and hence sequentially.

6. Select the owner from the *Owner* drop-down list. By default, Administratoristrator is selected.
7. Select the *Read, Write* and *Execute* checkboxes to set permissions for the Owner, Group or Other entities. By default, permissions are selected for the Owner and Group entities.
8. If mapping has been saved in the Data Mapper applet, then the XSL code is displayed in *Mapping XSL* field. This is a very useful feature. You can enter new XSL code or edit existing code to test or debug the mapping. If you save these changes, then they are reflected in the applet. For example, if you need to change the value of a constant, you can change it from this screen itself, instead of the applet. Additionally, any modifications made in the Mapping XSL code in the Data Mapper applet, will overwrite the XSL code in this field.

	If the Advanced Properties and the Data Mapper screens are open simultaneously, then the changes made in the Mapping XSL field in the Advanced Properties screen will not be reflected in the Data Mapper screen.

9. The mapping XML code is displayed in *Mapping XML* field. You can enter new code or edit existing code. If you save these changes, they will be reflected in the applet.
10. Click **Save** to save the splitting details. A screen is displayed where you need to enter comments related to the splitting done. (refer to Figure 10.35).
11. Enter the comments in the *Add Comments* field.
12. Click **OK** to save the comments. This displays a screen confirming that the mapping transformation activity has been updated successfully.

Set Splitter XPath

To split data you need to set the Splitter XPath on the source tree. The data is split and sent to the transformer which executes the split data. You can set the Splitter XPath in case of a single source schema only. You can also set multiple Splitter XPaths in a source schema.

The following example explains the use of multiple Xpath splitting:

Root
|
|- R1
| |-R1C1
| |-R1C2
|- R2
| |-R2C1
| |-R2C2

In this scenario we have **Root** element as the Root for schema. It has two child elements **R1** and **R2** at the first level. R1 element has two children **R1C1** and **R1C2. R2** element has two children **R2C1** and **R2C2**.

If in the source xml, there are multiple occurrences for **R1** and **R2** elements, then we need to set split XPath on both **R1** and **R2** elements. If we set split XPath on **R1** element only then the Data Mapper will assume that **R2** has only one occurrence and vice versa. Similarly, if child elements **R1C1**, **R1C2**, **R2C1** and **R2C2** have multiple occurrences, then also **R1** and **R2** should be set as split XPath. Thus the thumb rule is that the top most node which is repeating, should be set as splitter XPath.

Once splitter XPath is defined, during execution, mapping will split the source tree using each splitter Xpath and each chunk will have records specified by *Split Number of Records*.

In this scenario, when we set **R1** and **R2** as split XPath then *Split Number of Records* property is applied for both **R1** and **R2**. For example if the value of *Split Number of Records* property is 500 then 500 records of **R1** and 500 records of **R2** are selected.

Splitting data and setting Splitter XPath is very effective when the source data is very large. You can set the Splitter XPath in case of a single source and target schema. You can also set multiple Splitter XPaths in a source and target schema.

It is advised to always split the file and set the Splitter XPath if the file size is greater than 50 MB. This enhances the performance and increases data reliability, as in some cases such execution of large data can fail and generate errors.

Steps to set Splitter XPath

1. Ensure that all the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the source schema element that you want to split and select the **Set as Splitter XPath** option. A screen showing the Splitter XPath details is displayed (see Figure 10.58).

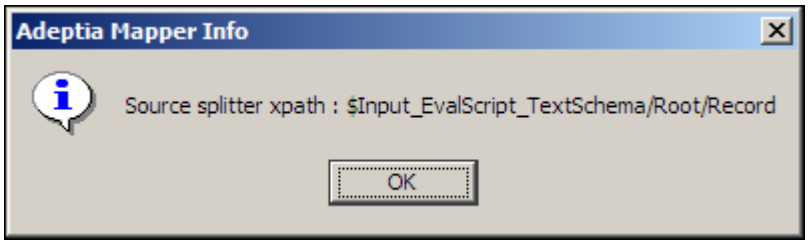


Figure 10.58: Splitter XPath Details

	Splitting can be set only on root or record levels. You cannot split on an element level.

3. Click **OK**. This splits and sets the Splitter XPath on the selected source element.

	Once a Splitter XPath is set for a source element, the letter **(S)** is displayed next to that source element. This signifies that the source element has been split and the Splitter XPath has been set. Refer to **Table of Suffixes** for details on suffixes displayed next to an element.

Once you have set a Splitter XPath, you can view it in the Output XML.

Enabling Quick Splitting

Another way of splitting source data is to enable the quick splitting feature. This splits the source and sets the Splitter Xpath.

Steps to split data by enabling quick splitting

1. Ensure that all the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Select **Actions** menu and select *Enable Quick Splitting* option. Alternately, you can press **<Ctrl> + <Q>**. This splits the data and displays the splitter Xpath details (see Figure 10.59).

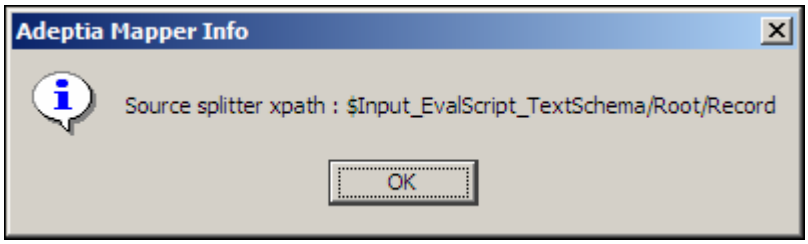


Figure 10.59: Splitter Xpath Details(Enable Quick Splitting)

3. Click **OK**. This splits and sets the Splitter XPath on the selected source element.

However, quick splitting splits the records in default numbers. To split into a specific number, you need to set the Splitter Xpath explicitly.

	Once a Splitter XPath is set for a source element, the letter **(S)** is displayed next to that source element. This signifies that the source element has been split and the Splitter XPath has been set. Refer to **Table of Suffixes** for details on suffixes displayed next to an element.

Steps to view Splitter XPath

1. Click **View** menu and select **View Mapping XML**. The Mapping XML window is displayed. The Splitter XPath details are displayed with a $Input tag prefixed to Source XPath code (see Figure 10.60).

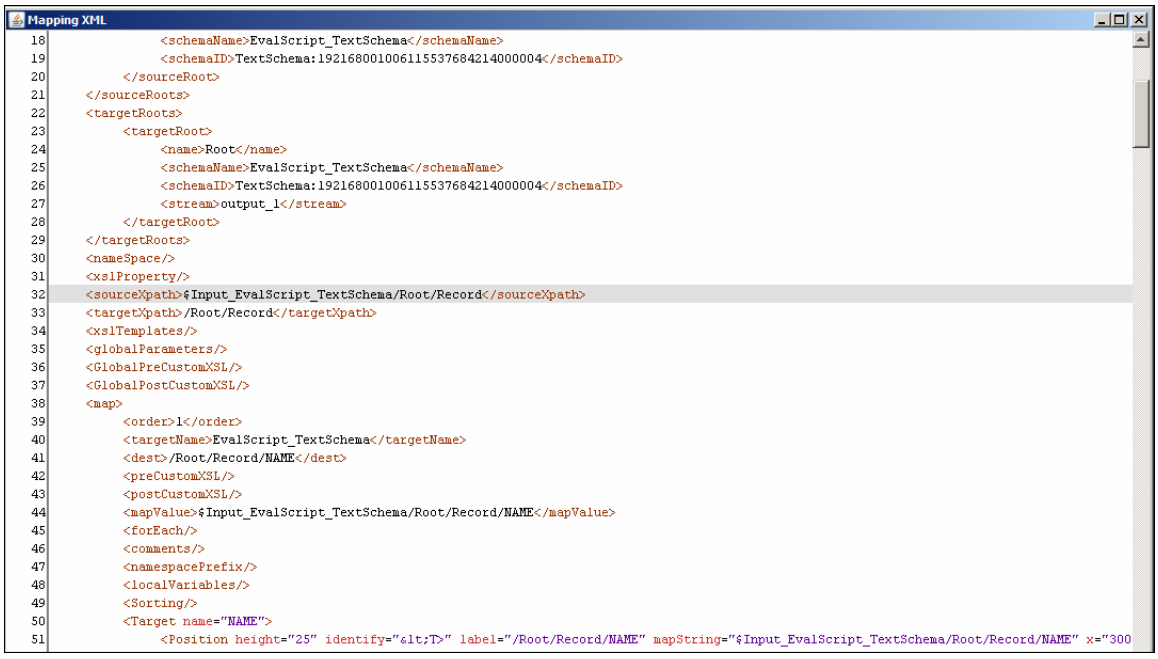


Figure 10.60: Splitter XPath in Mapping XML

You can remove a Splitter XPath that has been set for a source element.

Steps to remove Splitter XPath

1. Right-click the source schema element whose Splitter XPath you want to remove and select the **Remove Splitter XPath** option. A screen showing the Splitter XPath details is displayed (see Figure 10.61).

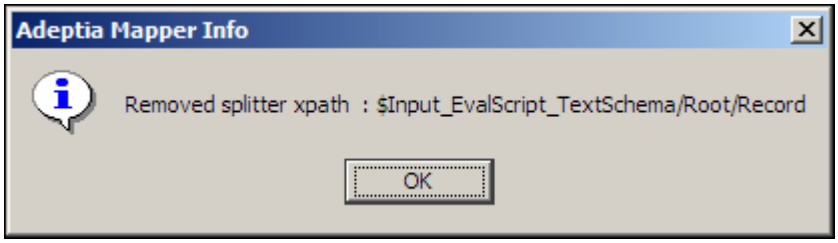


Figure 10.61: Remove Splitter XPath Details

2. Click **OK**. This removes the Splitter XPath set on the selected source element.

Set Merger XPath

Once you have split the source data by setting the Splitter XPath on a source element, you need to merge the split data at the target end. For this, you need to set the Merger XPath on a target element.

Setting the Merger XPath is largely dependent upon the Splitter XPath. You need to ensure that the number of Splitter XPaths and the Merger XPaths at the source and target schemas respectively are the same.

Steps to set Merger XPath

1. Ensure that all the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that a Splitter XPath has been set for a source schema element.
3. Right-click the target schema element on which you want to merge the split data and select the **Set as Merger XPath** option. A screen showing the Merger XPath details is displayed (see Figure 10.62).

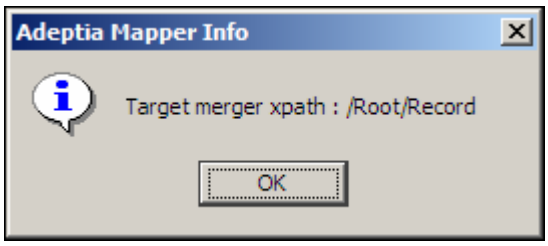


Figure 10.62: Merger XPath Details

	Merging can be set only on root or record levels. You cannot merge on an element level.

4. Click **OK**. This merges the split data and sets the Merger XPath on the selected target element.

	Once a Merger XPath is set for a target element, the letter **(S)** is displayed next to that target element. This signifies that the split data has been merged on the target element and the Merger XPath has been set. Refer to **Table of Suffixes** for details on suffixes displayed next to an element.

	You can view Merger XPath details in the output XML. A $ input tag is prefixed to the line containing the Splitter XPath details.

	When using Enable Quick Splitting feature, the Merger Xpath is also automatically set when this feature is selected.

Once you have set a Merger XPath, you can view it in the Output XML.

Steps to view Merger XPath

1. Click **View** menu and select **View Mapping XML**. The Mapping XML window is displayed. The Merger XPath details are displayed with *Target* XPath code (refer to Figure 10.60).

You can remove a Merger XPath that has been set for a target element.

Steps to remove Merger XPath

1. Right-click the target schema element whose Merger XPath you want to remove and select the **Remove Merger XPath** option. A screen showing the Merger XPath details is displayed (see Figure 10.63).

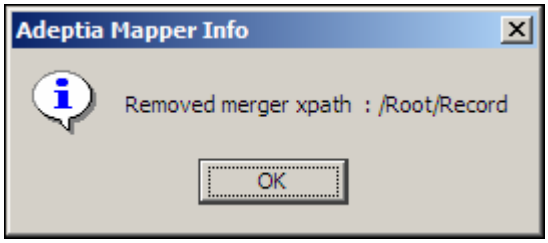


Figure 10.63: Remove Merger XPath Details

2. Click **OK**. This removes the Merger XPath set on the selected target element.

Assign Data Streams

When mapping multiple source and target schema elements, you need to assign streams to the root element of the target schema structures.

Steps to assign data streams

1. Ensure that all the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the root element of a target schema structure and select the **Assign Stream** option. The Assign Streams dialog box is displayed (see Figure 10.64).

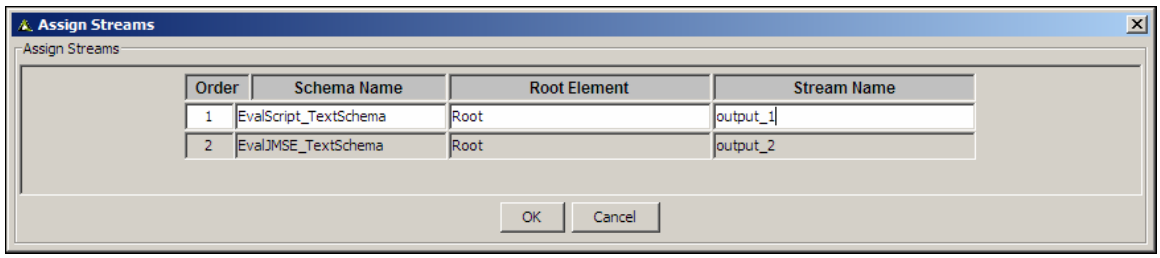


Figure 10.64: Assign Streams

3. This dialog box contains the fields listed in the table below. All fields are populated with values.

Table 10.11: Fields of Assign Streams Screen

Field	Description
Order	Represents the order of occurrence of target schema in the Data Mapper. This field is a read-only field and is automatically populated with value.
Schema Name	Represents the name of the schema which includes the root to which the stream is to be assigned. This field is a read-only field and is automatically populated with value.
Root Element	Represents the root element to which the stream is to be assigned. This field is a read-only field and is automatically populated with value.
Stream Name	Represents the name of the stream to be assigned. This field is automatically populated with value, but is an editable field.

	The root elements, occurrence order and default stream names of all the loaded target schemas are listed in the Assign Streams dialog box. You can edit the stream name.

	It is important to note that the order of streams and the stream names, listed in the Assign Streams dialog box (Mapping) and the Multiple Streams defined in Process Designer should be the same. If you edit a stream name of a mapping activity or add or delete a stream or schema after the activity has already been used in the Process Designer, then you need to update it manually in the Process Designer (Create Multiple Streams section). To know how to use multiple stream in Process

	Designer, refer to the section Creating Multiple Stream.

4. Enter the name of the data stream(s) that you want to edit, for the target element(s), in the *Stream Name* field.

	The name of the data stream does not accept special characters and white spaces.

5. Click **OK** to assign the stream(s), else click **Cancel**.

Remove Mapping of an Element

You can remove mapping associated with a particular element.

Steps to remove mapping of a particular element

1. Ensure that all the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the target element whose mapping you want to remove and select the **Remove Mappings** option. The Remove Options screen is displayed (see Figure 10.65).

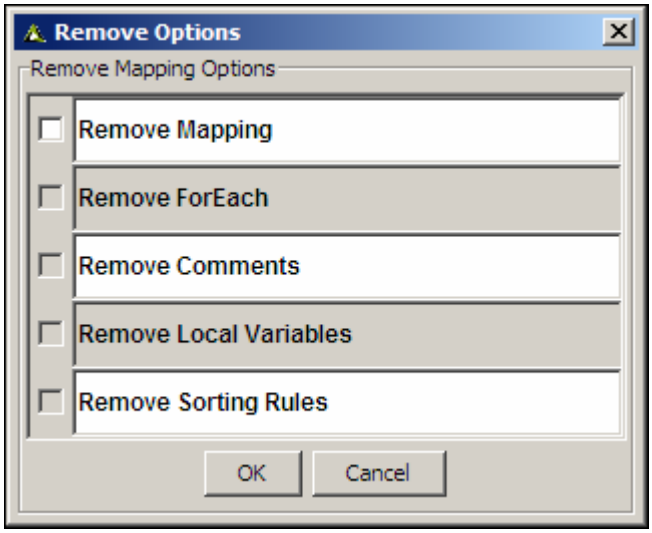


Figure 10.65: Remove Options (Particular Element)

This screen displays a list of properties and variables associated with the selected element that can be removed with the mapping.

	Only the *Remove Mapping* is displayed as always enabled. The other options are enabled only if they have been defined for the particular element.

3. Select the checkbox(s) of the options that you want to remove and click **OK** button. All the selected options and mapping associated with the particular element will be removed.

Using Mapping Functions

You can map source and target schema elements using the various mapping functions built-in the Data Mapper. All these mapping functions comprise of sub-functions, which are used to map elements.

Some mapping functions require use of constant values for mapping elements. For this, you need to add the constant node to the Mapping Graph Area.

Adding a Constant

Steps to add a Constant

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. Right-click the blank space in the Mapping Graph Area and select the **Constants** option. A *Constant* node is displayed in the Mapping Graph Area (see Figure 10.66).

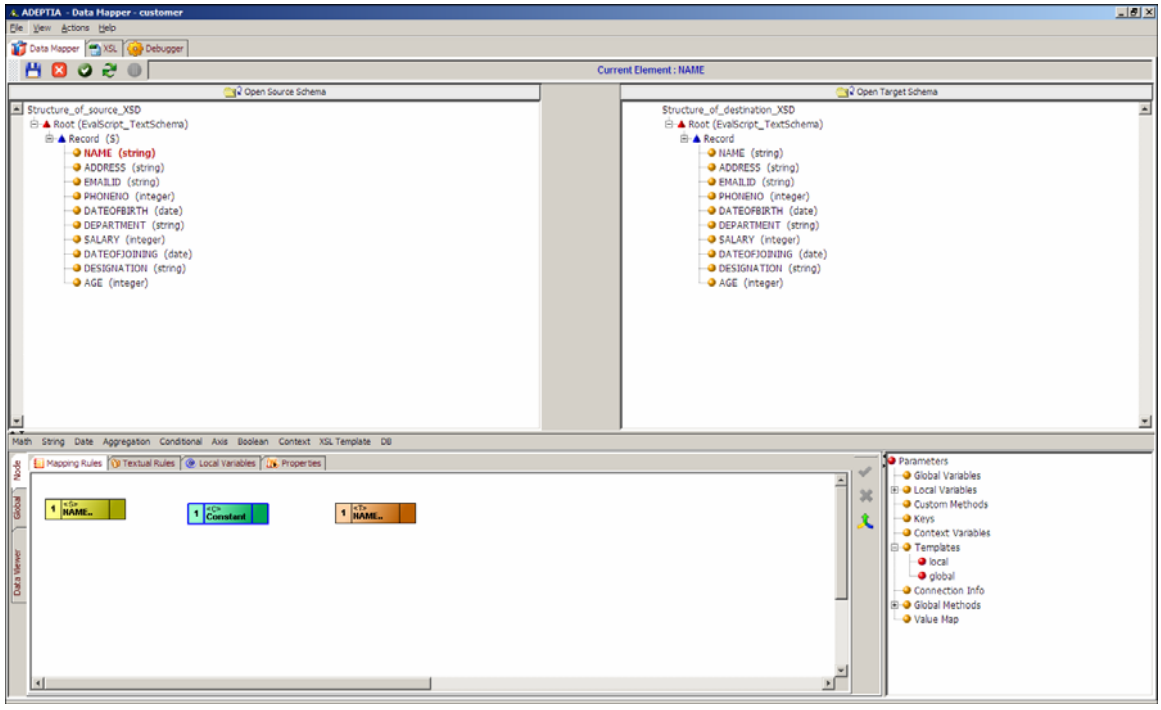


Figure 10.66: Add a Constant Node

4. Double-click the **Constant** node. The Input dialog box is displayed (see Figure 10.67).

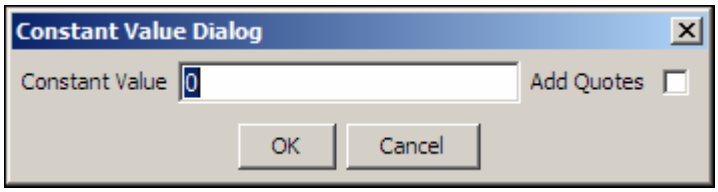


Figure 10.67: Enter Constant Value

5. Enter the desired constant value in the *Constant Value* field.

6. Mark the *Add Quotes* checkbox as checked, if you want to define the constant as a string. Else, the constant is defined as numeric data. By default, this checkbox is unchecked, implying that a constant is defined as a number. However, you can change the constant to string datatype by checking this checkbox anytime.

	If the constant is defined as a string data type, then Apposetrophy (') symbol is supported. Additionally, you can enter constants as an input for Select query too. You can also define an element name as a constant, and leave the *Add Quotes* checkbox as unchecked. This will generate a valid XSL.

7. Click **OK** button. This displays the entered value in the *Constant* node in the Mapping Graph Area.

Using Math Function

The Math mapping function enables you to map elements by performing simple mathematic operations. It comprises of various sub-functions, which are listed in the table below.

Table 10.12: Sub-Functions of Math Mapping Function

Mapping Function	Sub Functions	Description	Example
Math	Add	Adds two numeric elements.	*(10)* + *(5)* returns 15
	Ceiling	Rounds a passed number to the smallest integer that is greater than or equal to the passed number.	Ceiling *(33.9)* returns 34
	Division	Divides two numeric elements.	*(100)* div *(5)* returns 20
	Floor	Rounds a passed number to the largest integer that is not larger than the passed number.	floor *(33.9)* returns 33
	Mod	Returns the remainder of a division between two numeric values.	*(50)* mod *(3)* returns 2

	Multiply	Multiplies two numeric elements.	*(10) * (5)* returns 50
	Round	Rounds a passed number to the nearest integer.	round *(4.6)* returns 5
	Subtract	Subtracts one numeric value from other.	*(10) – (5)* returns 5
	Number	Converts parameter to a number.	number *("-17.3")* returns -17.3
	Format-number	Transforms input data into a specific format. The specific formats allowed are decimals (upto 2 places) and %. This function can be used only if the target node is of string data type.	format-number *(12.5, '$#.00")* returns $12.50

You can use the required sub-function of Math mapping function to map elements. The process of using the Math function is the same for all its sub-functions. Thus, the mapping process using one sub- function is outlined below.

Steps to map elements using the 'Division' Math Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. You need to **add a constant value** for using this function. Once it is added, a *Constant* node with its value is displayed in Mapping Graph Area.
4. Click **Math** function menu and select the **Division** sub-function. A div node is displayed in the Mapping Graph Area (see Figure 10.68).

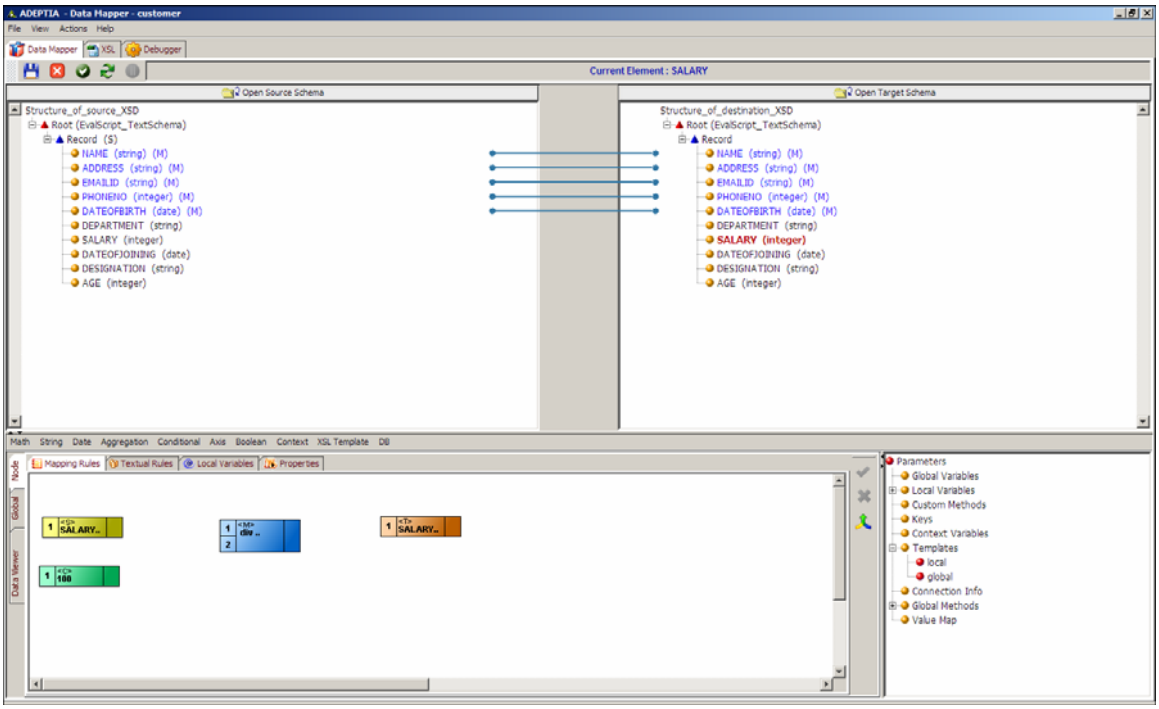


Figure 10.68: Add Div Sub-Function Node

5. Create a link from the output of the *Source* element to the first input of the *div* node.
6. Create a link from the output of *Constant* node to the second input of the *div* node.
7. Create a link from the output of the *div* function node to the *Target* element (see Figure 10.69).

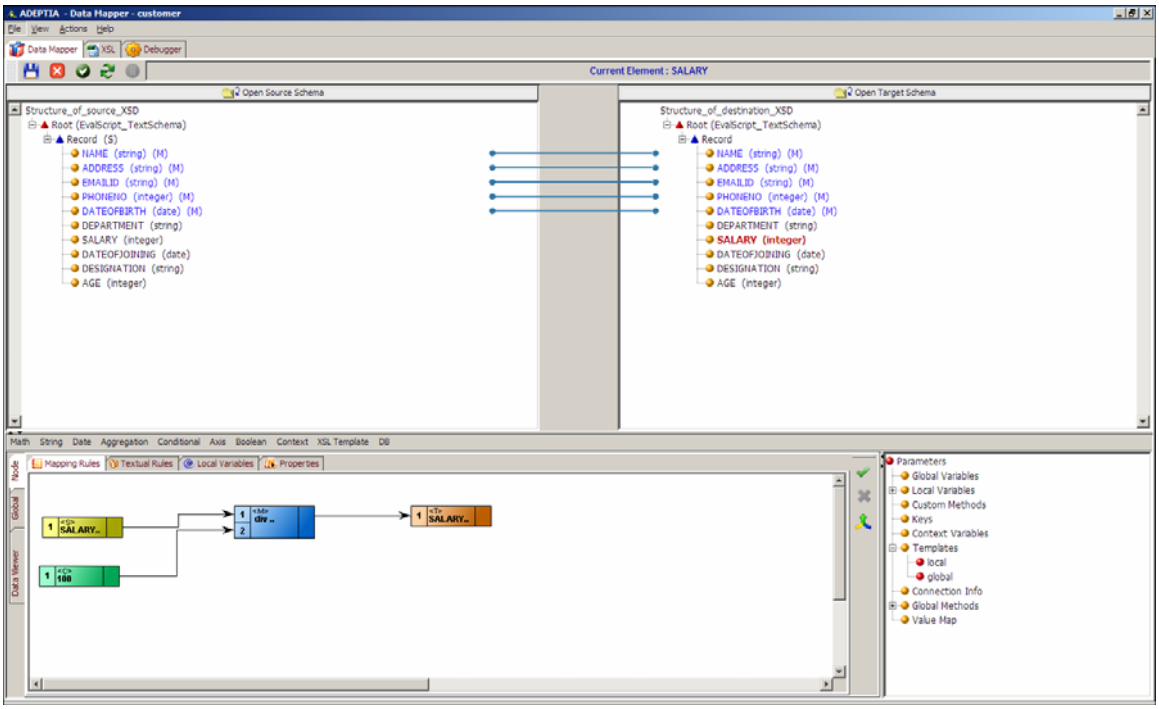


Figure 10.69: Create Links between Nodes for Mapping Elements using Div Function

8. Click **Apply Mapping** ( ) button. This maps elements using the *div* sub-function. A line is displayed between the source and target panels showing the mapping of the source element to the target element.

Using String Function

The String mapping function enables you to map elements by manipulating strings. It comprises of various sub-functions, which are listed in the table below.

Table 10.13 : Sub-Functions of String Mapping Function

Mapping Function	Sub Functions	Description	Example
String	Concat	Concatenates the second string after the first string. This function accepts only two parameters in graphical mode. However, there is no limitation to the number of parameters in text mode.	concat *('Jane', 'Brown')* returns the string JaneBrown

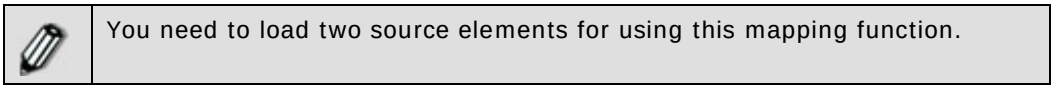
	Substring	Returns the sub string from the string starting from the specified position and of the specified length. When using this function, you always start counting its position from 1. This implies that the second argument (position) is always greater than or equal to 1.	substring *('abcdef' ,3, 4)* returns cdef
	String	Returns the string value of the argument.	string *('1000')* returns the string 1000
	Translate	Translates each occurrence of the first argument in the specified string to the second argument. Both the arguments must consist of one or equal number of characters.	translate *('alphabet', 'a','t')* returns tlphtbet
	Substring-after	Returns the string after the specified argument from the string.	substring-after *('print=yes', '=')* returns yes
	Substring-before	Returns the string before the specified argument from the string.	substring-before *('print=yes', '=')* returns print
	Starts-with	Returns true if the string starts with the specified argument or false otherwise.	starts-with *('Sales', 'S')* returns True
	String-length	Returns the length of the specified string.	string-length *('adeptia')* returns 7
	Contains	Returns true if the string contains the specified argument or false otherwise.	contains *('adeptia', 'tia')* returns True

	Normalize-space	Removes leading and trailing white space (space, tab and new line) from a string, and replaces internal sequences of white space with a single space character.	normalize-space *('Adeptia Inc ')* returns Adeptia Inc
	Name	Returns the name of the source node and field name on which mapping is currently done.	If the context node is an element named *<ms:schema>*, then name () will return ms:schema
	Local-name	Returns name with the name prefix. If no prefix was specified, then name and local name are same.	If context node is an element named *<ms:schema>*, then local-name () will return schema

You can use the required sub-function of String mapping function to map elements. The process of using the String function is the same for all its sub-functions. Thus, the mapping process using one sub- function is outlined below.

Steps to map elements using the 'Concat' String Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.



3. Click **String** mapping function and select **Concat** sub-function. A *Concat* node is displayed in the Mapping Graph Area (see Figure 10.70).

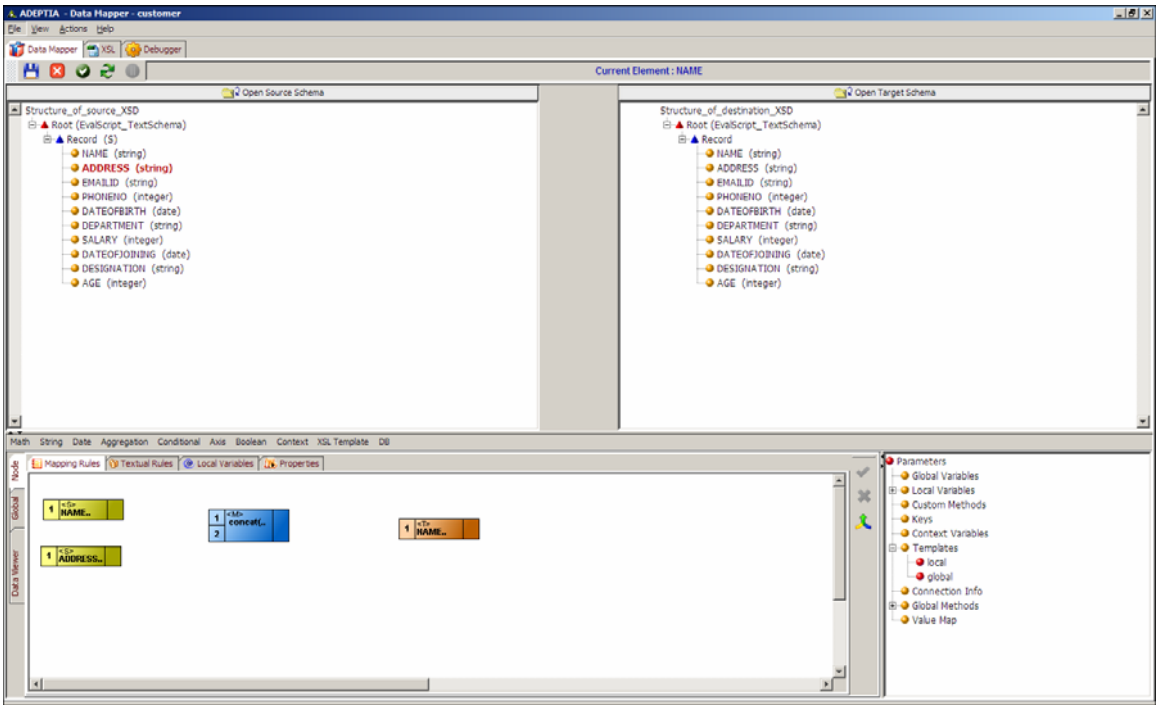


Figure 10.70: Add Concat Sub-Function Node

4. Create a link from the output of the first *Source* element to the first input of the *concat* node.
5. Create a link from the output of the second *Source* element to the second input of the *concat* node.
6. Create a link from the output of the *concat* node to input of the *target* element node (see Figure 10.71).

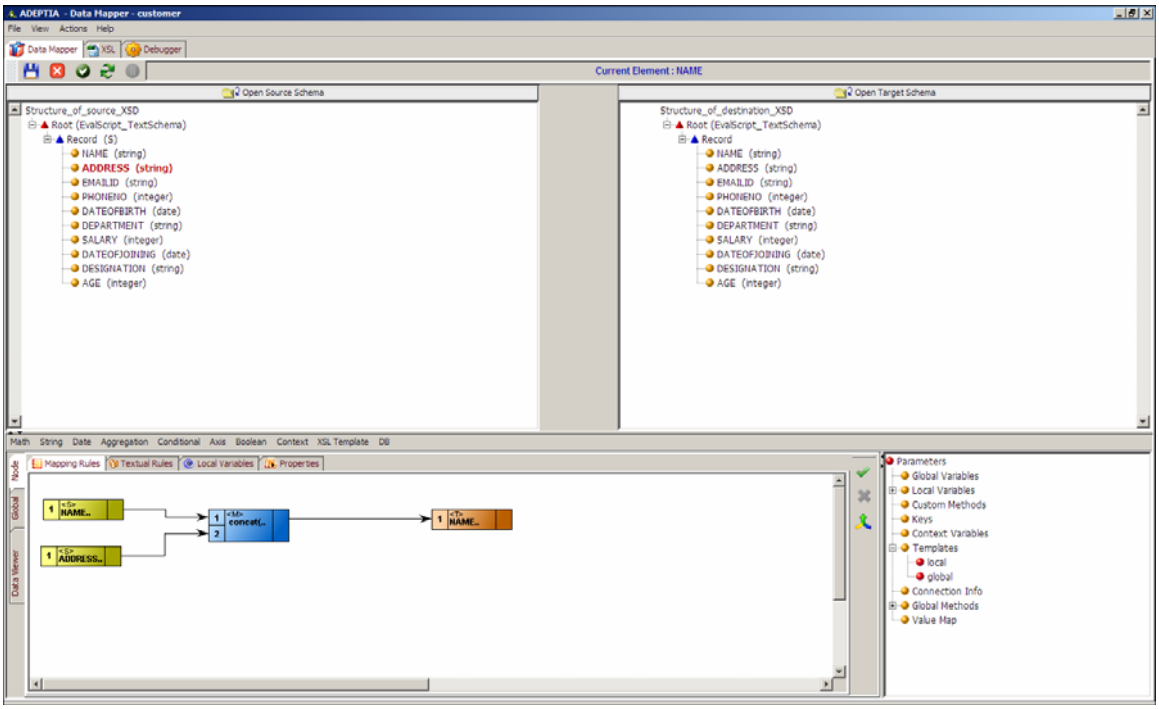


Figure 10.71: Create Links between Nodes for Mapping Elements using Concat Function

7. Click **Apply Mapping** ( ) button. This maps elements using the *concat* sub-function. A line is displayed between the source and target panels showing the mapping of the source element to the target element.

Using Date Function

The Date mapping function enables you to map elements by generating dates in specific formats. It also returns the current date and the difference between two dates. It comprises of various sub-functions, which are listed in the table below.

	All Date functions are applicable, only for XML Schemas at source and target.

Table 10.14: Sub-Functions of Date Mapping Function

Mapping Function	Sub Functions	Description	Example
Date	Date-Format	This is a customized mapping function. Changes the date from the input date format to specified output date format.	date-format *(employee/ 1998-03-31, 'yyyy-mm-dd', 'dd-mm-yyyy')* returns 31-03-1998

	Date -Difference	Generates the difference between two dates in milliseconds. Supports all java date formats.	date-difference*('12-08-2006','dd-mm-yyyy','12-08-2005', 'dd-mm-yyyy')* returns 31536000000
	Current Date	Returns the current system date in Month-day-year format. It accepts the argument according to standard Java arguments as defined for SimpleDateFormat class.	current date *('MM-dd-yyyy')* returns the current date in month-day-year format.

You can use the required sub-function of Date mapping function to map elements. The process of using the Date function is the same for all its sub-functions. Thus, the mapping process using one sub- function is outlined below.

Steps to map elements using the 'Date-Difference' Date Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. You need to **add a constant value** for using this function. Once it is added, a *Constant* node with its value is displayed in Mapping Graph Area.

This function requires four inputs. The first input represents the value of *Date1*. The second input represents the value as the format of *Date1*. The third input represents the value as *Date2*. The fourth input represents the value as format of *Date2*. The *Date1* and *Date2* values can either be in the form of constants or XPath values. The formats of *Date1* and *Date2* are however always in the form of constants.

4. Click **Date** mapping function and select **Date-Difference** sub-function. A date-difference node is displayed in the Mapping Graph Area (see Figure 10.72).

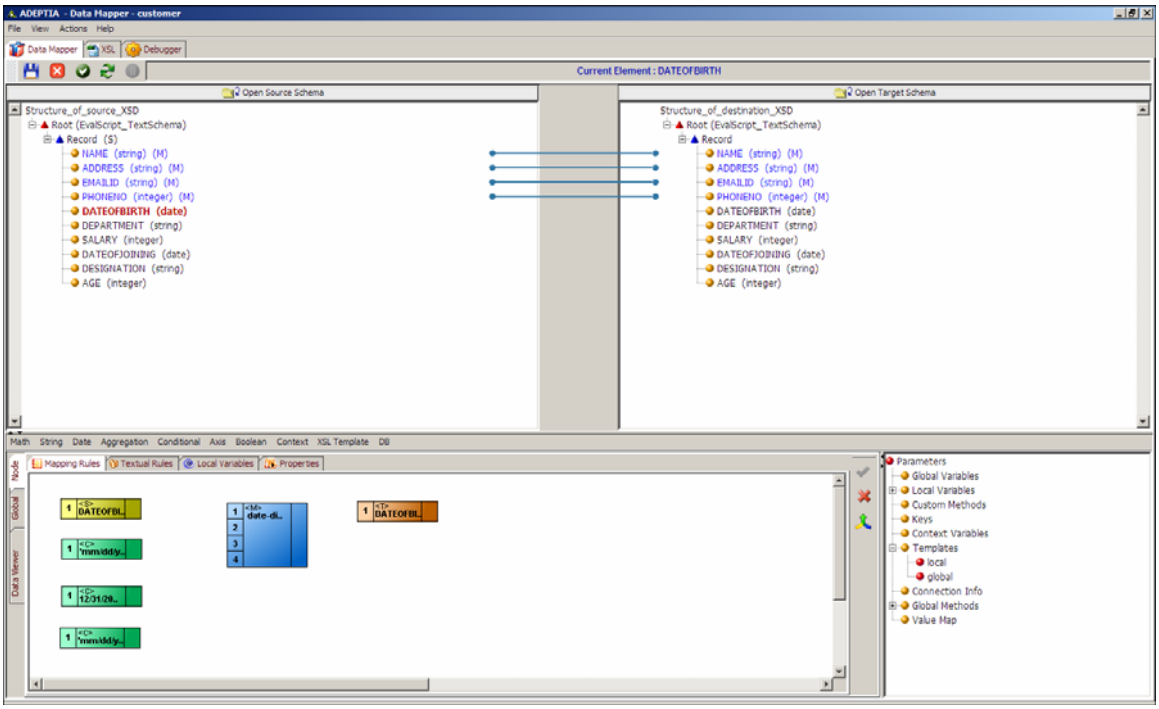


Figure 10.72: Add Date-Difference Sub-Function Node

5. Create a link from the output of the first *constant* value to the first input of the *date-difference* node.
6. Create a link from the output of the second *constant* value to the second input of the *date-difference* node.
7. Create a link from the output of the third *constant* value to the third input of the *date-difference* node.
8. Create a link from the output of the fourth *constant* value to the fourth input of the *date-difference* node.
9. Create a link from the output of the *date-difference* node to input of the *target* element node (see Figure 10.73).

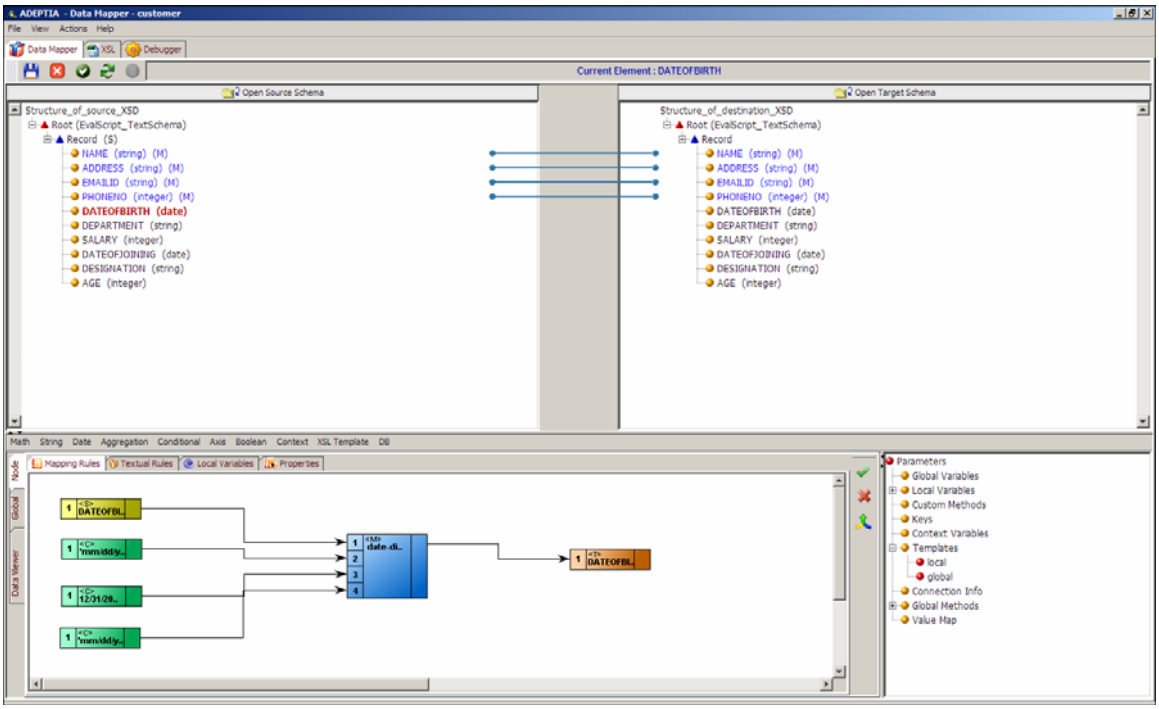


Figure 10.73: Create Links between Nodes for Mapping Elements using Date-Difference Function

10. Click **Apply Mapping** ( ) button. This maps elements by generating the difference between *Date1* and *Date2* using the date-difference sub-function.

Using Aggregation Functions

The Aggregation function enables you to map elements by aggregating or counting the values of all nodes in an element. It comprises of various sub-functions, which are listed in the table below.

Table 10.15: Sub-Functions of Aggregation Mapping Function

Mapping Function	Sub Functions	Description	Example
Aggregation	Sum	Returns total of all values under all nodes of the specified element.	Sum *(Age)* returns the sum of all Age elements under the specified node.
	Count	Returns the total number of nodes for the specified element.	Count *(Age)* returns the total number of records under the Age element.

	Position	Returns the position of the current context node in the specified element.	*position()* returns 1 for the first node, 2 for second node and so on.
	Key	Returns matched records using two parameters: Name of key to be used Fields to be matched	Refer to the section **Using Key Function**.

You can use the required sub-function of Aggregation mapping function to map elements. The process of using the Aggregation function is the same for all its sub-functions. Thus, the mapping process using one sub- function is outlined below.

Steps to map elements using the 'Sum' Aggregation Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. Click **Aggregation** function and select the **Sum** sub-function. A Sum node is displayed in the Mapping Graph Area (see Figure 10.74).

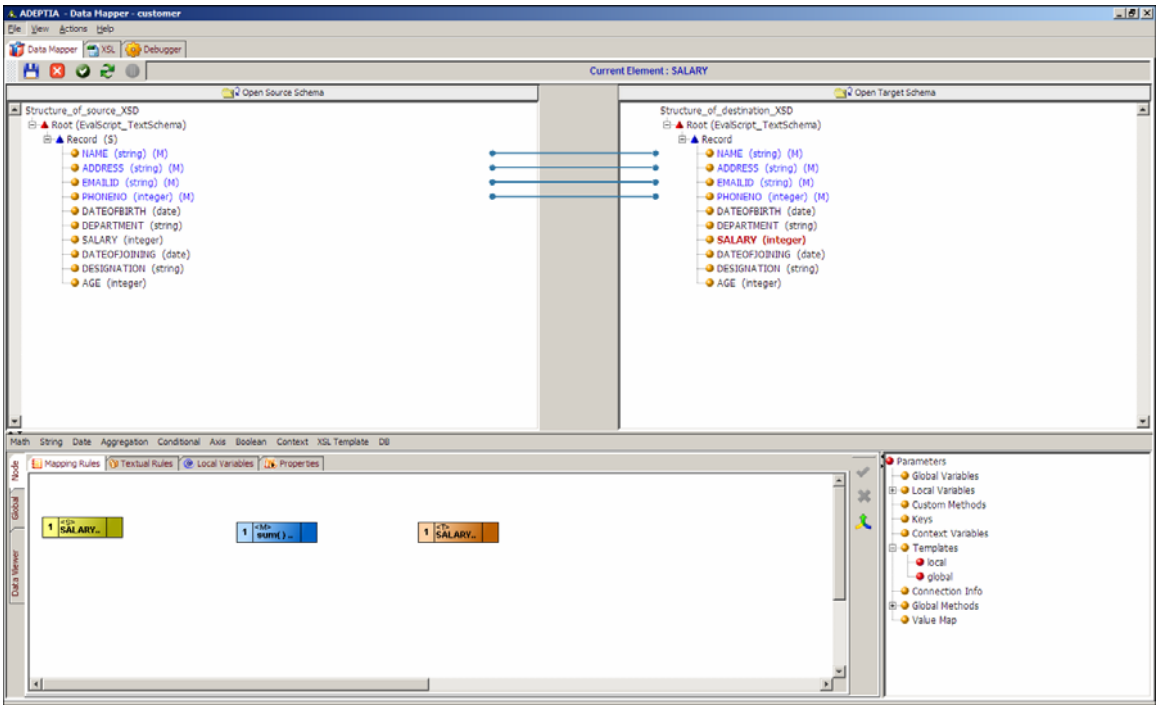


Figure 10.74: Add Sum Sub-Function Node

4. Create a link from the output of the *Source* element to the input of the *Sum* node.
5. Create a link from the output of the *Sum* function node to the *Target* element (see Figure 10.75).

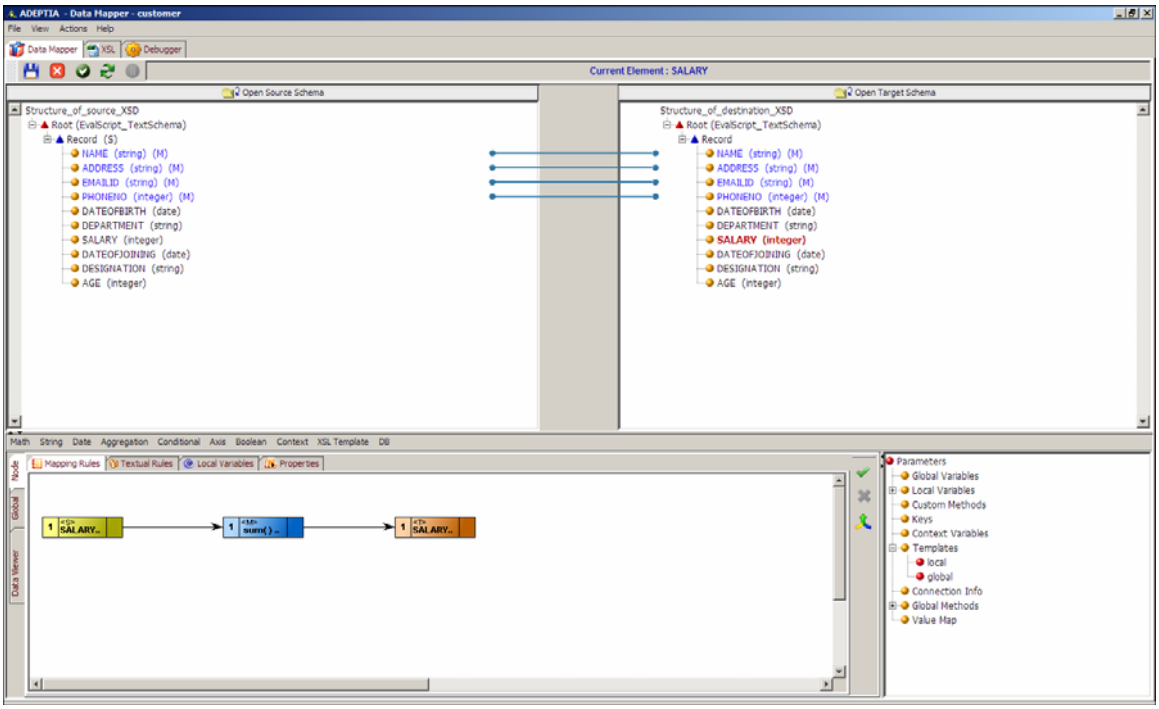


Figure 10.75: Create Links between Nodes for Mapping Elements using Sum Function

6. Click **Apply Mapping** ( ) button. This maps elements using the Sum sub-function. A line is displayed between the source and target panels showing the mapping of the source element to the target element.

Using Conditional Function

The Conditional mapping function enables you to map elements by building conditional expressions. It comprises of various sub-functions, which are listed in the table below.

Table 10.16: Sub-Functions of Conditional Mapping Function

Mapping Function	Sub Functions	Description	Example
Conditional	IF Conditions	This function comprises of two sub functions:	
	For Filtering Records	Filters the records/elements on basis of the specified condition.	*IFF CONDITION {#Age=25} Value= [100]* Returns the filtered target node-set/node if the Age is 25 in the source record.

	For Mapping To Elements	Returns a value if the specified condition is true.	*IF CONDITION {#Age=25} Value= [100]* Returns 100 if the Age has the value of 25 in the source record.
	When Condition	Returns a particular value if the specified condition is true, else it returns another value.	*WHEN Condition {contains (Email_ID, '@')} Value= [100] Otherwise Value= [200]* Returns 100 if the Email_ID contains @, else it returns 200.
	Select Query	This is a customized mapping function. Returns the output based on the applied select query on a table in the any of the three databases (Oracle, Sql, IBM DB2). Select Query supports single field selection and returns first matching record.	*SELECT FIELD = [Value1] FROM DB.TAB = [Value2].[Value3] WHERE {Condition}* Here: Value1 = Field name in the table, whose value is to be retrieved Value 2 = Name of the Database Info object where the table exists. It contains database location and the user ID and password to connect to that database. Value 3 = Name of the table for which the select query is implemented Condition = Condition, based on which value is returned For example: *SELECT FIELD = [EMPNO] FROM DB.TAB = [oracle_databaseserver].[EMP] WHERE {ENAME= 'Smith'}* Returns Employee number of employee from the table EMP with the name Smith.

	Sequence	This is a customized mapping function. Returns integer values starting from the first argument and incrementing it by the second argument for each record. Both the arguments must be integer constants.	*Sequence {10, 20}* Returns 10, 30, 50….and so on. You can also use multiple where condition. Incase of multiple where condition each condition must be within square braces as shown in the example below: *SELECT FIELD = [EMPNO] FROM DB.TAB = [oracle_databaseserver].[EMP] WHERE {[ENAME= 'Smith'] AND [AGE = 28]}*
	Append	This is a customized mapping function. Appends the second string after the first string. Used to append only IF Condition.	*Append ('IF CONDITION {condition} Value [Value1]', ('IF CONDITION {condition} Value [Value2]'* Returns IF CONDITION {condition} Value [Value1] Value [Value2]

Some conditional sub-functions are not included in the Data Mapper screen, but are implemented by extending the 'IF' and 'When' conditions declared in Table 13.14 These sub-functions are listed in Table 13.15.

Table 10.17: Conditional Functions ('If' and 'When')

Sub Function	Custom Function	Description	Example
IF Condition	Nested IF	Contains IF condition inside the value of previous IF condition. Returns output when all IF conditions are satisfied. For example: IF CONDITION {condition1} VALUE=[IF CONDITION {condition2} VALUE=[IF	*IF CONDITION {#Age=25} Value= [IF CONDITION {#Designation= 'Manager'} Value = IF CONDITION {#Salary= 20000} Value = [100]]]* Returns 100 if all conditions are satisfied such that a manager of age 25 has a salary of

		CONDITION {condition3} VALUE=[Value1]]] Returns Value1 when all IF conditions are satisfied.	20000.
When Condition	Nested When	Contains When condition inside the value of previous When condition. Returns varied output based on the When conditions being satisfied. For example: WHEN CONDITION {condition1} VALUE=[WHEN CONDITION {condition2} {VALUE=['value1'] OTHERWISE VALUE=['value2']] OTHERWISE VALUE=['value3'] Returns Value1 when both When conditions are satisfied. Returns Value3 when the first When condition fails. Returns Value2 when the first When condition is satisfied, but the second one fails.	*WHEN CONDITION {/employees/employee/Age=25} Value= [WHEN CONDITION {/employees/employee/Designation= 'Manager'} Value = ['100'] OTHERWISE Value = ['200'] OTHERWISE Value = ['300']]* Returns 100 if both conditions are satisfied such that the employee is a manager of age 25. It returns 300 if the employee is a manager but not of 25 years of age. It returns 200 if the employee is 25 years of age, but is not a manager.
IF and When Condition	IF and When conditions return Boolean	Returns value based on the function used as whether the condition returns True or False. If a string or source element contains a character or group of characters, then True is returned, else False is returned. Based on the return value of the condition, the output value is generated. For example:	*WHEN CONDITION {contains (var1, str)} Value= ['100'] OTHERWISE Value = ['200']* Returns true if Var1 contains Str, else it returns False. If the condition returns true, then this function returns 100, else it returns 200.

		WHEN CONDITION {contains (var1, str)} VALUE=[var2] OTHERWISE VALUE=[var3] If var1 contains str, then True is returned, else False is returned. If the condition returns True, then var2 is generated as output, else var1 is the output value.	
IF and When Condition	Mixed condition	Contains IF condition inside the value part of When condition or vice versa. Returns varied output based on the conditions being satisfied. For example: WHEN CONDITION {condition1} VALUE=[IF CONDITION {condition2} VALUE=['value1']] OTHERWISE VALUE=['value2'] Returns Value1 when both conditions are satisfied. Returns Value2 when the first condition fails. Returns no output when the first condition is satisfied but second condition fails or when both conditions fail.	*WHEN CONDITION {/employees/employee/ Salary=50000} Value= IF CONDITION {/employees/employee/ Designation= Project Manager} Value = ['100'] OTHERWISE Value = ['200']* Returns 100 if the employee is a Project Manager and has a salary of 50000. It returns 200 if the employee is a Project Manager but does not have a salary of 50000.
IF Condition	Multiple IF	Contains IF conditions placed sequentially, with each IF having its value part. Two IF conditions are appended using Append sub-function of the String Function. Returns varied output	*IF CONDITION {#Age=25} Value= ['500'] IF CONDITION {#Designation= 'Manager'} Value = [1000]* Returns 500 if employee is of age 25. It returns

		based on the condition being satisfied. For Example: IFCONDITION {condition1} VALUE=['value1'] IF CONDITION {condition2} VALUE=['value2'] Returns Value1 if condition1 is satisfied. Returns Value2 if condition2 is satisfied.	1000 if employee is a Manager.
When Condition	When condition with more than one conditions in it.	Contains When condition containing logical operators like and/not to make multiple conditions inside the When condition. There are many permutations/combinations for this sub-function. Example 1: WHEN CONDITION (cond1 and cond2) Value = ['value1'] otherwise value = ['value2'] Returns Value1 if both conditions are satisfied, else returns Value2. Example 2: WHEN CONDITION (cond1 or cond2) Value = ['value1'] otherwise value = ['value2'] Returns Value1 when either one or both conditions are satisfied, else returns Value2.	*WHEN CONDITION {/employees/employee/Age=25 and /employees/employee/Designation=Manager} Value= ['100'] OTHERWISE Value = ['200']* Returns 100 if both conditions are satisfied such that the employee is a manager of age 25. Else it returns 200.
IF Condition	IF condition with more than one conditions in it.	Contains IF condition containing logical operators like and/not to make multiple conditions inside the IF condition.	*IF CONDITION {/employees/employee/Age=25 and /employees/employee/Designation=Manager} Value= ['100']*

		There are many permutations/combinations for this sub-function. Example 1: IF CONDITION (cond1 and cond2) Value = [] Returns Value1 when both conditions are satisfied, else no output is returned. Example 2: IF CONDITION (cond1 or cond2) Value = [] Returns Value1 when either one or both conditions are satisfied, else no output is returned.	Returns 100 if both conditions are satisfied such that the employee is a manager of age 25. Else it returns 200.

You can use the required sub-function of Conditional mapping function to map elements. The process of using the Conditional function is the same for all its sub-functions. Thus, the mapping process using one sub- function is outlined below.

Steps to map elements using the 'IF' Condition (IF CONDITION {} VALUE=[]) Conditional Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. Click **Boolean** function menu and then select the **>=Greater than Equal** operator. A >= node is displayed in the Mapping Graph Area (see Figure 10.76).

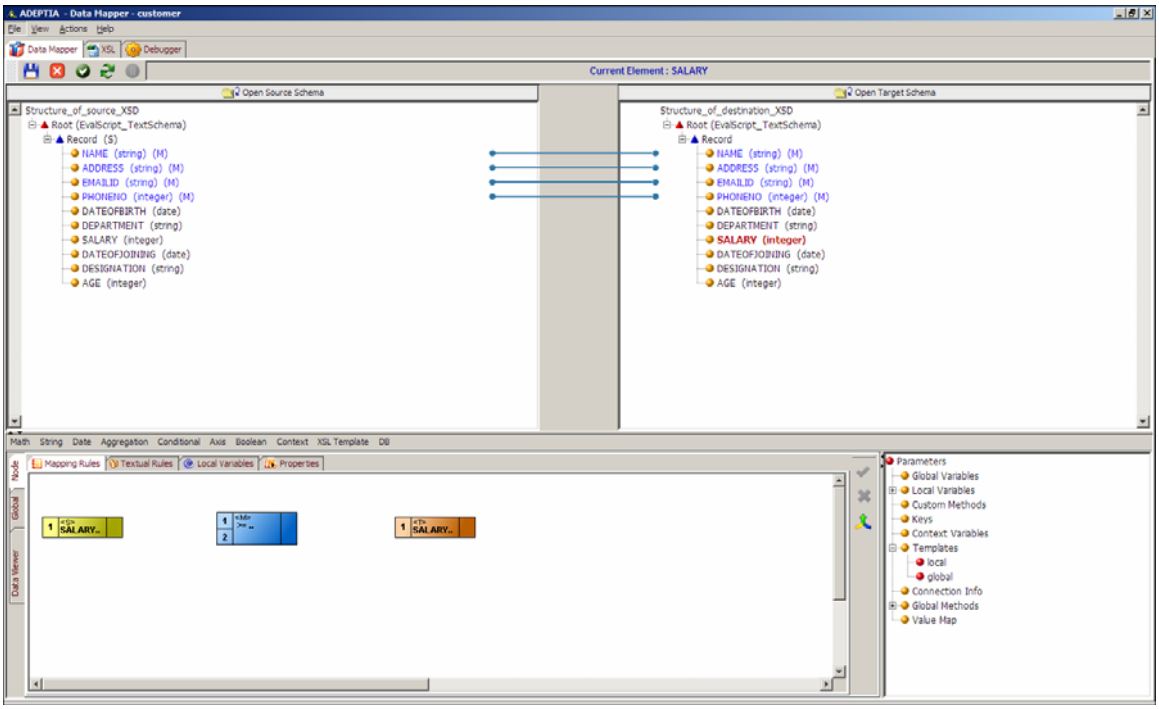


Figure 10.76: Add Greater Than or Equal To Boolean Function Node

	To learn more about Boolean Functions, refer to the section **Using Boolean Functions.**

4. **Add a constant value** in the Mapping Graph Area.
5. Create a link from the output of the *source* element to the first input of the >= operator node.
6. Create a link from the output of the *Constant* node to the second input of the >= operator node (see Figure 10.77).

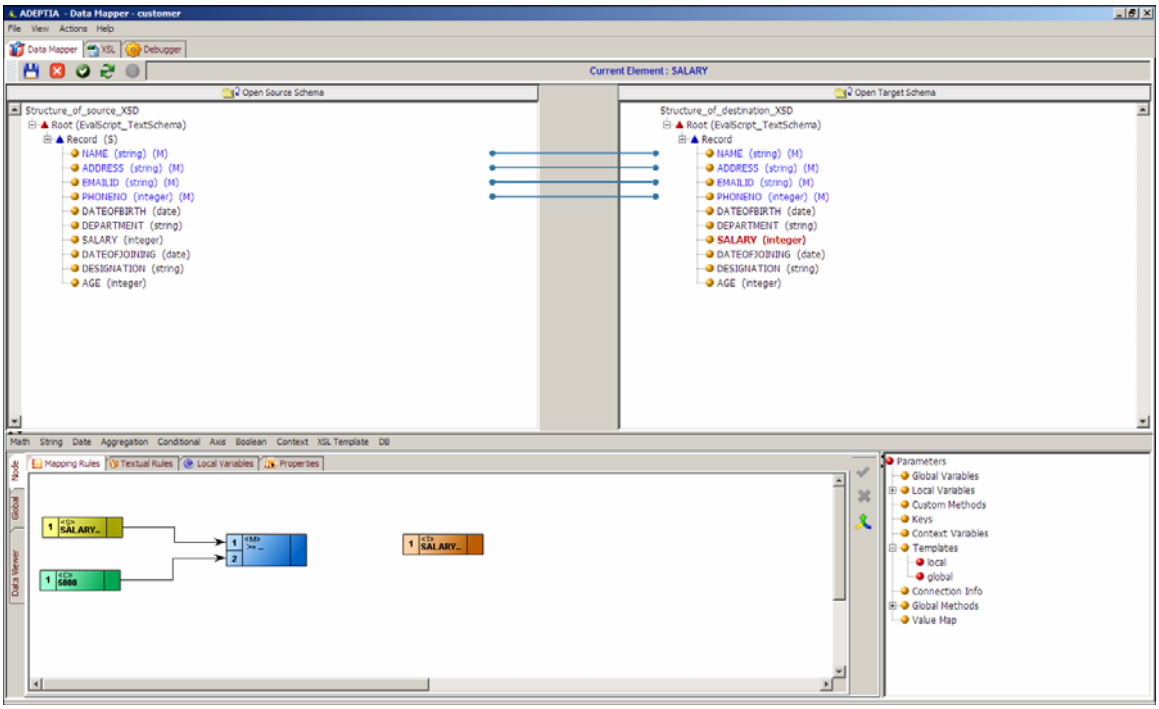


Figure 10.77: Create Links between Nodes for Mapping Elements using Greater Than Equal To Function

7. Click **Conditional** mapping function and select **IF CONDITION** sub-function. Further select For Mapping to Elements option. An *IF CONDITION* node is displayed in the Mapping Graph Area (see Figure 10.78).

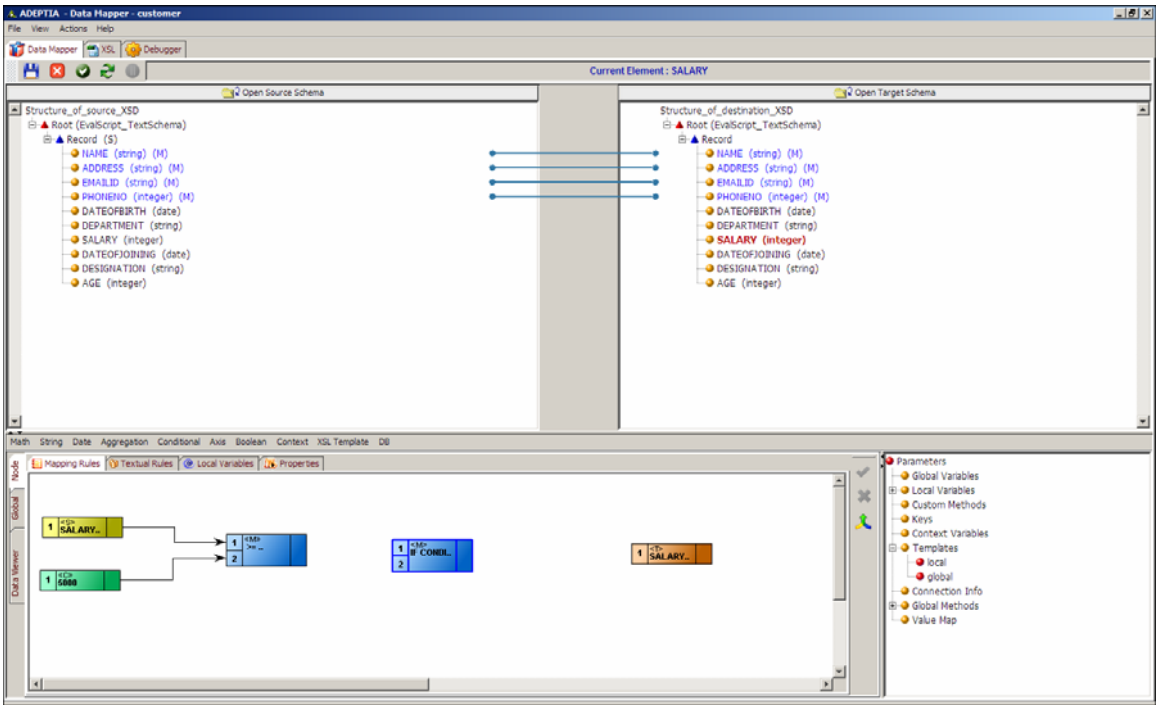


Figure 10.78: Add IF Conditional Function Node

8. Create a link from the output of the >= operator element to the first input of the *IF CONDITION* node.
9. **Add a constant value** of integer type (for example, 25) in the Mapping Graph Area.
10. Create a link from the output of the *Constant* element to the second input of the *IF CONDITION* node.
11. Create a link from the output of the *IF CONDITION* node to the input of the *target* element node (see Figure 10.79).

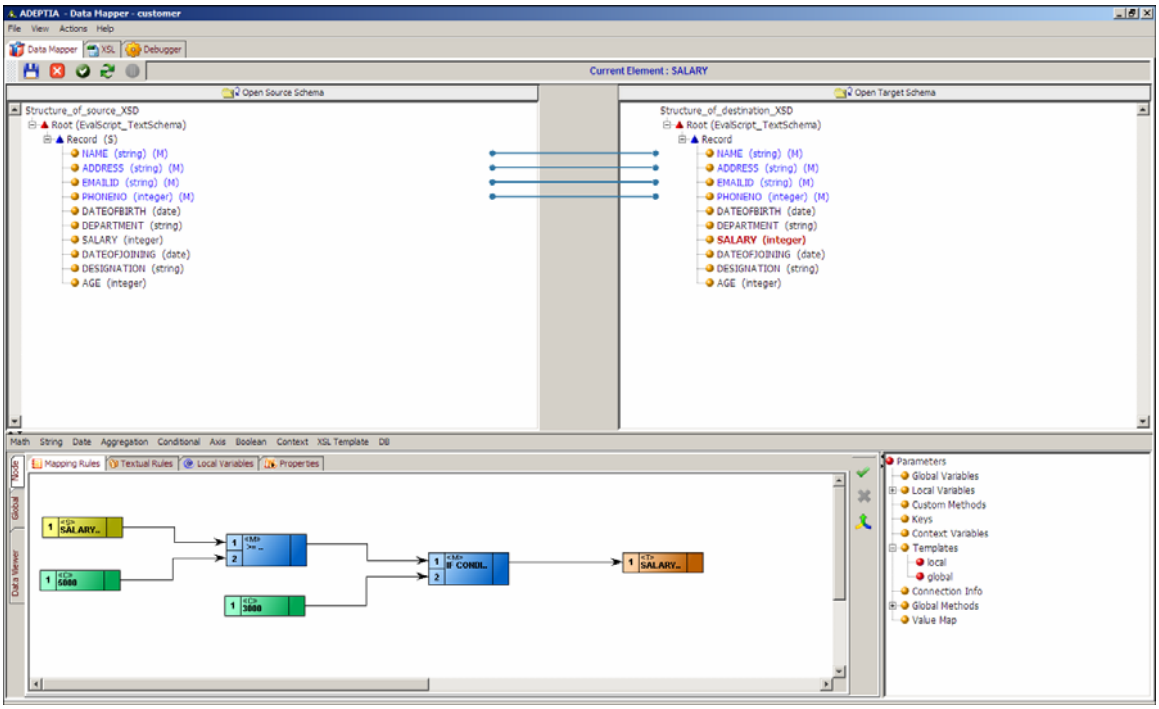


Figure 10.79: Creating Links between Nodes for Mapping Elements using IF Conditional Function

	If the value of *SALARY* element in the source data is greater than or equal to 5000, output will be 3000. If the value of *TESTCASENO* element in the source data is less than 100, output will be an empty tag.

12. Click **Apply Mapping** () button. This maps elements using IF CONDITION function. A line is displayed between source and target panels showing the mapping of the source element to the target element.

	This function simply filters the records based on the specified Boolean condition, but it does not create a data stream of the filtered records. You need to explicitly create the filtered data stream by creating another mapping (or adding same output schema in Target Panel) and using the IFF CONDITION () function based on the opposite Boolean condition.

Using Axis Function

The Axis function contains a list of advance XSL functions that are used to map elements only when the For Each function is applied on the Target node. It comprises of various sub-functions, which are listed in the table below.

Table 10.18: Sub-Functions of Axis Mapping Function

Mapping Function	Sub Functions	Description	Example

Axis	File Path – XPath	This is a customized mapping function. This is an XML Lookup which returns the value of the element/attribute within the XML file whose path is specified. The File path and XPath must be specified in single quotes.	FilePath *['C:/correct_file.xml'] XPath {'/employees/employee/Name'}* Returns value of Name element from the file correct_file.xml.
	Self	Returns the element which is the context node itself.	*Self::Employee* Returns Employee which is the context node itself.
	Child	Returns the element which is the child of the context node.	*Self::Name* Returns Name which is a child of the context node.
	Parent	Returns the element, which is the parent (if any) of the context node.	*Parent::Name* Returns Name which is a parent of the context node.
	Following-sibling	This axis contains all the following siblings of the context node. It returns the first occurrence of the element/attribute in the next sibling of the context node. If the context node is an attribute node or namespace node, then this axis is empty.	*following-sibling: employee/Name* Returns the first occurrence of Name in the next employee node.
	Ancestor	Returns the element which is the ancestor of the context node. This axis always includes the root node, unless the context node is the root node.	*Parent::Name* Returns Name which is an ancestor of the context node.

	Ancestor-or-self	This axis contains the context node and the ancestors of the context node. It always includes the root node. It returns the context node or ancestors of the context node.	*ancestor-or-self::employees/@val* Returns context node or ancestors of context node.
	Preceding-sibling	This axis contains all the preceding siblings of the context node. It returns the first occurrence of the element/attribute in the preceding sibling of the context node. If the context node is an attribute node or namespace node, then this axis is empty.	*preceding-sibling: employee/Name* Returns the first occurrence of Name in the previous employee node.
	Descendant	Returns the element which is the descendant of the context node. This axis does not contain attribute nodes.	*descendant::Name* Returns Name, which is a descendant of the context node.
	Descendant-or-self	This axis contains the context node and the descendants of the context node. It returns the context node or descendants of the context node.	*descendant-or-self::employees/salary* Returns context node or descendants of context node.
	Generate-id	Generates a unique id for the element/attribute specified by the XPath argument.	*generate-id (employees/employee/Name)* Returns unique ID for Name element
	Current	Removes all tags from the specified xml file and returns the data.	*current()* Returns data of the xml file without tags.

You can use the required sub-function of Axis mapping function to map elements. The process of using the Axis function is the same for all its sub-functions.

Using Boolean Function

The Boolean mapping function enables you to map elements by applying various logical operators in building Boolean expressions. It comprises of various sub-functions, which are listed in the table below.

Table 10.19: Sub-Functions of Boolean Mapping Function

Mapping Function	Sub Functions	Description	Example
Boolean	Equal	Returns TRUE the first variable argument is equal to the second variable argument.	*(5) = (5)* returns true
	Not Equal	Returns TRUE the first variable argument is not equal to the second variable argument.	*(5) != (50)* returns true
	Greater Than	Returns TRUE the first variable argument is greater than the second variable argument.	*(5)> (3)* returns true
	Less Than	Returns TRUE the first variable argument is less than the second variable argument.	*(5) < (8)* returns true
	Greater than Equal	Returns TRUE the first variable argument is greater than or equal to the second variable argument.	*(5)>= (5)* returns true *(5)>= (3)* returns true
	Less than Equal	Returns TRUE the first variable argument is less than or equal to the second variable argument.	*(5) <= (8)* returns true *(5) <= (5)* returns true

	Or	Returns TRUE if either of the two variable arguments evaluate to TRUE otherwise FALSE. These variable arguments can only be of Boolean data type. Hence, their value can either be TRUE or FALSE only.	*(True) Or (True)* returns true *(True) Or (False)* returns true
	And	Returns TRUE if both of the two variable arguments evaluate to TRUE otherwise FALSE. These variable arguments can only be of Boolean data type. Hence, their value can either be TRUE or FALSE only.	*(True) And (True)* returns true *(True) And (False)* returns false

A Boolean sub-function is always used in conjunction with a Conditional function. Thus for the process of using the Boolean function, refer to **Steps to map elements using the IF Conditional Mapping Function.**

Using Context Functions

The Context mapping function enables you to map elements by setting or getting value of process flow context variables. It comprises of various sub-functions, which are listed in the table below.

Table 10.20: Sub-Functions of Context Mapping Function

Mapping Function	Sub Functions	Description	Example

Context	Set-Context	This is a customized mapping function. Sets the value of the specified variable argument in the process flow context. This value can be string constant, numeric constant, XPath or output of limited set of mapping function, which can be used before set-context function. This function does not return a value. If mapped to a target element, it creates an empty tag of that target element in output XML.	*set-context ('index', '123')* sets the value of index variable as 123 in the process flow.
	Get-Context	This is a customized mapping function. Returns the value of the specified variable argument from process flow Context. This value can be string constant, numeric constant, XPath or output of limited set of mapping function. If no value has been set, then the value specified in the second argument is returned.	*get-context ('index', '100')* Returns the value of index variable from the process flow. If no value has been set for index, then 100 is returned. In get-context function you can also use any variable, which is not declared. In such case get-context creates a variable with the specified name and value, which can be further used with the selected target element. This can be done only in Textual Rule panel.

You can use the Set-Context sub-function of Context mapping function to set value of the process flow variable and send it to the Process Designer. The Get-Context sub-function is used to return the value of the process flow variable.

Using Set-Context Function

You can use the Set-Context Function in various ways. These are outlined as:

- **Mapping Set-Context function using local variable**
- **Global declaration of Set-Context function**

- **Mapping Set-Context function to any target element**

Mapping Set-Context function using local variable

Concept

You can map the Set-Context function to a local variable, when you want to set the value of process flow context variable based on output of a mapping function or condition. You can pass the output of mapping function or condition to a local variable and use that local variable as value of set-context function.

For example, if you want to set the value of process flow context variable based on the output of WHEN condition. The WHEN condition is used to return "True" when value of a source element "SALARY" is greater than 8000, else it will return "False". The output of WHEN condition is mapped to a local variable "*BolSalary8K*". To set the value of "BolSalary8K" to the process flow context variable, you can use Set-Context function. You have to create another local variable "*Var1*", where value of process flow context variable "*ContextVar*" is set using Set-Context function. This is done by mapping Set-Context to the value part of the local variable *"Var1*". The first argument of the Set-Context function is the name of the process flow context variable "*ContextVar*" and second argument is the local variable "*BolSalary8K*", which contains the output of WHEN condition.

Advantages

- Need not be mapped to a target element
- Local variable can be used as second argument of Set-Context function
- XPath of a source element can be used as second argument of Set-Context function
- Set-Context function is executed for each record of the source data

Disadvantages

- Local Variable used to map set-context function will have an empty value.

Map Set-Context Function using Local Variable

Steps to pass output of mapping function or condition to a local variable

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
3. Click **Local Variables** tab. The Local Variable pane is displayed. (refer to Figure 10.105).
4. Click any *target* element to declare the local variable in the Local Variable pane.
5. Click *Local Variable Name* text field and enter the name of the local variable you want to create (e.g. BolSalary8K). Press the **Tab** or **Enter** key. This takes the control to the Local Variable Value field.

6. Enter the value of the local variable in the *Local Variable* Value field. In this example, you have to specify the WHEN condition as displayed in Figure 10.80.

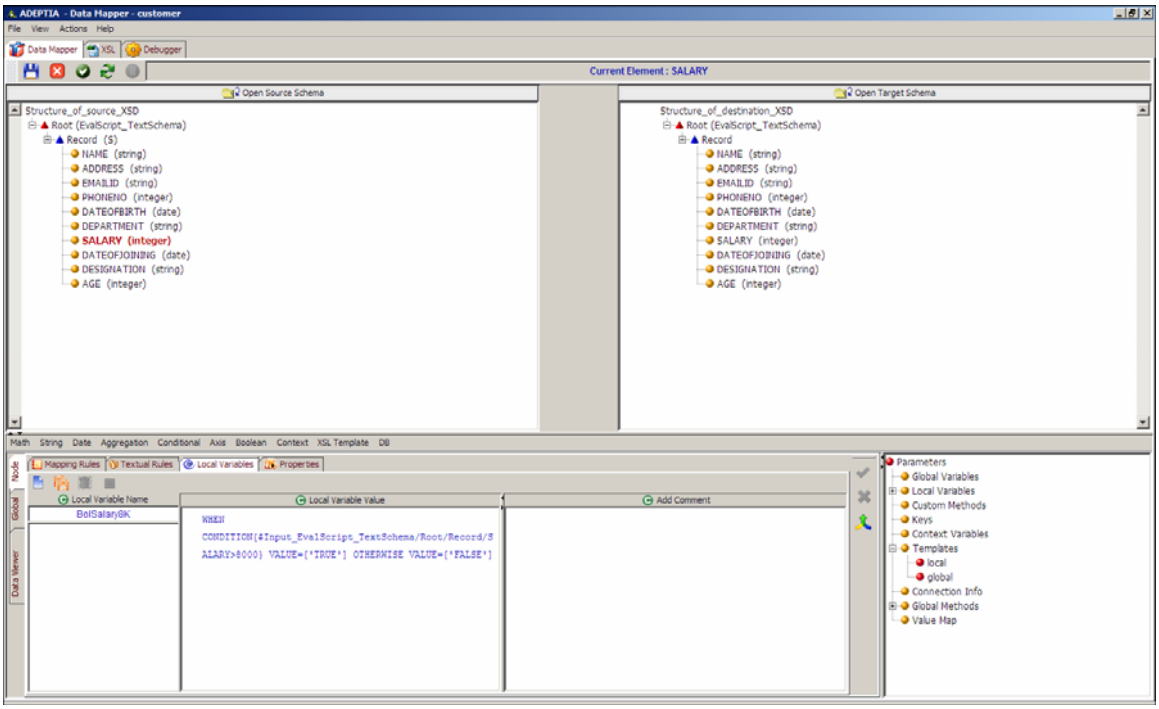


Figure 10.80: Create Local Variable

7. Click **Save Local Variable** ( ) button to save the declared local variable.

Steps to map Set-Context function using local variable

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
3. Click **Local Variables** tab. The Local Variable pane is displayed. (refer to Figure 10.105).
4. Click any *target* element to declare the local variable in the Local Variable pane.
5. Click *Local Variable Name* text field and enter the name of the local variable you want to create (e.g. Var1). Press the **Tab** or **Enter** key. This takes the control to the *Local Variable Value* field.
6. Click **Context** mapping function menu and select **Set-Context** sub-function. Syntax for Set-Context function is displayed in *Local Variable Value* field.
7. Enter the process flow variable name you want to create (e.g. *ContextVar*) in the first argument of the *Set-Context* function.
8. Enter the value of the variable in the second argument of the *Set-Context* function (see Figure 10.81).

	Second argument can be string constant, integer constant, XPath, local variable or output of limited set of mapping function, which can be used before Set-Context function. In this example, *BolSalary8K* variable is used as second argument of set-context function.

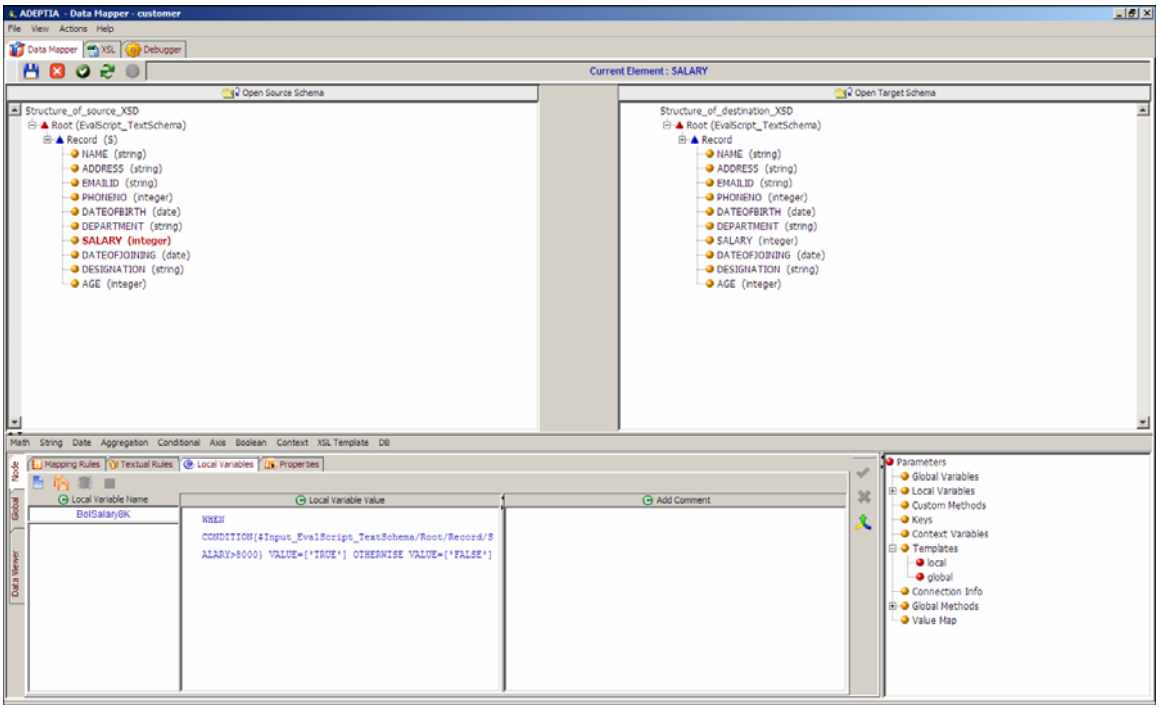


Figure 10.81: Mapping Set-Context to Local Variable

9. Click **Save Local Variable** ( ) button to save the declared local variable.

Global Declaration of Set-Context Function

Concept

Another way of using the Set-Context function is to declare it globally. When doing this, you need not map it to a target element or local variable.

Advantages

- Easy to use
- Need not be mapped to a target element
- Need not create any local variable

Disadvantages

- Local variable cannot be used in value part of Set-Context function
- If XPath of a source path is used, Set-Context function is executed for the first record only.

Declare Set-Context Function Globally

Steps to declare Set-Context function globally

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Global** tab displayed in the Mapping Graph Area. All tabs of the Global tab are displayed.
3. Click **Context Variables** tab. The Context Variables pane is displayed (see Figure 10.82).

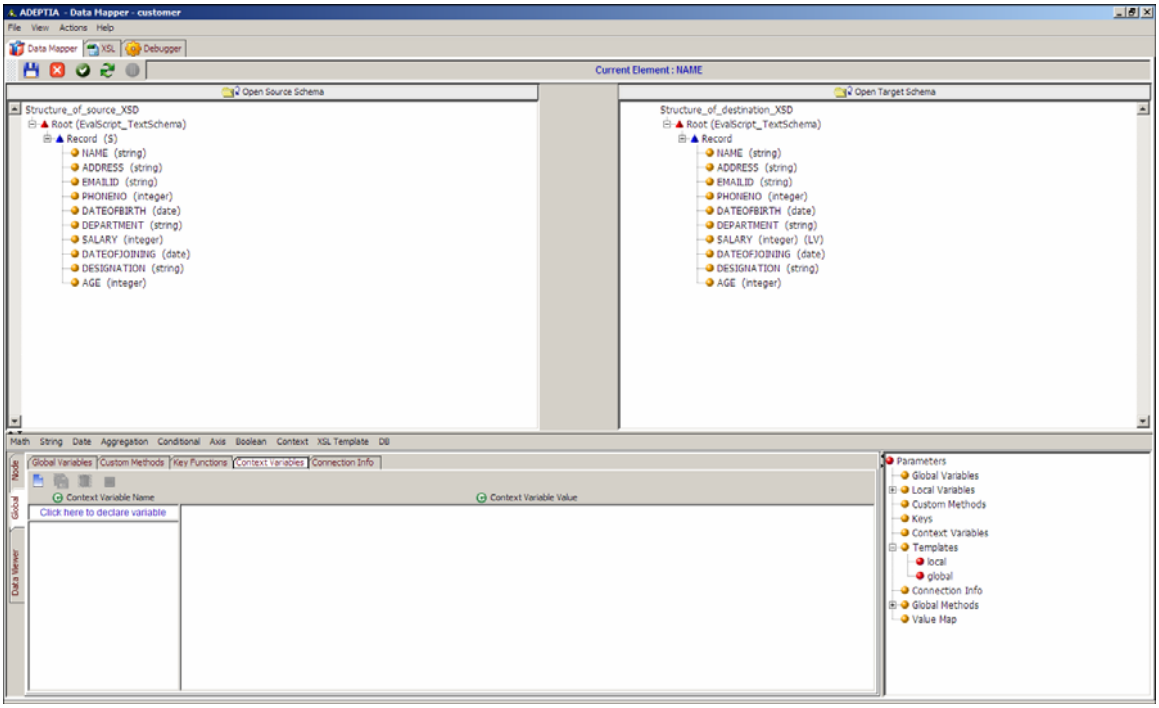


Figure 10.82: Context Variables Pane

4. Click *Context Variable Name* text field and enter the name of process flow context variable you want to create (e.g. *Var1*) Press the **Tab** or **Enter** key. This takes the control to the *Context Variable Value* field.
5. Enter the value of process flow context variable in *Context Variable Value* field (see Figure 10.83).

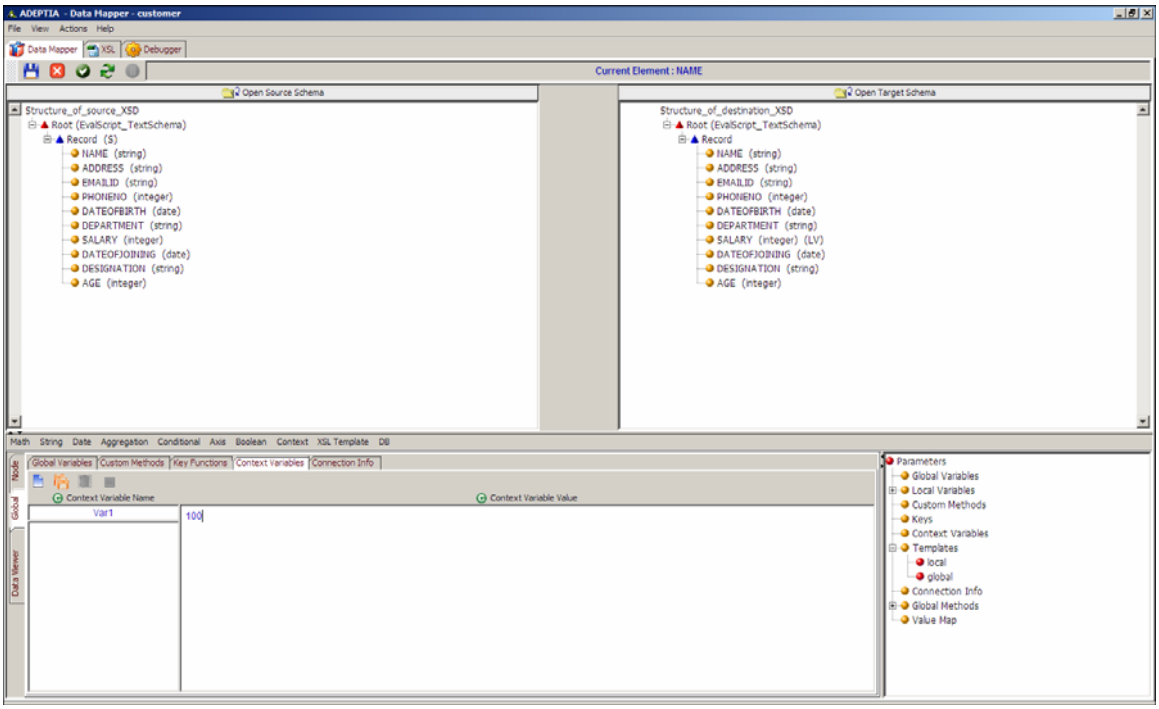


Figure 10.83: Enter Context Variable Value

6. Click the **Save Context Variable** ( ) button to save the declared process flow context variable. This context variable is added to the list of existing context variables in the Context Variables tab. It is also displayed under Context Variables in the Parameters Panel. If you shift focus to another node, or click any of the Mapping Rules, Textual Rules, Node, Properties, *XSL* or *Debugger* tabs, without saving the context variable, an alert message is displayed (refer to Figure 10.100).

7. Click **No** to save the context variable and shift focus to the other object. If you click **Yes**, then the defined context variable is cleared and the focus is shifted to the other object.

	You can also set the value of a context variable that is used in a particular condition. For example, if the context variable *Var1* is used in an IFF condition in such a way the value of Var1 depends on the condition being True or False. If it is True then *Var1* will have a different value. You can set this value of Var1 if the condition is True, using the Set Context function. Similarly, you can retrieve the value of a context variable used in a particular condition, if the condition is True.

	You can edit or delete a context variable from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

Mapping Set-Context function to any target element

Concept

You can map the Set-Context function to a target element. Since this function does not return a value, it will create an empty tag of that target element in output XML. Thus, it is recommended that you should map the Set-Context function to a target element, whose value is not required at the target end.

Advantages

- Need not to create any local variable
- Executed for each record

Disadvantages

- Need an additional target element, whose value is not required at the target end

Map Set-Context Function to a Target Element

Steps to map Set-Context function to a target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Select the target element that you want to map with Set-Context function.
3. Click **Context** mapping function menu and select **Set-Context** sub-function. A set-context node is displayed in the Mapping Graph Area (see Figure 10.84).

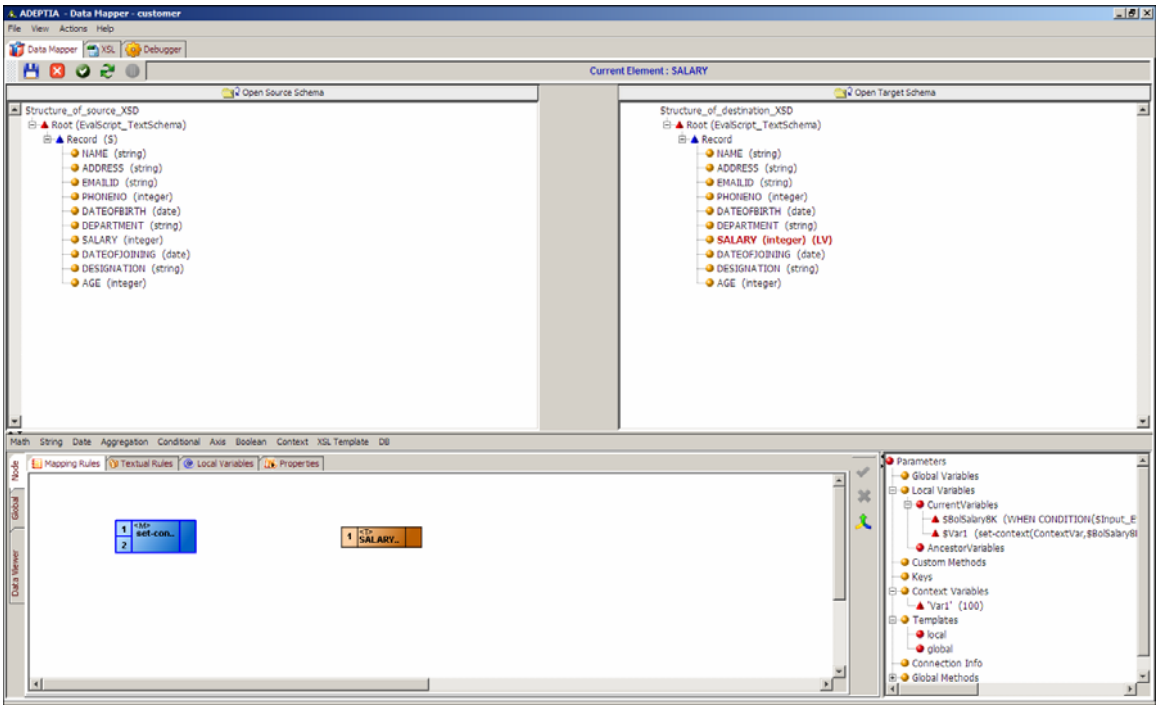


Figure 10.84: Set-Context Node

4. **Add a constant node** with the name of variable (e.g. *index*), whose value is to be set in the process flow context.
5. Create a link from the output of the *constant* node (*index*) to the first input of the *set-context* node.
6. **Add another constant node** for the second argument (e.g. 100) of the *set-context* function.

	Second argument can be string constant, integer constant, XPath, local variable or output of limited set of mapping function, which can be used before Set-Context function.

7. Create a link from the output of second *constant* node (*100*) to the second input of the *set-context* node.
8. Create a link from the output of the *set-context* node to the *target* element (see Figure 10.85).

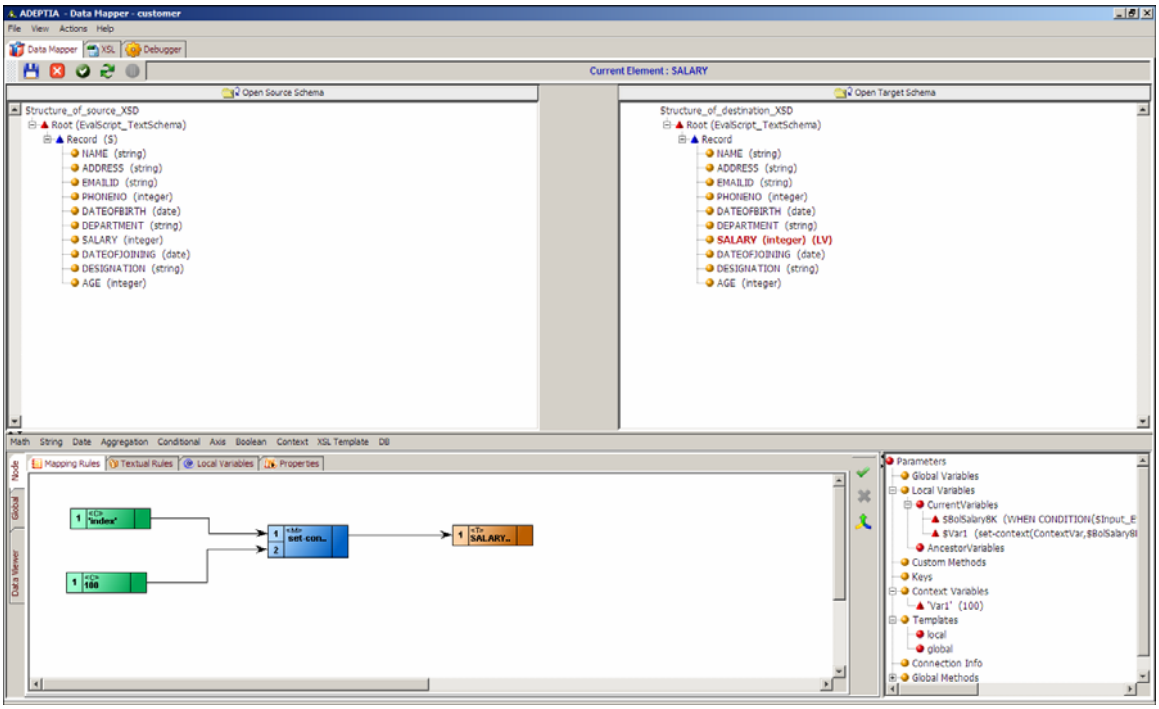


Figure 10.85: Create Links between Nodes for Mapping Set-Context Function to Target Element

9. Click **Apply Mapping** ( ) button. This maps the Set-Context function to the target element. The defined variable (*index*) is passed to the process flow with the value 100. If you shift focus to another node, or click any of the *Mapping Rules*, *Textual Rules*, *Node*, *Properties*, *XSL* or *Debugger* tabs, without applying the mapping, an alert message is displayed (refer to Figure 10.100).

10. Click **No** to apply the mapping and shift focus to the other object. If you click **Yes**, then the mapping activity is cleared and is replaced by the previous mapping in the Mapping Graph Area and the focus is shifted to the other object.

	At times the mapping is too large and the applet memory may exceed 256M. In such cases, the mapping will fail and return an error of insufficient memory. You can overcome this situation by increasing the applet memory. For details, refer to *Appendix A* in the *Administrator Guide*.

Using DB Function

The DB function enables you to get data from the database, based on a condition. It comprises of a sub-function, which is listed in the table below.

Table 10.21: Sub-Function of DB Mapping Function

Mapping Function	Sub Functions	Description	Example

DB	DBQuery	Returns data from the database, based on a query.	*DBQuery('select EmpID from EMP where EMPID=1035', var1, 'true')* returns all the records from the EMP table, where EMPID is 1035. Here, 'var1' is the Connection Info variable. If you change the last boolean argument from *true* to *false*, only the first record is returned.

You can use this sub-function of DB mapping function to get data from the database. The process of using this sub-function is outlined below.

Steps to get data from the database using the 'DBQuery' DB Mapping Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that a *Connection Info* variable is created for the DBQuery and is listed in the Parameters Panel.

For details on creating a Connection Info variable, refer to the **Declaring Connection Info Variable** section.

3. Double-click the target element that you want to map with the DBQuery. It is displayed in the Mapping Graph Area.
4. Click **DB** function and select the **DBQuery** sub-function. A DBQuery node is displayed in the Mapping Graph Area (see Figure 10.86).

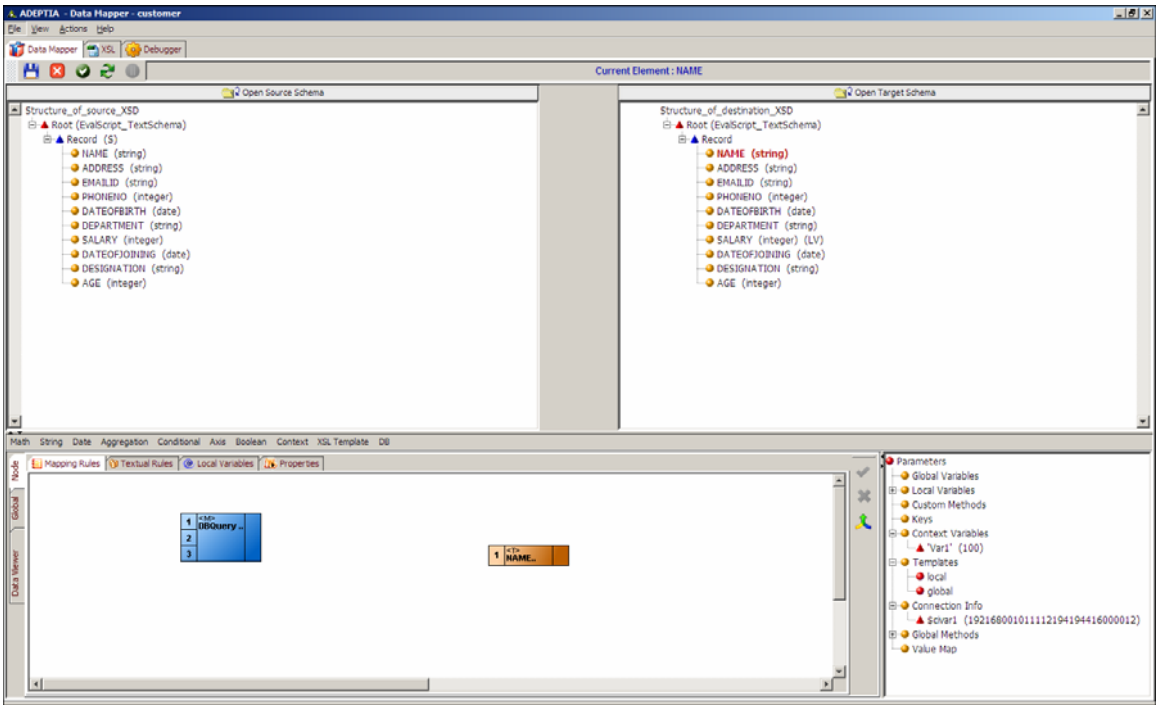


Figure 10.86: Add DBQuery Sub-Function Node

5. **Add a constant node** with the query for searching data from the database based on the specified condition (e.g. *select Name from EMP where designation= 'Manager'*).
6. Create a link from the output of the *constant* node to the first input of the *DBQuery* node.
7. Double-click the required *Connection Info* variable under Connection Info in the Parameters Panel. The selected Connection Info variable node is displayed in the Mapping Graph Area (see Figure 10.87).

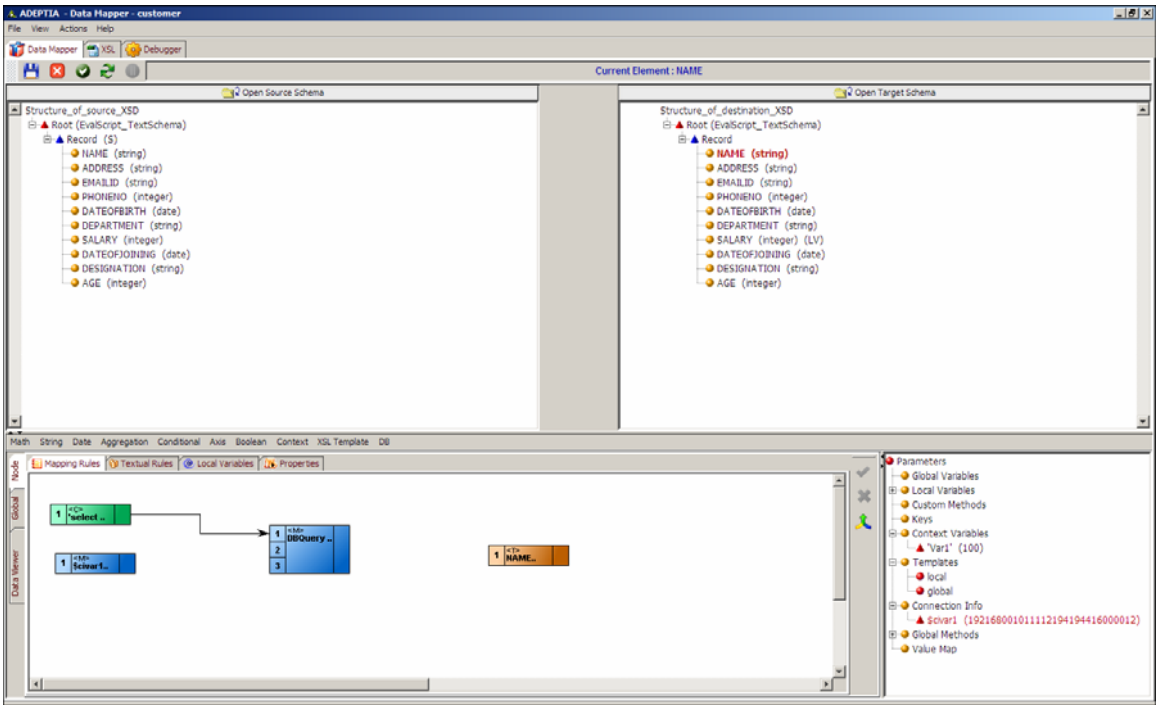


Figure 10.87: Connection Info Variable in Mapping Graph Area

8. Create a link from the output of the *Connection Info* variable to the second input of the *DBQuery* node.
9. **Add another constant node** for the third argument of the *DBQuery* function. This is a Boolean value as True or False.
10. Create a link from the output of this *constant* node to the third input of the *DBQuery* node.
11. Create a link from the output of the *DBQuery* node to the input of the *target* element (see Figure 10.88).

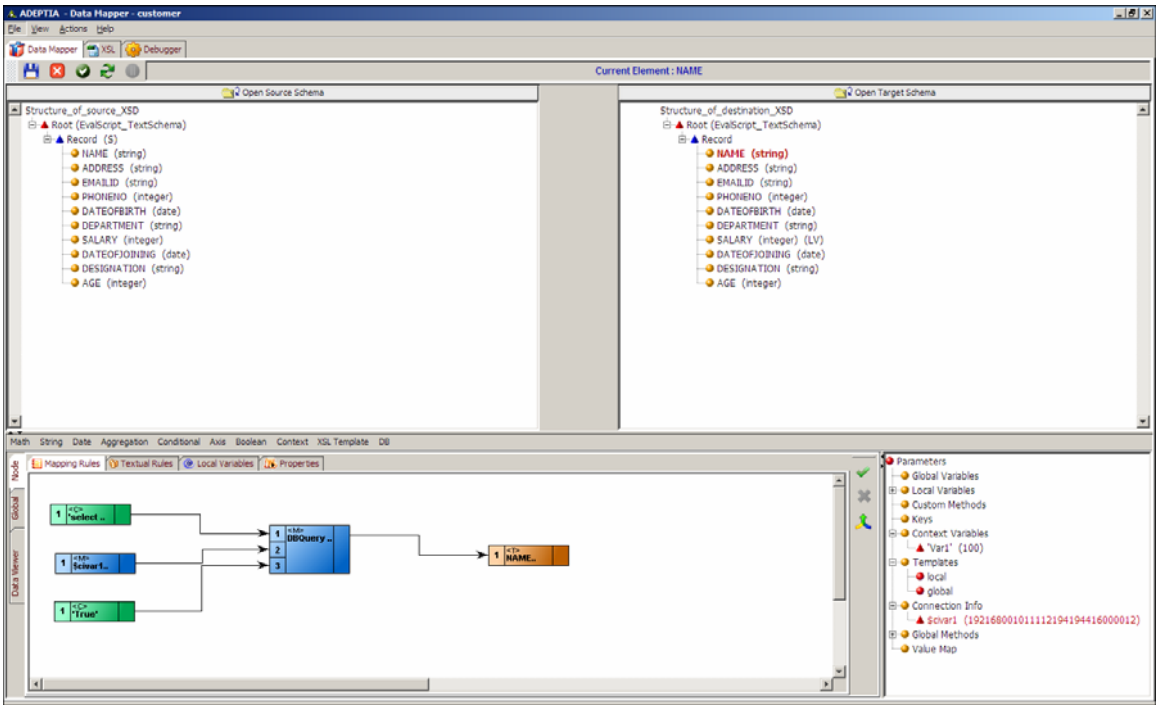


Figure 10.88: Create Links between Nodes for Mapping DBQuery Function to Target Element

12. Click **Apply Mapping** ( ) button. This maps the DBQuery function to the target element. The defined query is passed to the database where a connection is created using the Connection Info variable. If the Boolean value is set as 'True' then the database fetches all the values that match the query. If the value is 'False', then the database fetches only the first matching value. If you shift focus to another node, or click any of the *Mapping Rules*, *Textual Rules*, *Node*, *Properties*, *XSL* or *Debugger* tabs, without applying the mapping, an alert message is displayed (refer to Figure 10.100).

13. Click **No** to apply the mapping and shift focus to the other object. If you click **Yes**, then the mapping activity is cleared and is replaced by the previous mapping in the Mapping Graph Area and the focus is shifted to the other object.

	At times the mapping is too large and the applet memory may exceed 256M. In such cases, the mapping will fail and return an error of insufficient memory. You can overcome this situation by increasing the applet memory. For details, refer to *Appendix A* in *Administrator Guide*.

Using Textual Rules

Textual Rules are used for mapping source elements, mapping functions or constants to target elements, by defining the elements in syntax form.

Steps to map elements using the Concat function using Textual Rules

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Select the *target* element that you want to map. It is displayed in the Mapping Graph Area.
3. Click **Textual Rules** tab in the Mapping Graph Area.
4. Click S**tring** mapping function and select **Concat** sub-function. The syntax for the Concat sub-function is displayed in the Mapping Graph Area (see Figure 10.89).

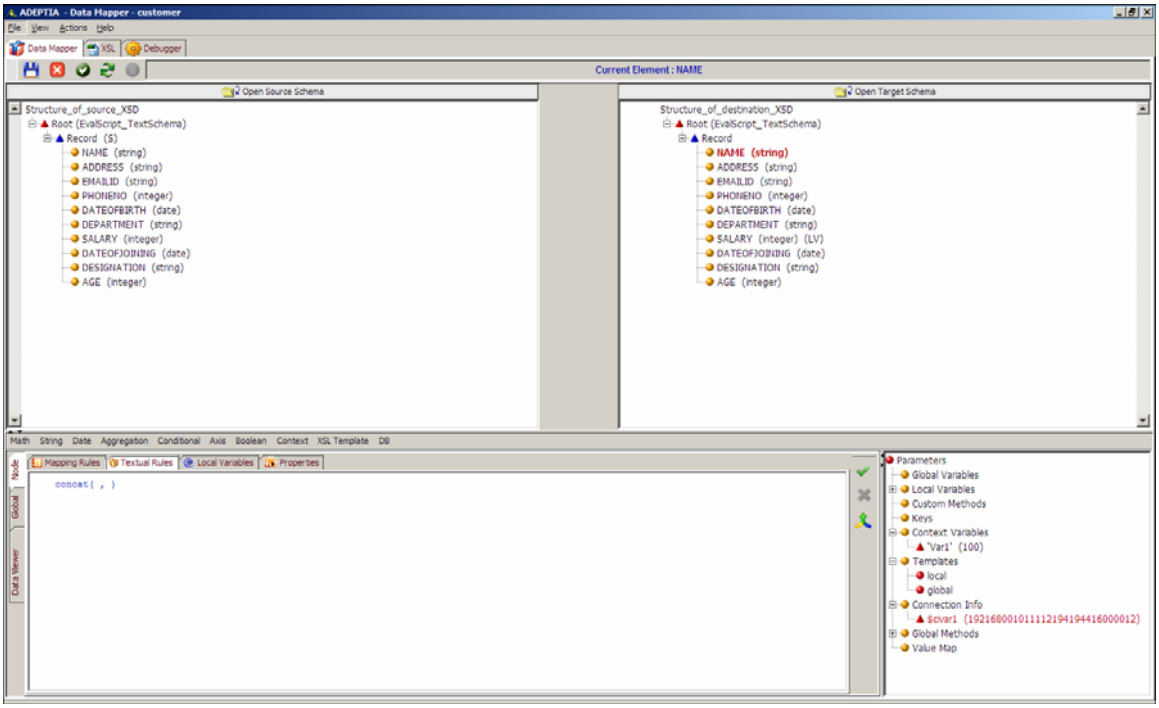


Figure 10.89: Syntax for Concat Sub-Function

5. This syntax has two parameters. Double-click a *source* element that you want to set as the first parameter. This displays the selected source element in the syntax.
6. Similarly, double-click another *source* element as the second parameter. This is displayed in the syntax (see Figure 10.90).

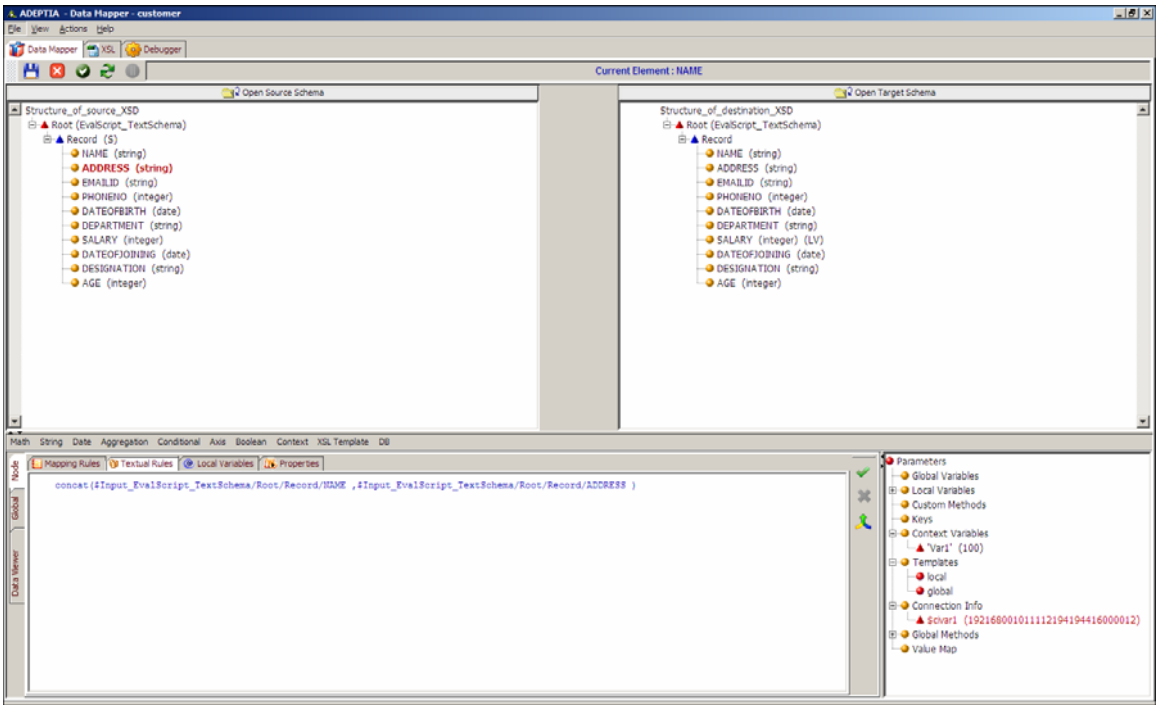


Figure 10.90: Parameters of the Concat Syntax

	You can indent the textual rule content as per your requirement.

7. Click **Apply Mapping** ( ) button. This maps elements using the concat sub-function. A line is displayed between the source and target panels showing the mapping of the source element to the target element.

Using Namespace

Namespace allows you to add, view, edit, and remove the namespace in the XSL. All the Namespaces present in the source and target XSD's are automatically placed in the XSL generated. These Namespaces can be viewed using **View Namespace** option in the **Namespace** submenu.

Adding Namespace

Steps to add a Namespace in the XSL

1. Click **Actions** menu and select **Namespace**.
2. Click the **Namespace** submenu and select **Add Namespace**. The Namespace dialog box is displayed (see Figure 10.91).

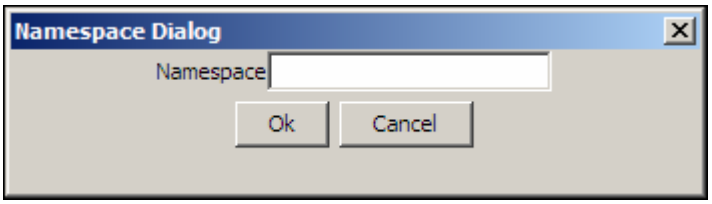


Figure 10.91: Adding Namespace

3. Enter the Namespace and click **OK** button. The Namespace is added to the XSL.

   For Example, *xmlns:oa="http://www.openapplications.org/oagis"*

	The namespace added is shown in the XSL stylesheet tag.

Editing Namespace

Steps to edit a Namespace in the XSL

1. Click **Actions** menu and select **Namespace**.
2. Click the **Namespace** submenu and select **Edit Namespace**. The Select Namespace screen is displayed (see Figure 10.92).

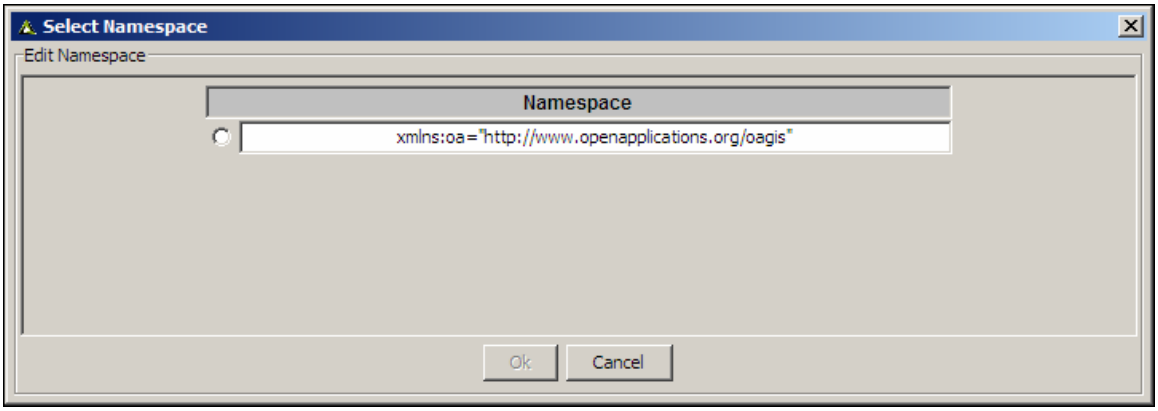


Figure 10.92: Select Namespace

3. Select the namespace you want to edit. This activates the **OK** button. This displays the *Namespace* dialog box in edit mode.
4. Make the necessary changes.
5. Click **OK** to save the modifications.

	Similarly, you can view or remove an existing namespace.

Using XSL Template

An XSL template is a customized template. If you need to repeatedly perform a set of functions, you can create an XSL template defining those functions, and then later call that XSL template whenever required. For example, if you need to perform addition of two numbers at regular intervals, you can define the addition function in an XSL template. You can then call this XSL template whenever required.

The XSL Template option enables you to manage XSL templates. You can add, update or delete an XSL template using this option.

Once you have added an XSL template, you can use it to map elements.

There are two type of XSL Templates:

- **Local:** Local templates are available within the mapping activity, in which you have created the template.

- **Global:** Global templates are available within all the mapping activities. Once you create a global template in a mapping activity, you can use it in all the mapping activities.

Add XSL Template

Steps to add an XSL template

1. Click **XSL Template** option displayed next to mapping functions on the Mapping Functions Panel. Then select **Manage XSL Template** option. The Manage XSL Template screen is displayed (see Figure 10.93).

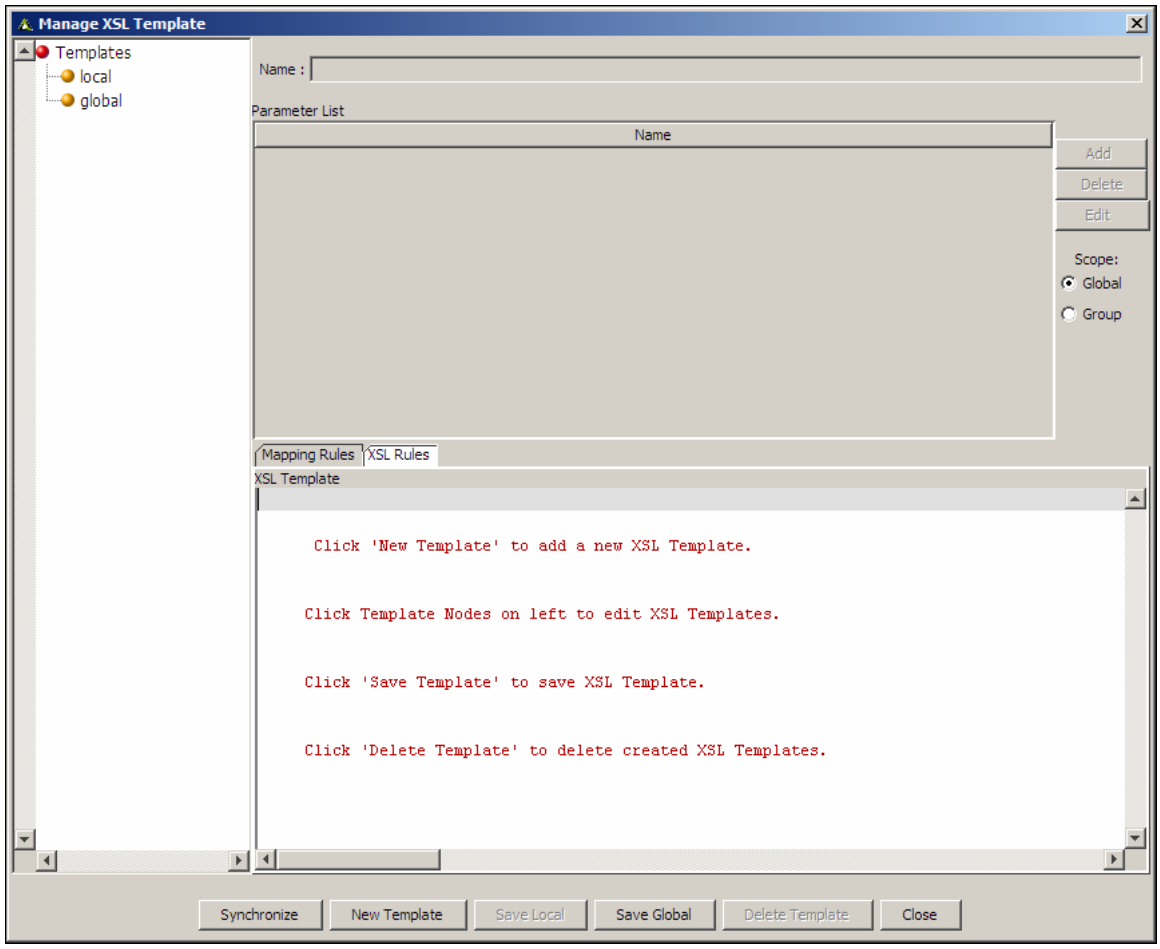


Figure 10.93: Manage XSL Template

2. Click **New Template** button. This activates the *Name* and *Parameter List* fields.
3. Enter the name of the new XSL template that you want to create, in the *Name* field.
4. Click **Add** button to add parameters for the new XSL template. This displays the Parameter Dialog box (see Figure 10.94).

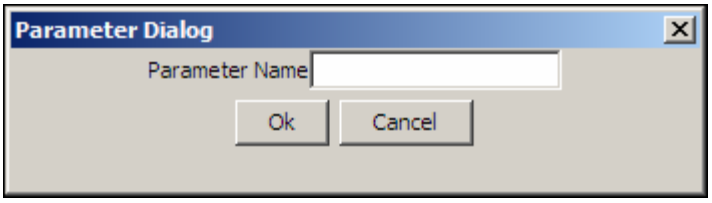


Figure 10.94: Parameter Dialog Box

5. Enter the name of the parameter you want to add, in the *Parameter Name* field and click **OK**. This closes this screen and adds the new parameter in the Parameter List field. It also generates the corresponding XSL in the XSL Template editor on the Manage XSL Template screen (see Figure 10.95).

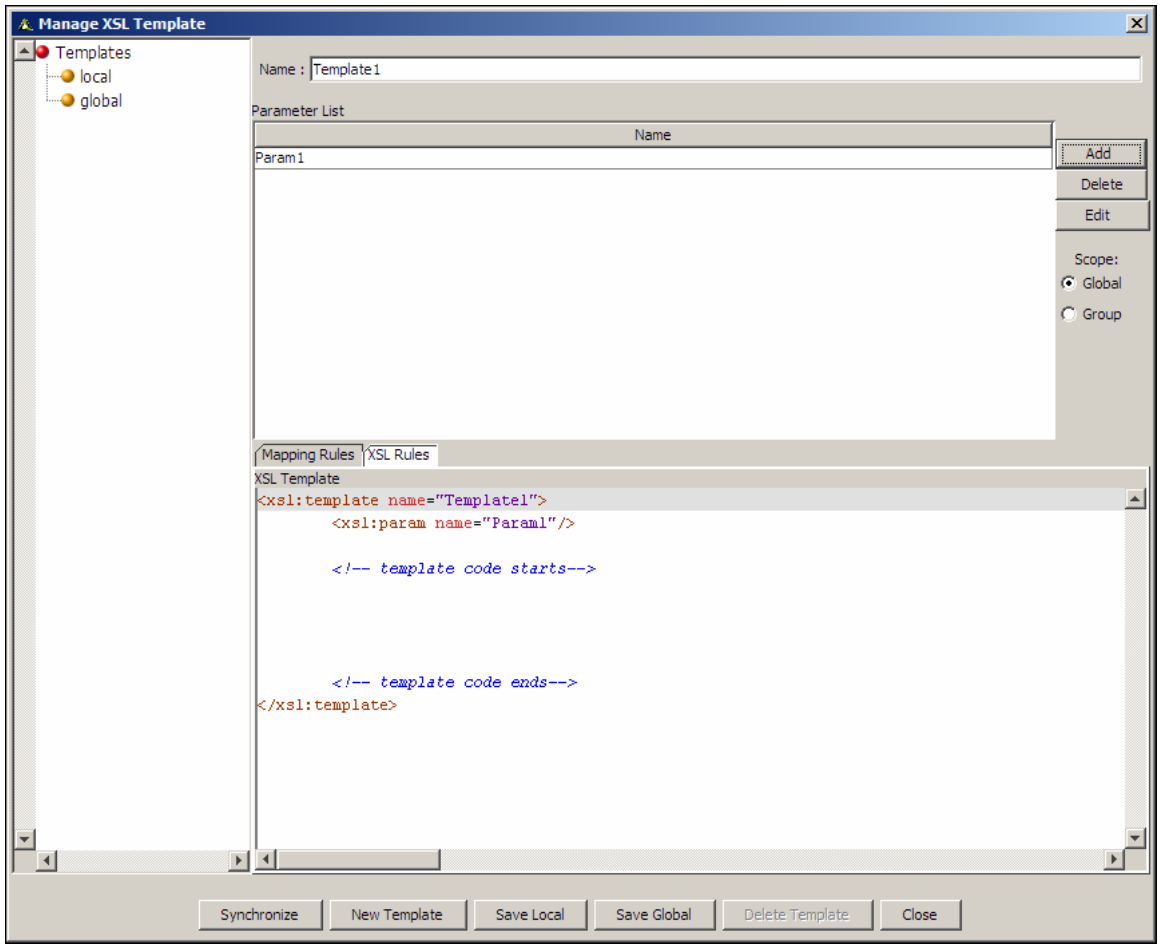


Figure 10.95: Parameters Added in XSL Template

6. Similarly, add additional parameters to the XSL template.

	You can also define variables as the parameters of the XSL Template. To add a global variable, refer to the section **Using Global Variable**.

7. Enter the code for the function that you want to perform, for example, addition of two numbers, in the 'add here' line, in the *XSL Template* field. The syntax of the addition function would be:

   <xsl: value-of select ="$param1 + $param2"/>

   where:

   value-of is the syntax for the value to be returned

   param1 and *param 2* are the defined parameters

8. Once you have entered the code, click **Save Local** button to save the Local XSL template. This displays the saved template under the Templates list on the XSL Template screen. It is also displayed under Templates in the Parameters Panel.

9. If you want to save the template as Global Template, click **Save Global** button. This will save the XSL template as Global XSL template and can be used in any mapping activity.

	On saving the template, XSL validation is performed. If XSL is invalid, then an alert message is displayed specifying the line number and reason for error.

	The name and parameters of the new XSL template also get displayed under the Templates list in the Parameters Panel. You can edit or delete an XSL template from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

	To delete a parameter, click **Delete** button next to the Parameter List field. To delete the active XSL template, click **Delete Template** button. To edit a template, click **template** node of the template you want to edit, under the Templates list. To exit the screen at any time, click **Close** button.

Once you have created a XSL template, you can use it to map elements.

Map Elements using XSL Template

Steps to map elements using an XSL Template

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. Click the XSL template that you want to use for performing the required function, from the *Templates* list in the Parameters Panel. A Call XSL Template node is displayed in the Mapping Graph Area (see Figure 10.96).

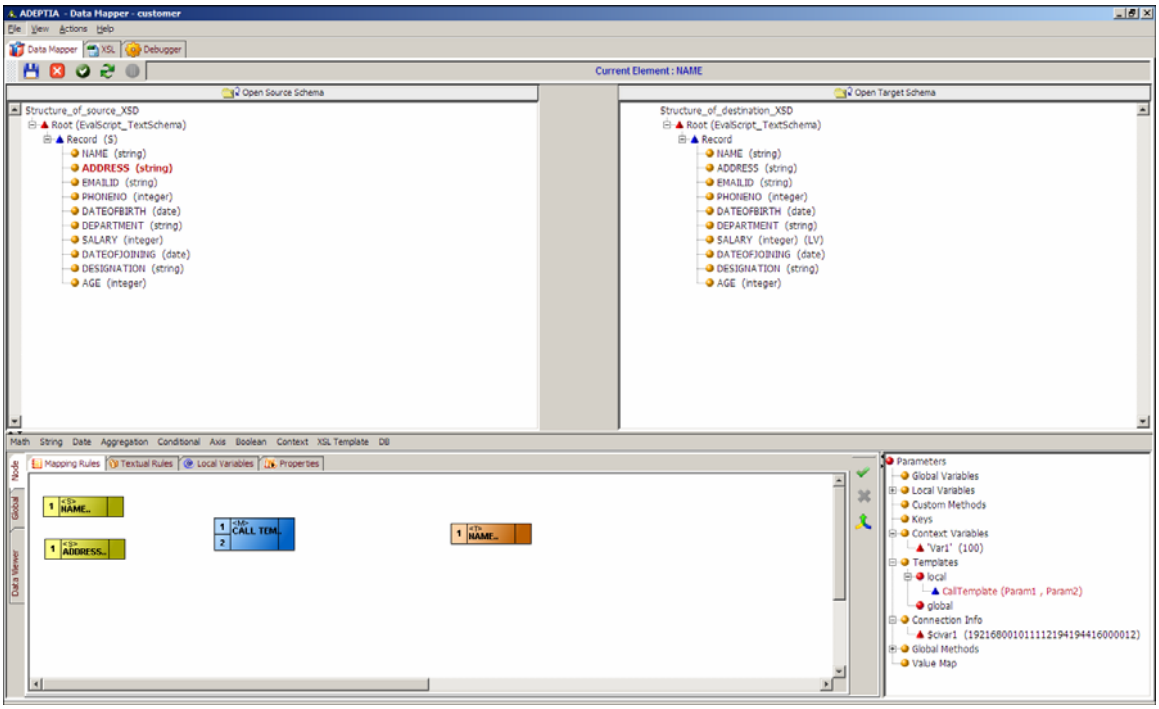


Figure 10.96: Call XSL Template Node

4. Create a link from the output of the first *Source* element to the first input of the *Call XSL Template* node.
5. Create a link from the output of the second *Source* element to the second input of the *Call XSL Template* node.
6. Create a link from the output of the *Call XSL Template* function node to input of the *Target* element node (see Figure 10.97).

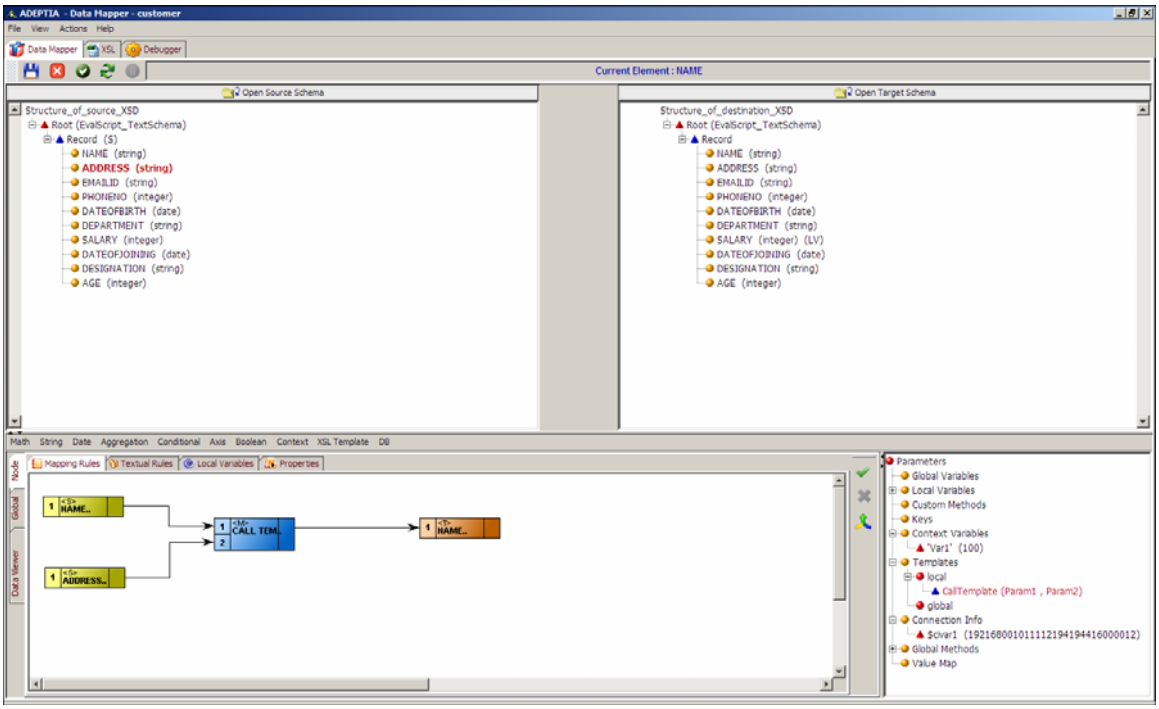


Figure 10.97: Create Links between Nodes for Mapping Elements using XSL Template

7. Click **Apply Mapping** ( ) button. This maps elements using the Call XSL Template function node.

Using Global Variable

Global variables are used for mapping source elements, mapping functions or constants to target elements. They are declared and used globally for all target elements.

To use a global variable in a mapping activity, you need to map it to a target element. This will successfully execute the global variable when you execute the mapping activity in the process flow.

	A global variable differs from a context variable, in the respect that it is available only within the selected mapping activity. Context variables are available in the process designer too.

Declare Global Variable

Steps to declare a Global Variable

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.

2. Click **Global** tab in the Mapping Graph Area. All tabs of the Global tab are displayed.

3. Click **Global Variables** tab. The Global Variable pane is displayed in the Mapping Graph Area (see Figure 10.98).

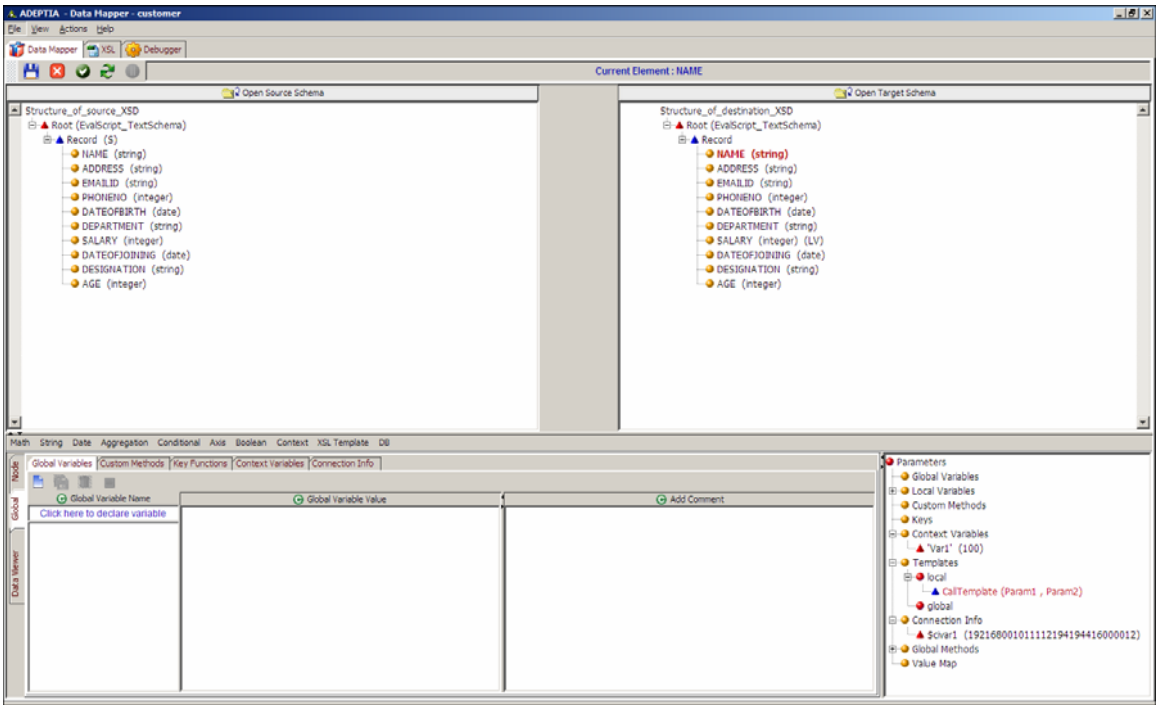


Figure 10.98: Global Variable Pane

4. Click *Global Variable Name* text field and enter the name of the global variable you want to create (e.g. *emp_age*). Press the **Tab** or **Enter** key. This takes the control to the *Global Variable Value* field.

5. *Enter* the value of the global variable in the *Global Variable Value* field. You can enter the value of the global variable using one of the listed methods:

   - Click required source element
   - Select the required Mapping function
   - Type the required value manually
   - Select a Custom Method

	To select a Custom Method as the value of a global variable, double click the desired **Custom Method** under *Custom Methods* in the Parameters Panel. The selected Custom Method is displayed in the Variable Definition field. To learn how to declare a Custom Method, refer to section **Using Custom Method.**

6. Enter comments for the global variable in the *Add Comment* field (see Figure 10.99). For example, if the global variable value contains complex XSLT logic, you can enter its description in this field.

	The *Comments* field is resizable.

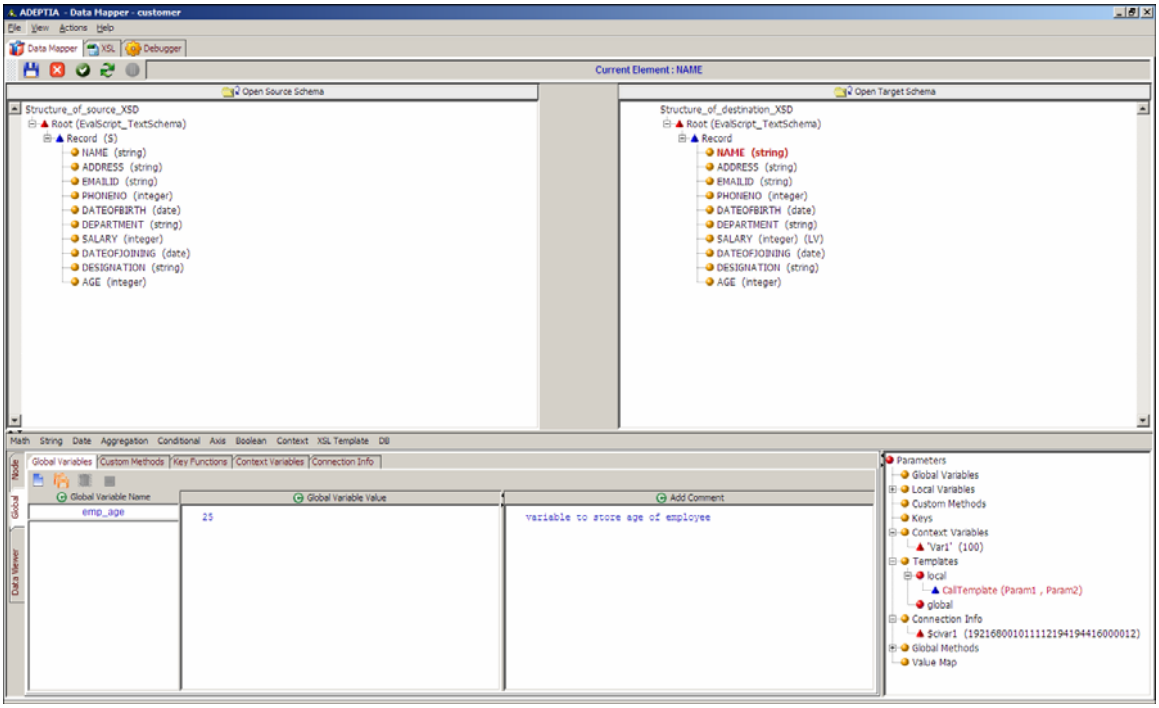


Figure 10.99: Enter Name, Value and Comments of Global Variable

7. Click **Save Global Variable** ( ) button to save the global variable. This global variable is added to the list of existing variables in the *Global Variable Name* field. It is also displayed under *Global Variables* in the Parameters Panel. If you shift the focus to another node, or click any of the *Mapping Rules, Textual Rules, Local Variables, Properties, XSL* or *Debugger* tabs, without saving the global variable, an alert message is displayed. (see Figure 10.100). The comments added for the global variable are saved and displayed above the global variable declaration in the Mapping XSL (refer to Figure 10.101).

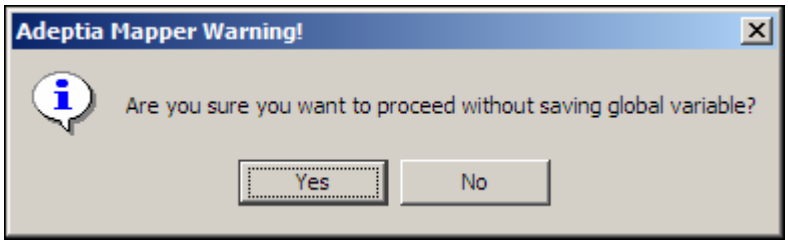


Figure 10.100: Alert Message

	For example, you have defined a global variable *Var1*, and you click the **Name** target element, without saving *Var1*, then the alert message is displayed. The alert message is also displayed when shifting focus between Global and Node tabs.

8. Click **No** to save the global variable and shift the focus to the other object. If you click **Yes**, then the defined global variable is cleared and the focus is shifted to the other object.

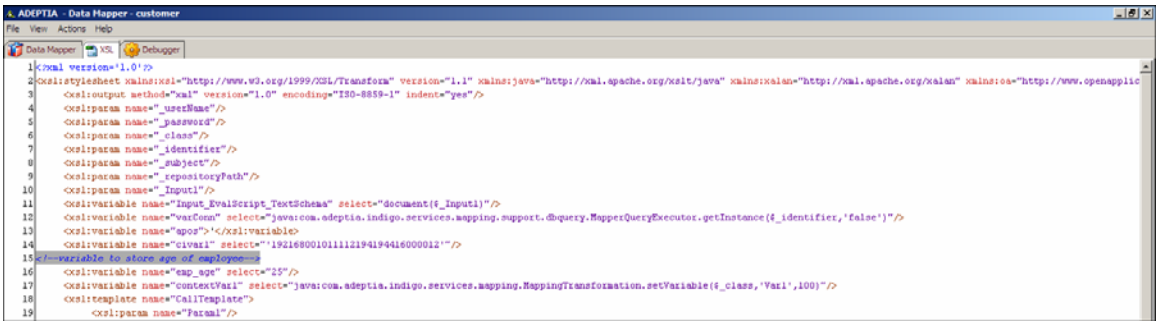


Figure 10.101: Comment in Mapping XSL

	You can rearrange the global variables by dragging it up or down in the list.

	To remove a global variable, select the global variable and click **Remove selected global variable** () button. To remove all global variables, click **Remove all global variables** () button.

Managing a Global Variable from the Parameters Panel

Once the global variable is added to the Parameters Panel, you can edit it or delete it anytime from this Panel itself.

Steps to manage a Global Variable from the Parameters Panel

1. Right-click the **global variable** that you want to edit or delete. This displays the right drop-down menu (see Figure 10.102).

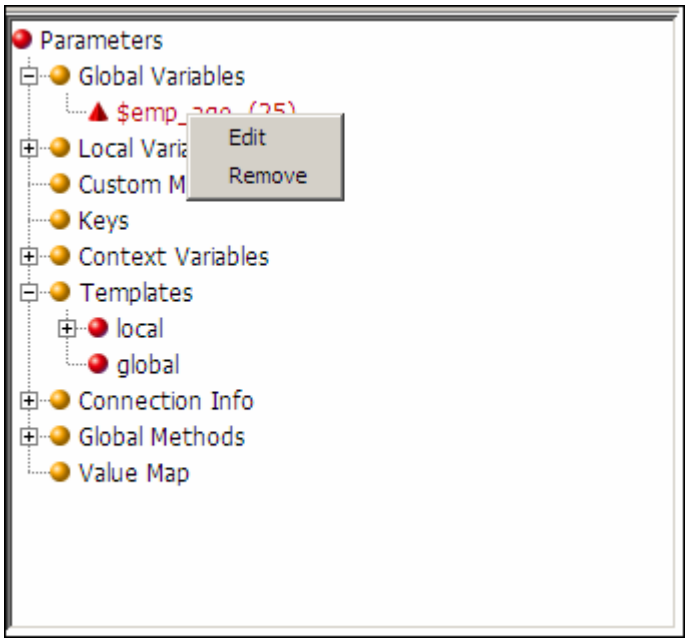


Figure 10.102: Right-Click Menu of Elements in Parameter Panel

2. Select **Edit** to edit the variable. This displays the selected global variable in Edit mode. You can edit the value of the global variable in the *Global Variable Value* field.
3. Alternately, select **Remove** to delete the selected variable. This displays a confirmation screen for the delete operation (see Figure 10.103).

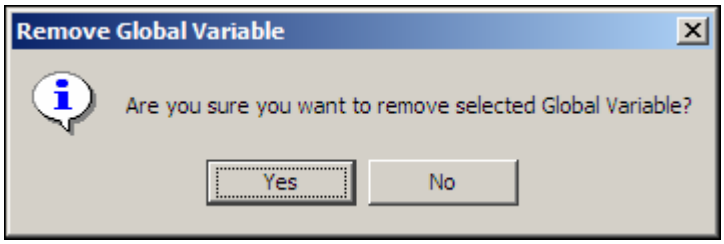


Figure 10.103: Confirm Delete of Global Variable

4. Click **Yes** to delete the selected variable.

Map Global Variable to Target Element

Once you have created a global variable, you can map it to a target element.

Steps to map Global Variable to target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed. Click **Mapping Rules** tab.

3. Select a *target* element. This displays the selected target element node in the Mapping Graph Area.

4. Double-click the required **global variable** under *Global Variables* in the Parameters Panel. The selected global variable node is displayed in the Mapping Graph Area (see Figure 10.104).

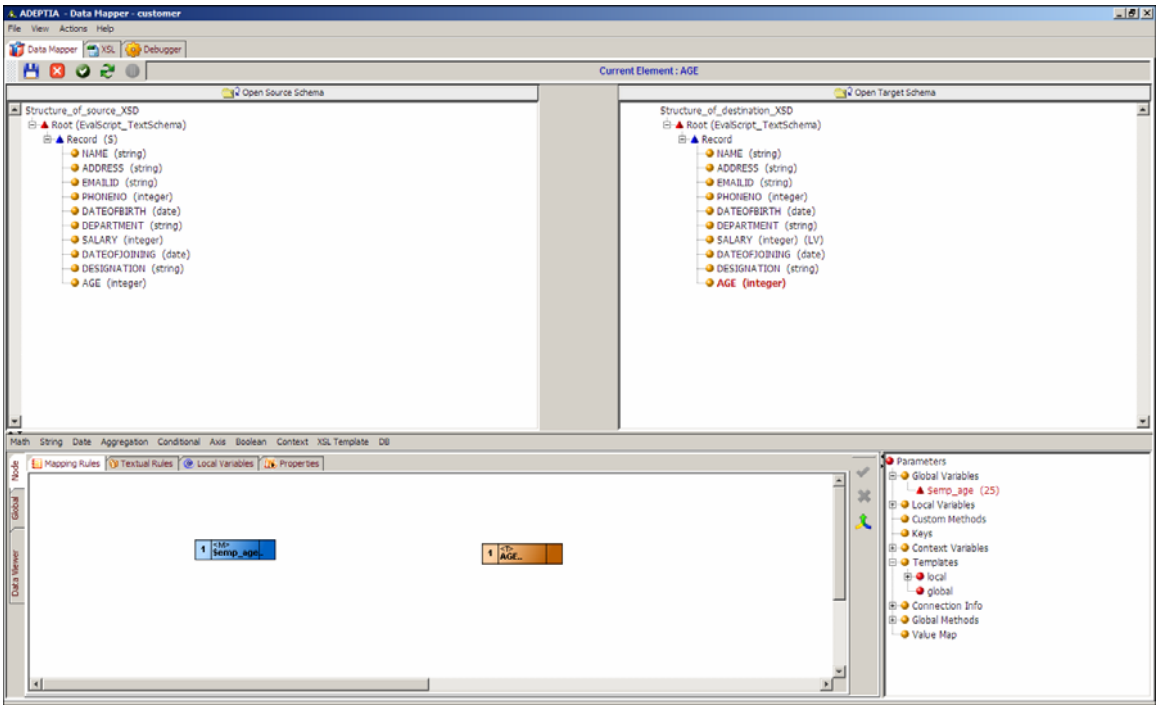


Figure 10.104: Add Global Variable Node

5. Create a link from the output of the global variable to the input of the target element.

6. Click **Apply Mapping** ( ) button. This maps the global variable to the target element. If you shift the focus to another target element, or click any of the *Node, XSL* or *Debugger* tabs, without applying the mapping, then an alert message is displayed. (refer to Figure 10.100).

7. Click **No** to apply the mapping and then shift the focus to the other object. If you click **Yes**, then the mapping activity is cleared and is replaced by the previous mapping in the Mapping Graph Area and the focus is shifted to the other object.

	Once a global variable is mapped to a target element, the letter **(M)** is displayed next to the mapped target element. This signifies that a target element has been mapped. Refer to **Table of Suffixes** for details on suffixes displayed next to a target element.

8. **Save** the mapping activity and exit the Data Mapper.

	You can **view and validate the generated mapping XSL**, **view the target XML** and **view and validate mapping output**, before

	saving the mapping activity.

Using Local Variable

Local variables are used for mapping source elements, mapping functions or constants to a specific target element. They are declared and used only for the specific target element or attribute for which they are declared.

To use a local variable in a mapping activity, you need to map it to the specific target element or attribute. This will successfully execute the local variable when you execute the specific mapping activity in the process flow.

Declare Local Variable

Steps to declare a Local Variable

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
3. Click **Local Variables** tab. The Local Variable pane is displayed in the Mapping Graph Area (see Figure 10.105).

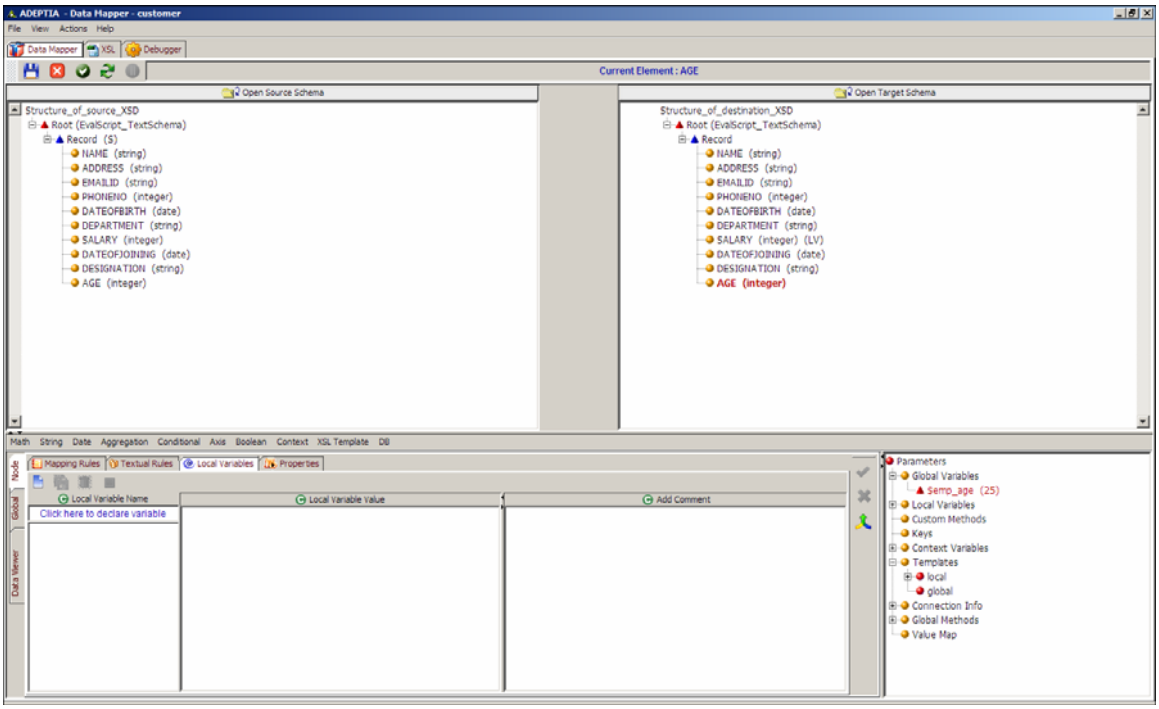


Figure 10.105: Local Variable Pane

4. Click *Local Variable Name* text field and enter the name of the local variable you want to create (e.g. *varQuery*). Press the **Tab** or **Enter** key. This takes the control to the *Local Variable Value* field.
5. Enter the value of the local variable in the *Local Variable Value* field. You can enter the value of the local variable using one of the listed methods:
   - Click required source element
   - Select the required Mapping function
   - Type the required value manually
   - Select a Custom Method
6. Enter comments for the local variable in the *Add Comment* field (see Figure 10.106). For example, if the local variable value contains complex XSLT logic, you can enter its description in this field.

	The *Comments* field is resizable.

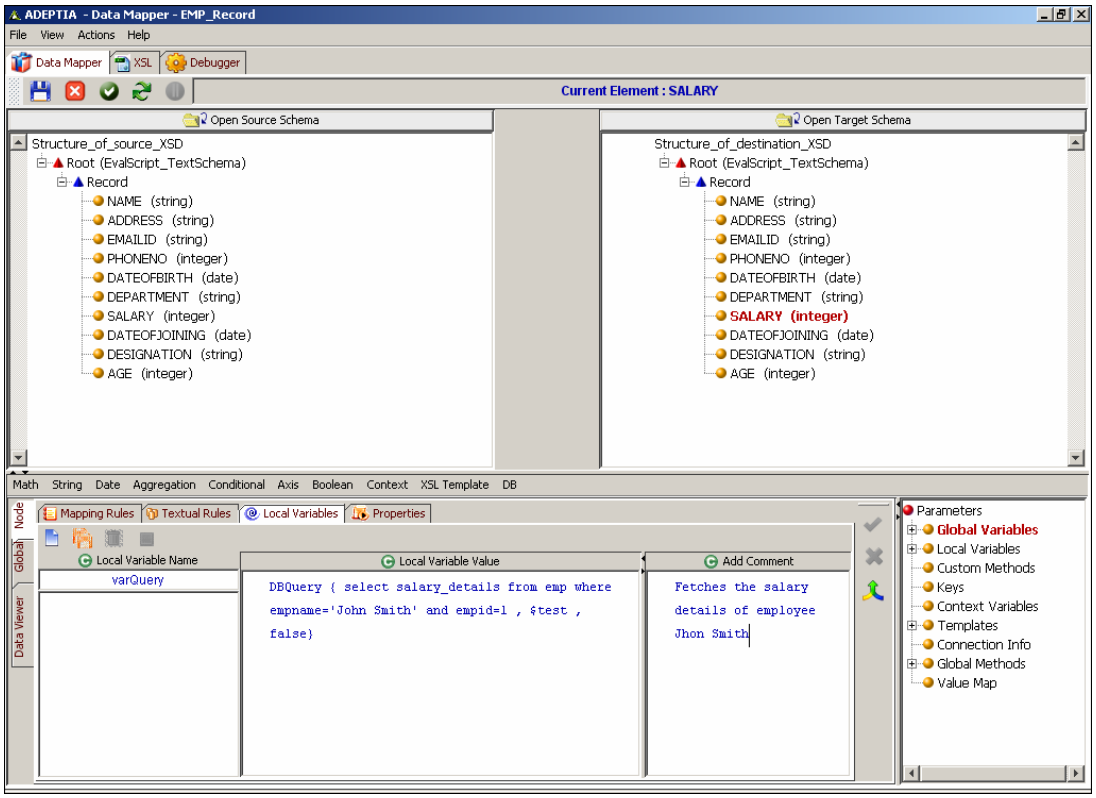


Figure 10.106: Enter Name, Value and Comment for Local Variable

	To select a Custom Method as the value of a local variable, double click the desired **Custom Method** under *Custom Methods* in the Parameters Panel. The selected Custom Method is displayed in the *Variable Definition* field. To learn how to declare a Custom Method, refer to section **Using Custom Method**.

7. Click **Save Local Variable** ( ) button to save the local variable for the selected target element. This local variable is added to the list of existing variables in the *Local Variable Name* field. It is also displayed under *Local Variables* in the Parameters Panel. If you shift the focus to another node, or click any of the *Mapping Rules, Textual Rules, Global Variables, Properties, XSL* or *Debugger* tabs, without saving the local variable, an alert message is displayed (refer to Figure 10.100). The comments added for the local variable are saved and displayed above the local variable declaration in the Mapping XSL (refer to Figure 10.101).

	The alert message is also displayed when shifting focus between Global and Node tabs.

8. Click **No** to save the local variable and shift the focus to the other object. If you click **Yes**, then the defined local variable is cleared and the focus is shifted to the other object.

	Once the local variable is saved, it is categorized as *Current Variable* or *Ancestor Variable* in the Parameters Panel. If it is created for a child node, then it is saved as a *Current Variable*. If it is created for a parent's parent node, then it is saved as an *Ancestor Variable*.

	You can rearrange the local variables by dragging it up or down in the list.

	To remove a local variable, select the local variable and click **Remove selected local variable** ( ) button. To remove all local variables, click **Remove all local variables** ( ) button. Alternately, you can edit or delete a local variable from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

Map Local Variable to Target Element

Once you have created a local variable, you can map it to the specific target element or attribute.

Steps to map Local Variable to specific target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed. Click **Mapping Rules** tab.
3. Select a *target* element. This displays the selected target element node in the Mapping Graph Area.
4. Double-click the required **local variable** under *Local Variables* in the Parameters Panel. The selected local variable node is displayed in the Mapping Graph Area (see Figure 10.107).

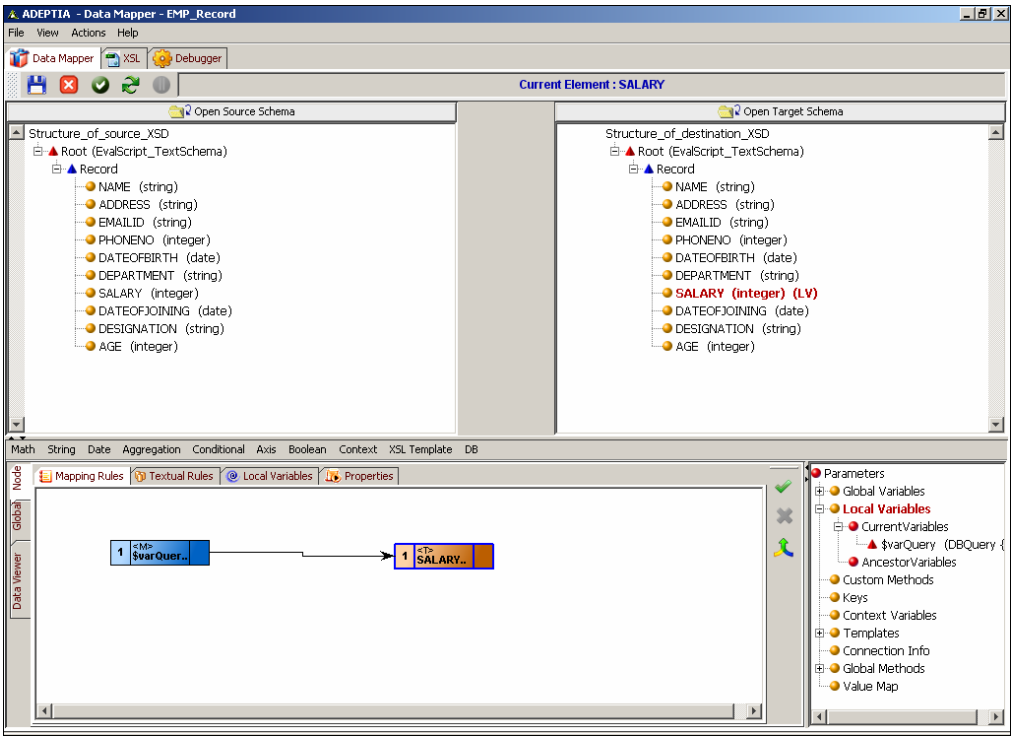


Figure 10.107: Add Local Variable Node

5. Create a link from the output of the *local variable* to the input of the *target* element.

6. Click **Apply Mapping** ( ) button. This maps the local variable to the selected target element. If you shift the focus to another target element, or click any of the *Global, XSL* or *Debugger* tabs, without applying the mapping, then an alert message is displayed (refer to Figure 10.100).

7. Click **No** to apply the mapping and then shift focus to the other object. If you click **Yes**, then the mapping activity is cleared and is replaced by the previous mapping in the Mapping Graph Area and the focus is shifted to the other object.

	Once a local variable is mapped to a target element, the letter **(LV)** is displayed next to the mapped target element. This signifies that a local variable has been mapped to the target element. The letter **(M)** is also displayed indicating the target element has been mapped. Refer to **Table of Suffixes** for details on suffixes displayed next to a target element.

	Local variables do not allow duplicate names at the same level. You need to right-click to edit the local variable, where you can select the source.

8. **Save** the mapping activity and exit the Data Mapper.

	You can **view and validate the generated mapping XSL**, **view the target XML** and **view and validate mapping output**, before saving the mapping activity.

Using Custom Methods

A custom method is used to call a function from a custom Java class in a package. It is declared and used globally for all target elements.

To use a custom method in a mapping activity, you need to map it to a target element. This will successfully execute the custom method when you execute the mapping activity in the process flow.

Prerequisites:

The package containing the custom java class must be copied in the folder *../../AdeptiaServer-5.0/ServerKernel/customClasses.*

Declare a Custom Method

Steps to declare a Custom Method

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Global** tab displayed in the Mapping Graph Area. All tabs of the Global tab are displayed.
3. Click **Custom Methods** tab. The Custom Methods pane is displayed (see Figure 10.108).

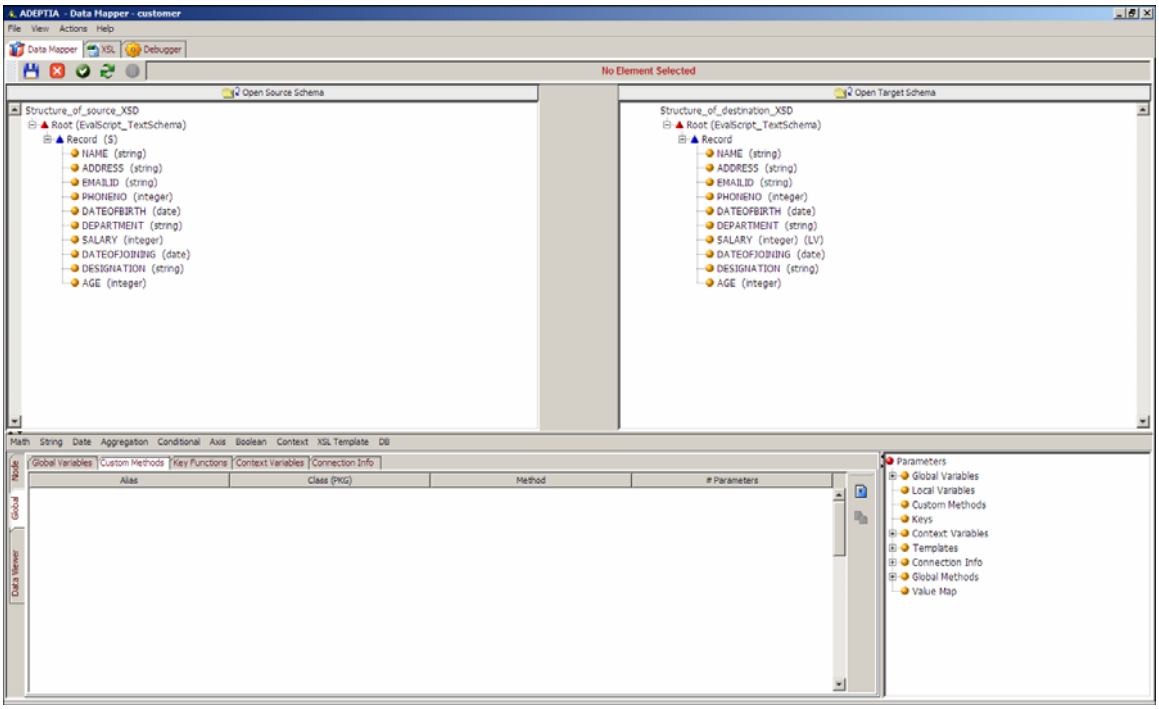


Figure 10.108: Custom Methods Pane

4. Click **Add New Method** ( ) button to add a Custom Method. A row is inserted with the listed columns as displayed in Figure 10.109:

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

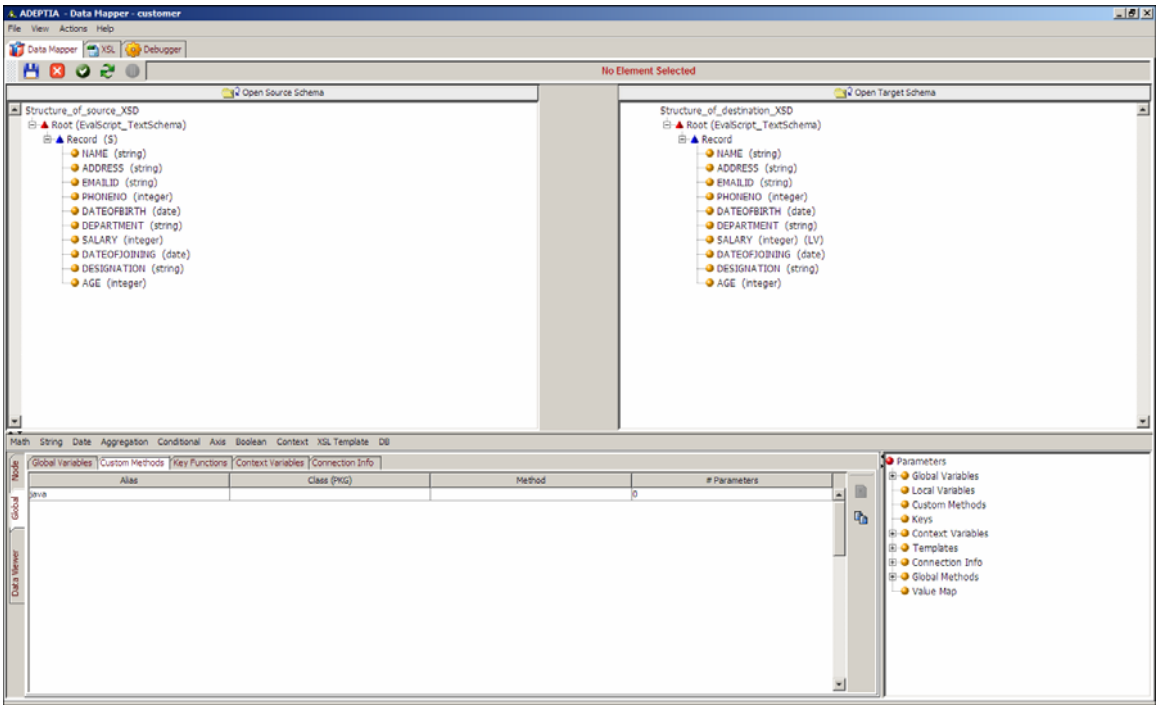


Figure 10.109: Add Custom Method

The value 'java' is automatically displayed in the *Alias* column. This is a read-only field.

5. Click *Alias* field and then press **Enter** or **Tab** key to activate the *Class (PKG)* field.
6. Enter the fully qualified name of the Java class in the *Class (PKG)* field and press **Enter** or **Tab** key to activate the *Method* field.
7. Enter the name of the custom method in the *Method* column and press **Enter** or **Tab** key to activate the *#Parameter* field.
8. Enter the number of arguments taken by Method in the *# Parameter* column. A custom method can take any number of arguments. However, if you enter a value greater than 4, then a warning message appears (see Figure 10.110).

Figure 10.110: Warning for Adding more than 4 Parameters

9. Click **OK** to close the message. This saves the custom method and displays it under *Custom Methods* in the Parameters Panel (see Figure 10.111).

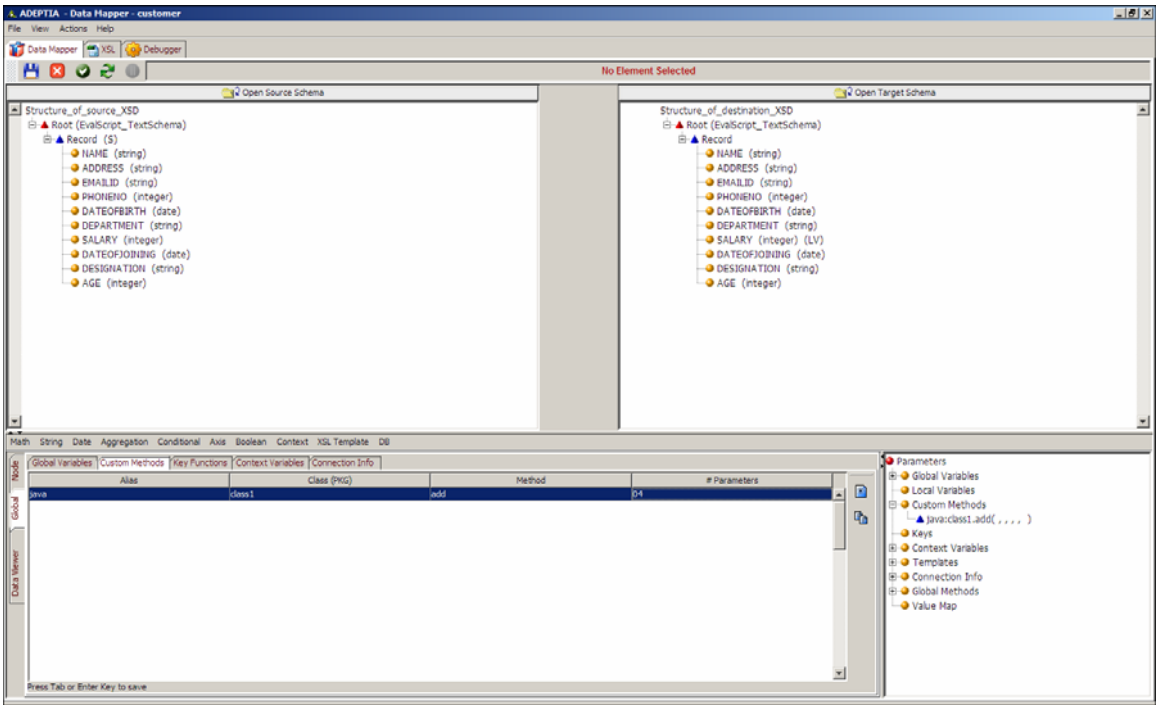


Figure 10.111: Custom Method added to Parameters Panel

	To remove a custom method, select the custom method and click **Remove selected method** () button. Alternately, you can edit or delete a custom method from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

Map Custom Method to Target Element

Once you have created a custom method, you can map it to a target element.

Steps to map Custom Method to target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed. Click **Mapping Rules** tab.
3. Select a *target* element. This displays the selected target element node in the Mapping Graph Area.

4. Double-click the required **custom method** under *Custom Methods* in the Parameters Panel. The selected method node is displayed in the Mapping Graph Area (see Figure 10.112).

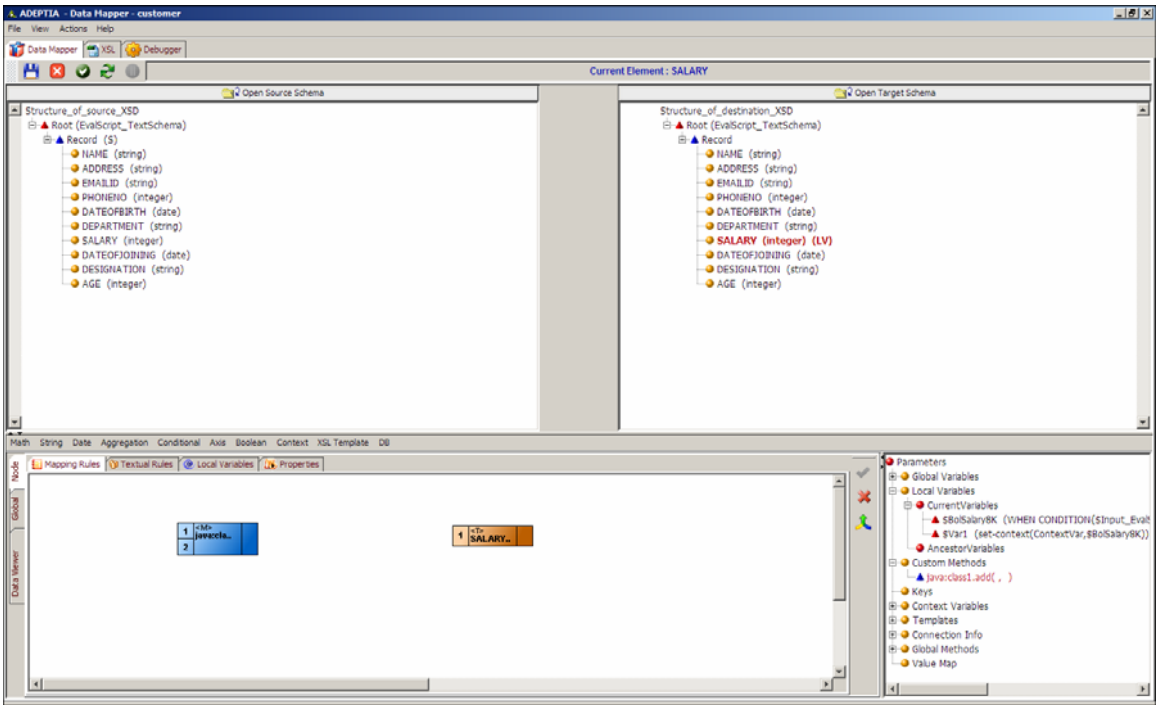


Figure 10.112: Add Custom Method Node

	If the custom method is defined with more than 4 parameters, then double-clicking the custom method will display the warning message (refer to Figure 10.110). Once you click **OK**, you need to define the mapping using Textual Rules.

5. Click two *source* elements. The selected source nodes are displayed in the Mapping Graph Area.
6. Create a link from the output of the first *source* element to the first input of the *custom method* node.
7. Create a link from the output of the second *source* element to the second input of the *custom method* node.
8. Create a link from the output of the *custom method* node to the input of the *target* element (see Figure 10.113).

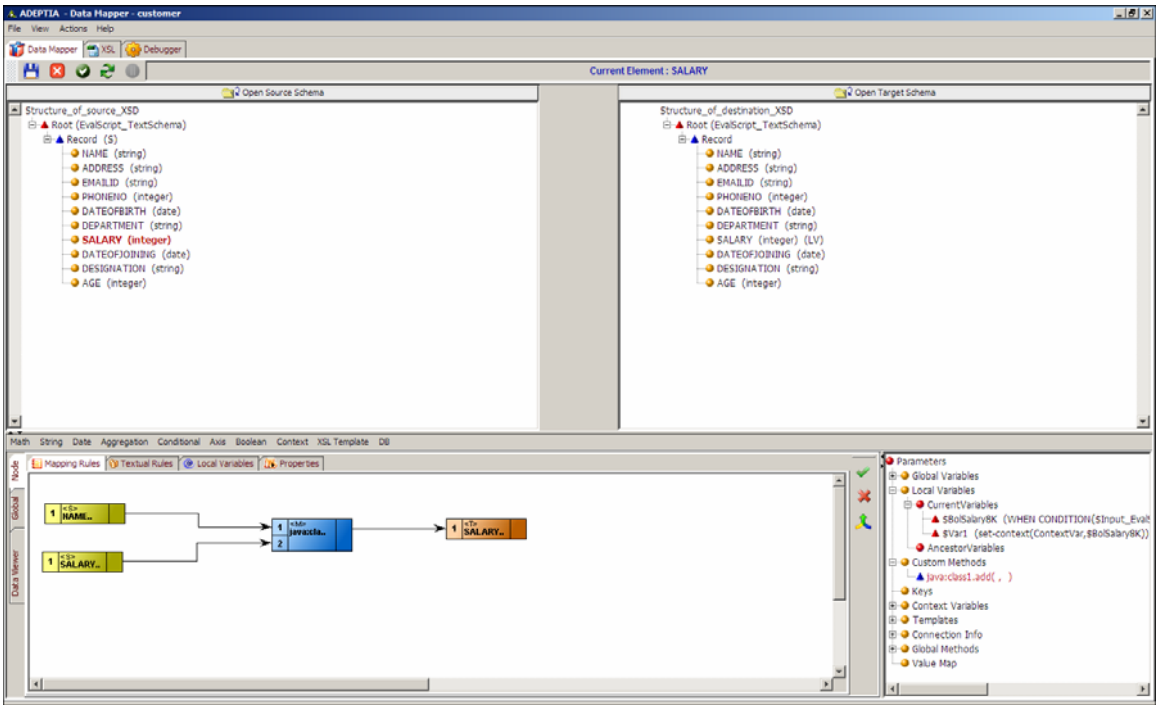


Figure 10.113: Create Links between Nodes

9. Click **Apply Mapping** ( ) button. This maps the custom method to the target element.
10. **Save** the mapping activity and exit the Data Mapper.

	You can **view and validate the generated mapping XSL**, **view the target XML** and **view and validate mapping output**, before saving the mapping activity.

Dynamically Loading Custom Methods

Adeptia supports dynamic loading of custom methods in the Data Mapper. It loads the custom methods available in the Custom Classes folder, under the Global Methods node in the Parameters Panel. This enables you to access any custom method from the Data Mapper itself, without any need to specify method parameters, such as class name, method name and the number of arguments in the Data Mapper applet.

Steps to dynamically load Custom Methods

1. Copy all the class files in the Custom Classes folder and then restart the kernel.

	The current implementation of Global Methods does not support package hierarchy. So all class files that are directly placed in the Custom Classes folder, will be visible through the Global Methods node in the Parameters

	Panel.

2. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes. The *Global Methods* node is already displayed in the Parameters Panel, each time Data Mapper is opened (see Figure 10.114)

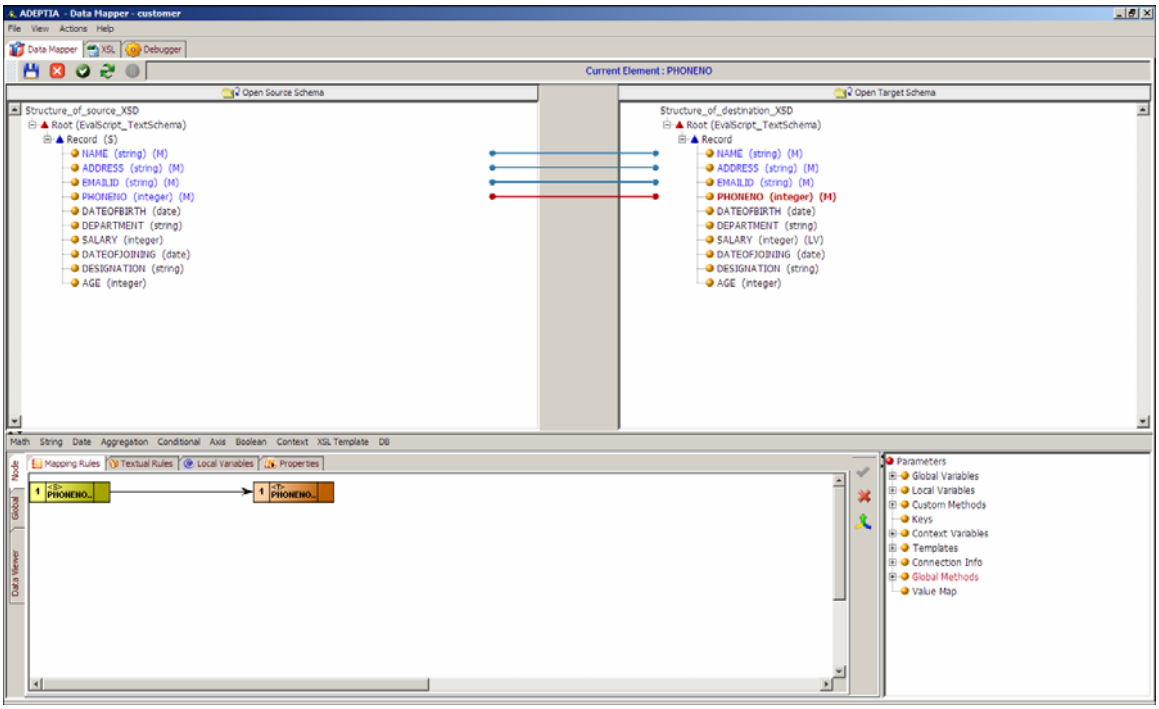


Figure 10.114: Global Methods Node

3. Expand the **Global Methods** node to display all the class files present in the Custom Classes folder (see Figure 10.115).

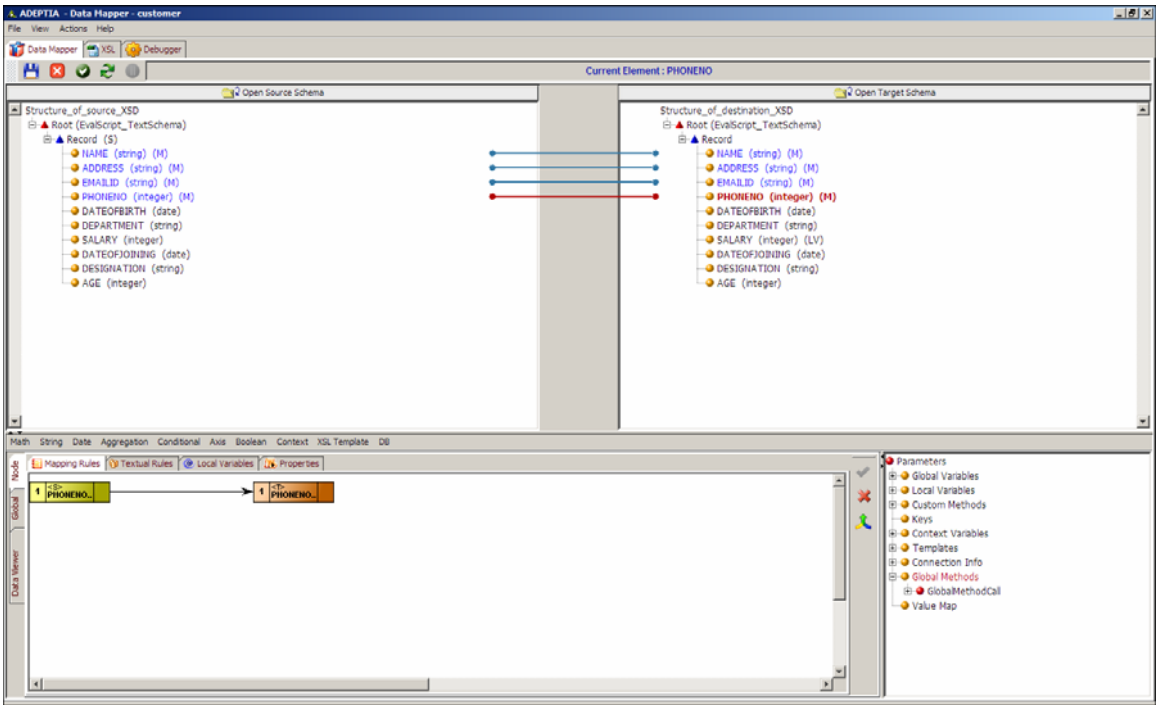


Figure 10.115: List of Class Files in Custom Classes Folder

4. Expand a class node to display all custom methods available for that class (see Figure 10.116).

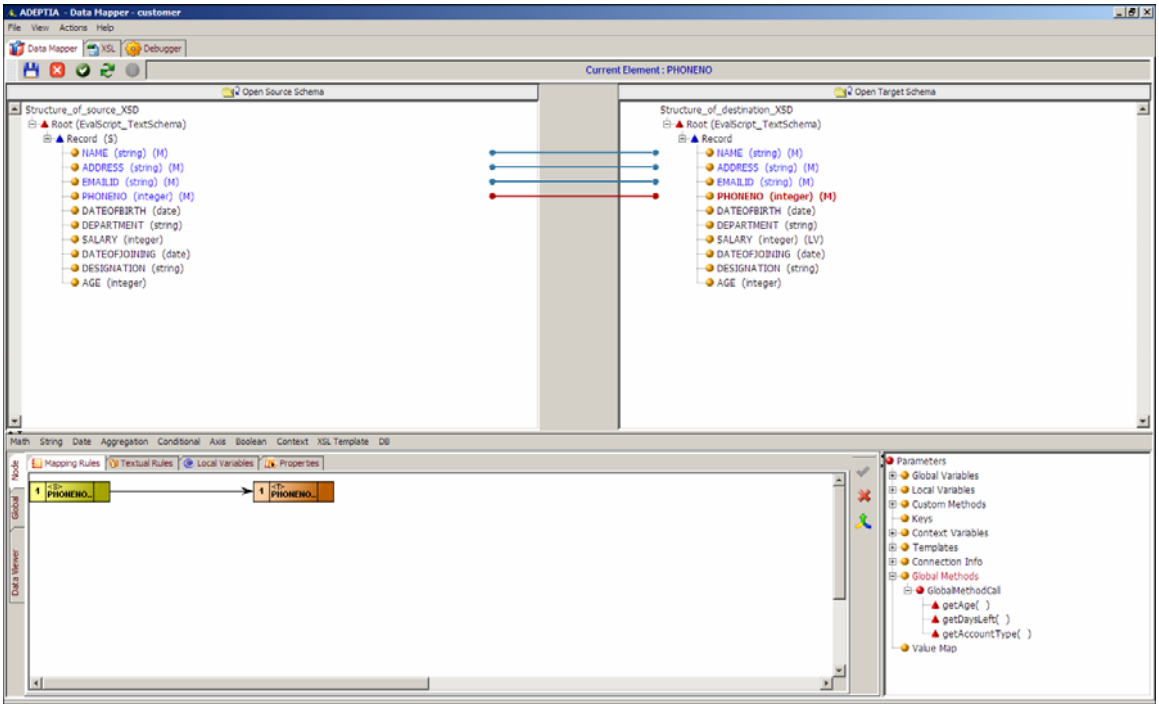


Figure 10.116: Custom Methods in a Custom Class

You can view more information (such as return type and parameter types) about a custom method as a tooltip, by placing the cursor over the custom method node.

You can then map the custom method to a target element by double-clicking the custom method and then mapping by providing the input parameters.

Using Key Function

The *Key* function is used to extract information from two or more related input data sources. For example, if there are two input data sources: *source1* and *source2*, then the *Key* function will extract information from *source2* based on the matching field from *source1*.

Key function is always created in context of *source2* (the data source from where data needs to be fetched based on matching field). All the parameters of the *Key* function correspond to *source2* only. It has the following 3 parameters:

- *Key Name* : It is a name that you want to give to the *Key* function (for example, *Key1*))
- *Key Match* : It is the XPath of *source2* that will be returned as output by the *Key* function . This XPath is assumed to be present in the *source2* . A valid Xpath needs to be specified as this path will be returned as output. A schema name cannot be used here in the Xpath.
- *Key Use*: It is the name of the field from *source2* which will be used to find the matching record from *source2* based on the value passed to *Key* function. For example, if the *Key Match* is */Root/Record* and *Key Use* is *EmpID,* and if we pass value 25 to the *Key* function , then it will start scanning *EmpID* of the *source2* inside each record of *source2*. Whenever EmpID=25, the *Key* function will pass that record (*Root/Record*) as output.

Once the *Key* function is created, you need to pass the value to the *Key* function that will be used to match the *Key Use* defined for *source2* and map the output to the target element. For this, you need to use the *Key* mapping function. The syntax of the Key Mapping is :

key(,)

This function accepts 2 parameters. The first parameter is the name of the *Key* function (*Key1*) to be used and the second parameter is the actual value that will be passed to the *Key* function to perform the match on *source2*. The parameters will be:

key('key1', 25)

Once XSLT transformer finds this mapping it will call the *Key* function *Key1* and pass the value *25* to that function. The *Key* function *Key1* on recieving the mapping will pick this value *25* and start scanning *EmpID* of the *source2* inside each record of *source2*. When ever EmpID= 25, the *Key* function will pass that record (*Root/Record*) as output.

The *Key* function always executes inside current context or nearest *For Each* that is applied to the target element where *Key* function is mapped. So if the *For Each* is applied is for *source2* on the target element, then the *Key* function will use *source2* as the data source on which EmpID

scanning will be performed . So the *For Each* applied, helps the *Key* function to decide which data source to use, to perform the scanning and fetch the output.

Usage Scenario

For example, there are two source schemas: *Schema_EmpDetails* and *Schema_Emp_incrementalDetails*. There is one target schema: *Schema_EmpCompleteDetails*. The objective is to map the fields *EmpName, EmpID ,Age, Salary, City and Email* from *Schema_EmpDetails* and map the field *Address* from *Schema_Emp_incrementalDetails* to the target schema.The *Address* from the *Schema_Emp_incrementalDetails* is fetched using *Key* function and using *EmpID* as *Key Use*.

The parameters are defined as:

- Key Name : Key1
- Key Match : /Root/Record
- Key Use : EmpID

The first *For Each* is applied from record of first Schema on target Record node. This will insure that target data will contain as many records as there are in *Schema_EmpDetails*. Now *Key* mapping is used on the target *Address* element. So the deciding *For Each* (*$Input_Schema_Emp_incrementalDetails/Root*) for *Key* function is applied on this element. The *For Each* applied here is upto *Root* element only not *Record* so that only single *Address* target element is created per record . The Key mapping is defined as:

key('key1',$_varEmpID)/Address

Local variable *_varEmpID* is used as second parameter which is dynamically picking value from the EmpID of the first schema (due to *For Each* applied on target *Record* node). . Once the *Key* function finds the match, it will return the matching record *Root/Record* from the second schema. Since the *Address* element is inside the *Record* element, */Address* is appended to the *Key* mapping, which becomes *Root/Record/Address*.

Using key function involves:

- **Creating a key**
- **Mapping the Key with target element**

Creating a Key

Steps to create a key

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Global** tab displayed in the Mapping Graph Area. All tabs of the Global tab are displayed.

3. Click **Key Functions** tab. A list of existing keys is displayed in the Key Functions pane (see Figure 10.117).

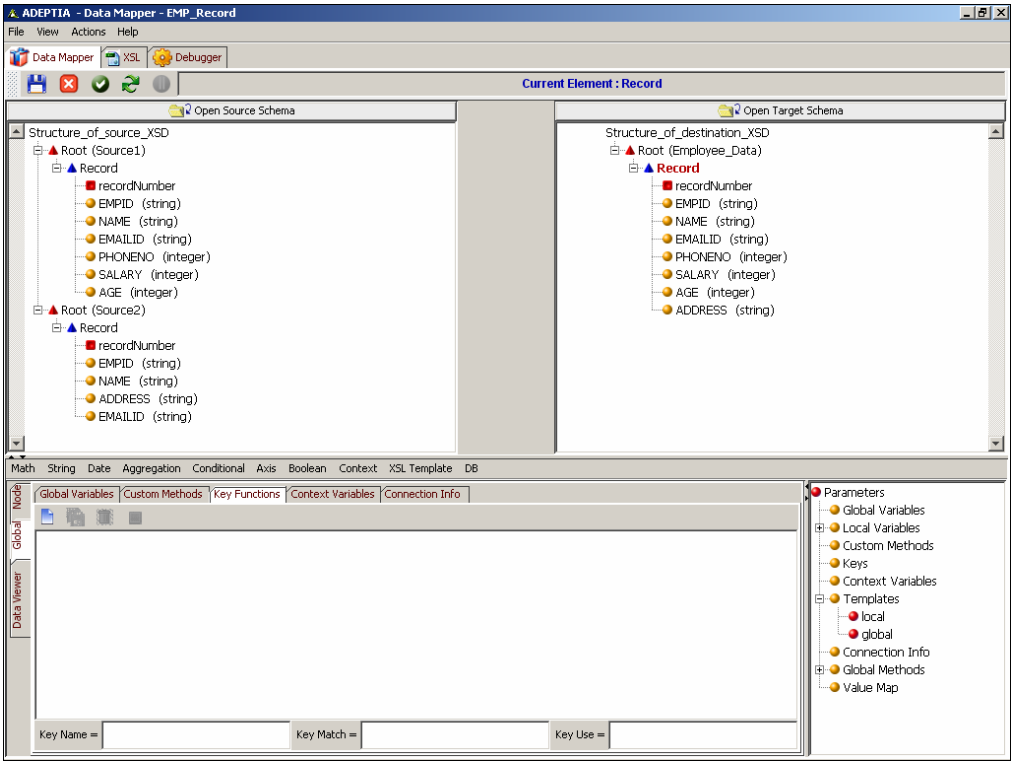


Figure 10.117: Key Functions Tab

4. Click **Add Key** ( ) button to activate the *Key Name* field.
5. Enter the name of the new key in the *Key Name* field.
6. Enter the path of the source element (at the root node) to be matched, in the *Key Match* field. The path always starts with a '/'.

	Here in the field, you need not to enter the complete Xpath. In path you need to enter only the *Root* and *Record.* For example */Root/Record* .

7. Enter the name of the field to be matched in the *Key Use* field For Example *EMPID.*

	The *Key Match* and the *Key Use* fields are case-sensitive.

8. Click **Save Key** ( ) button to save the key. This key is added to the list of existing keys in the **Key Function** tab. It is also displayed under Keys in the Parameters Panel. (see Figure 10.118).

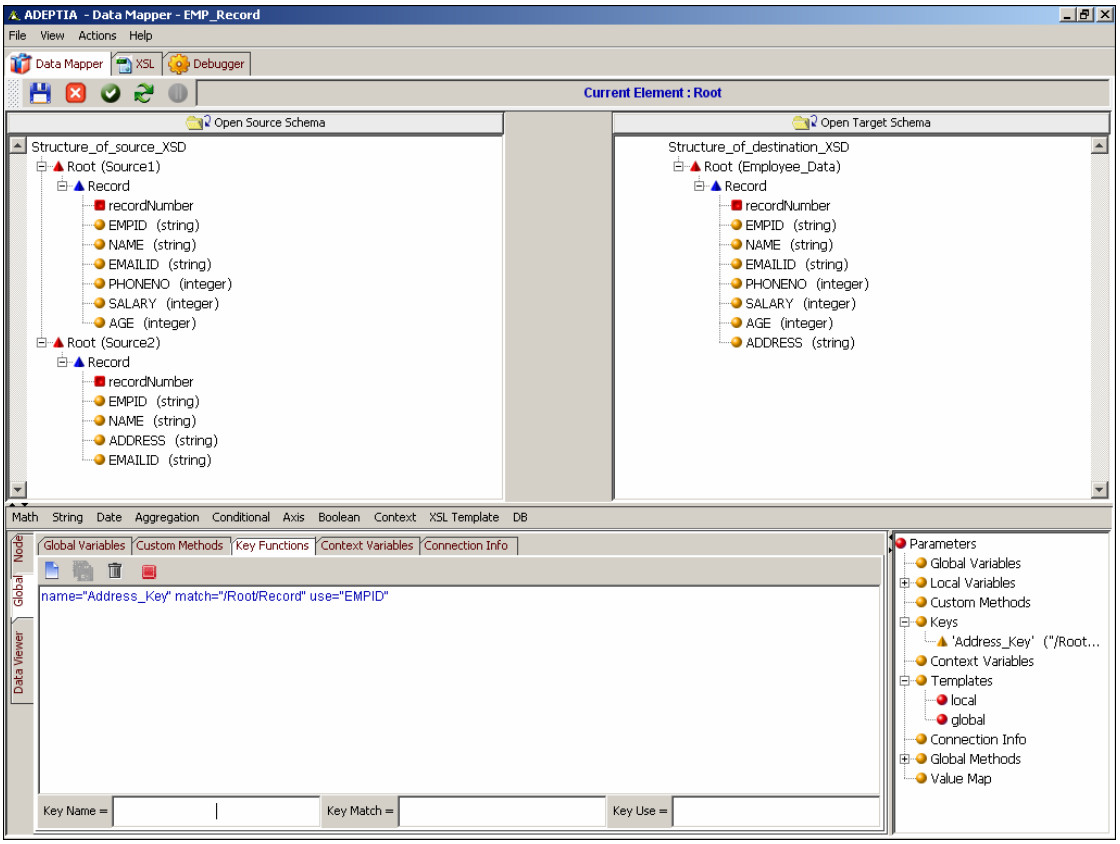


Figure 10.118: Key Added to Parameters Panel

9. If you shift focus to another node, or click any of the *Mapping Rules, Textual Rules, Properties, XSL* or *Debugger* tabs, without saving the key, an alert message is displayed (refer to Figure 10.100).

	The alert message is also displayed when shifting focus between the *Global* and *Node* tabs.

10. Click **No** to save the key and shift the focus to the other object. If you click **Yes**, then the defined key is cleared and focus is shifted to the other object.

	You can rearrange the keys by dragging it up or down in the list.

	You can edit or delete a key from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

Mapping the Key with Target Element

Steps to use the Key Function

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Map source and target elements as displayed in figure below (see Figure 10.119).

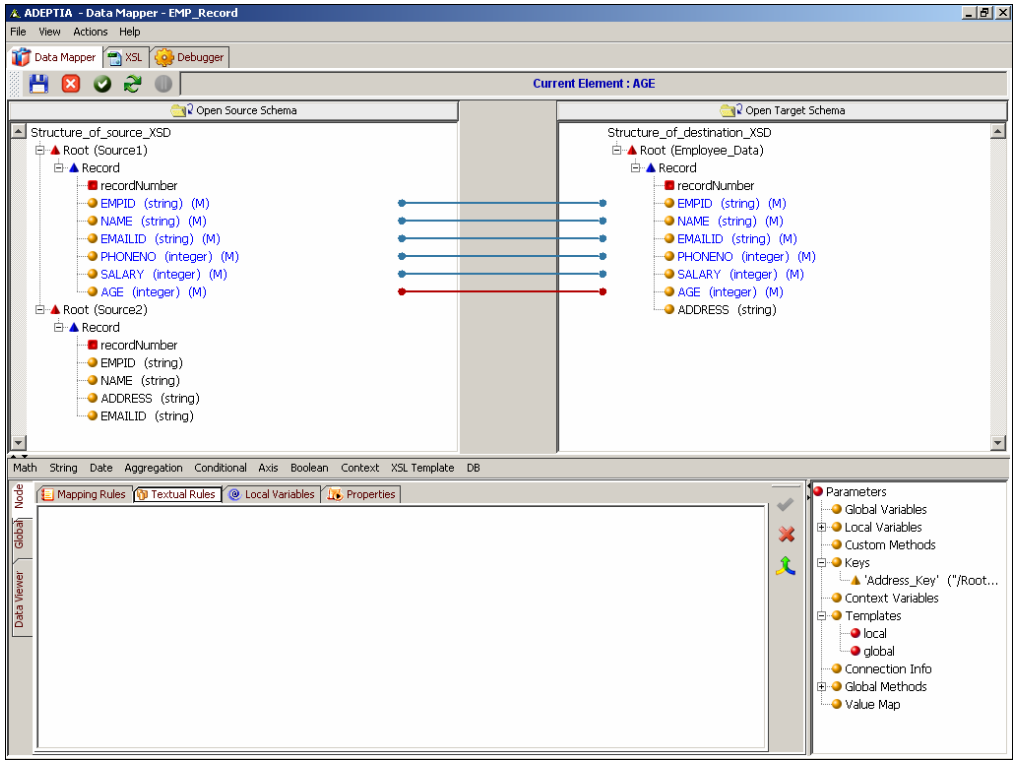


Figure 10.119: Map Source and Target Elements

3. Click the parent node (e.g. *Record*) of the target element and apply the *For Each* for parent node (e.g. *Record*) of schema *Source1*
4. Create a local variable at Record level of the target schema. Select *EMPID* element of Source1 schema as the value of the variable.
5. Click *target* element (e.g. Address) on which you want to apply the key.
6. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
7. Click **Textual Rules** tab. The Textual Rules pane is displayed in the Mapping Graph Area.
8. Click **Aggregation** mapping function and select **Key** sub-function. The Key function is displayed in the Textual Rules pane (see Figure 10.120).

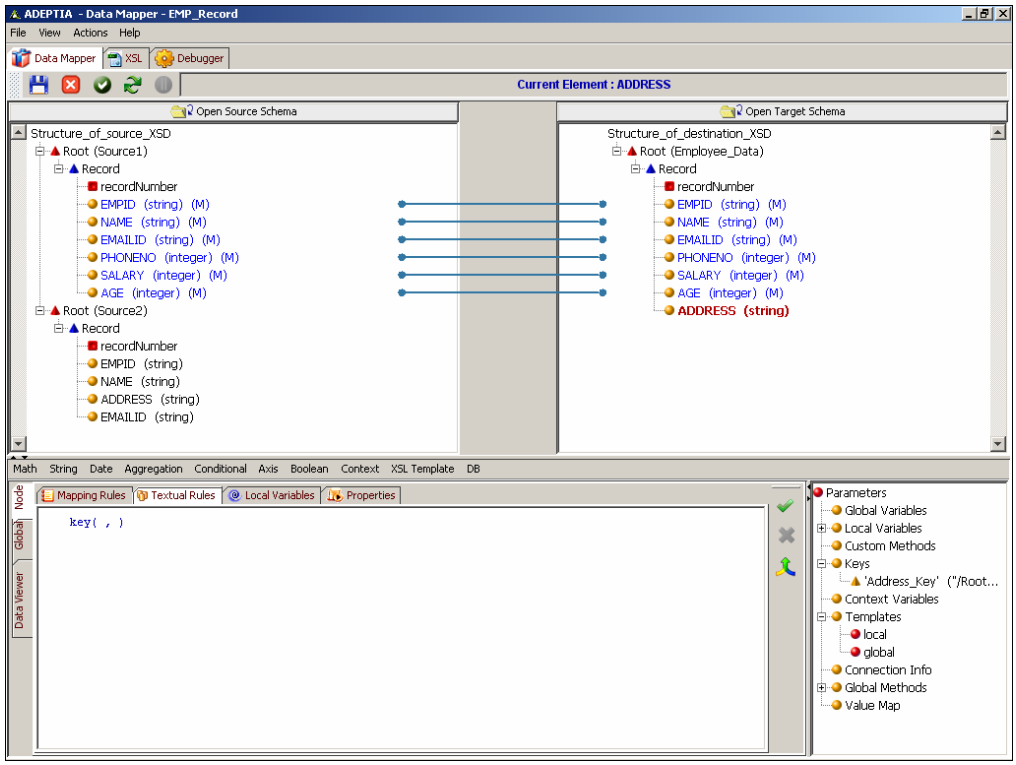


Figure 10.120: Key Function

9. Place the cursor, at the first argument of key function and double-click the required key that you want to apply, from *Keys* in the Parameters Panel. This is the first argument for the key function.

10. Place the cursor at the second argument of key function and double click the required *local variable*, under *Local/Ancestor Variables* in the Parameters Panel. This is the second argument for the key function.

11. The syntax of the key function in the textual rules pane will be displayed as:

    key ("key1" , $VarEmpidSource1)

12. Append the name of the element (*Address*), of the *source2* schema, which needs to be mapped to the target element (*Address*). Now the syntax of the mapping will be displayed as:

    key ("key1" , $ VarEmpidSource1)/Address

(see Figure 10.121)

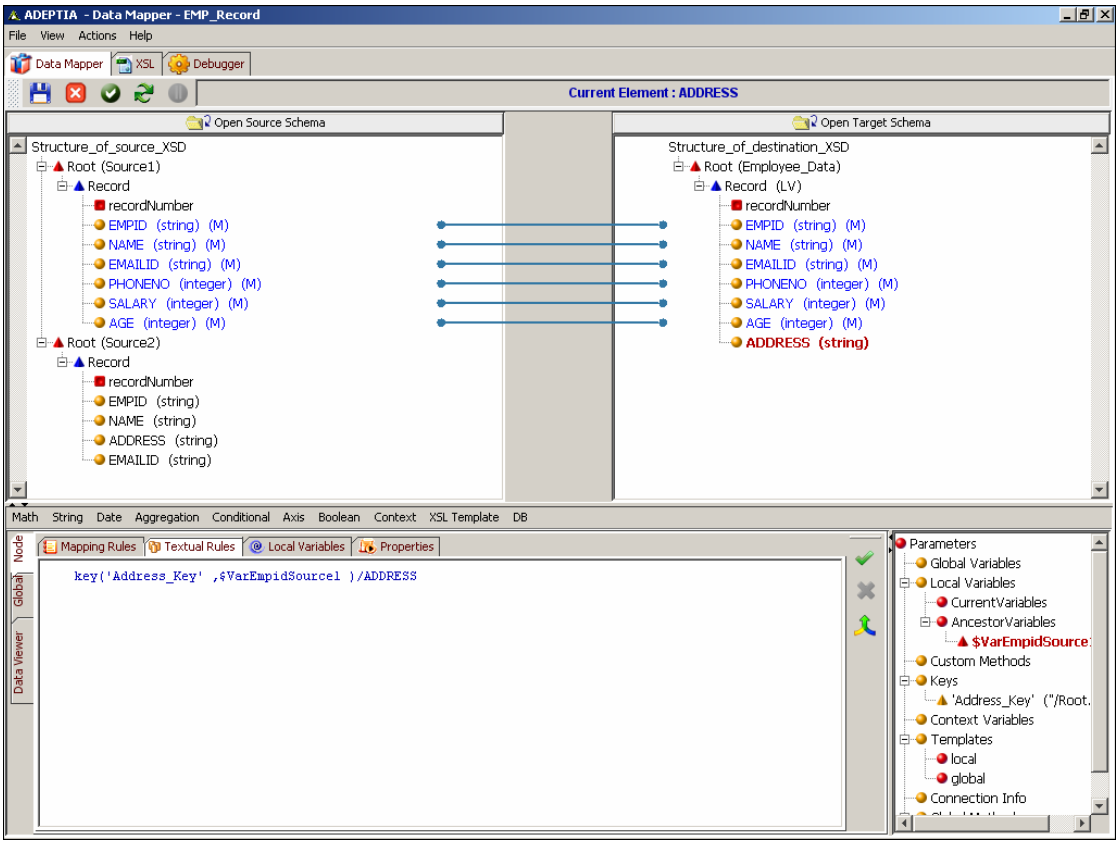


Figure 10.121: Map Key Function Node to Target Element

13. Click **Apply Mapping** ( ) button. This maps the key to the address element of the target schema. If you shift the focus to another target element, or click any of the *Mapping Rules, Textual Rules, Properties, Node, XSL* or *Debugger* tabs, without applying the mapping, then an alert message is displayed (refer to Figure 10.100).
14. Click **No** to apply the mapping and shift focus to the other object. If you click **Yes**, then the mapping activity is cleared and is replaced by the previous mapping in the Mapping Graph Area and the focus is shifted to the other object.
15. Apply *For Each* on element *address* of target schema from Root level of *Source2* schema. This will execute key in the context of second source schema (i.e. *Source2*).
16. **Save** the mapping activity and exit the Data Mapper.

	You can **view and validate the generated mapping XSL**, **view the target XML** and **view and validate mapping output**, before saving the mapping activity.

Declaring Connection Info Variable

Connection Info variables are used as a parameter in the DBQuery function, when extracting information from the database.

Steps to declare a Connection Info Variable

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click **Global** tab in the Mapping Graph Area. All tabs of the Global tab are displayed.
3. Click **Connection Info** tab. The Connection Info Variable pane is displayed in the Mapping Graph Area (see Figure 10.122).

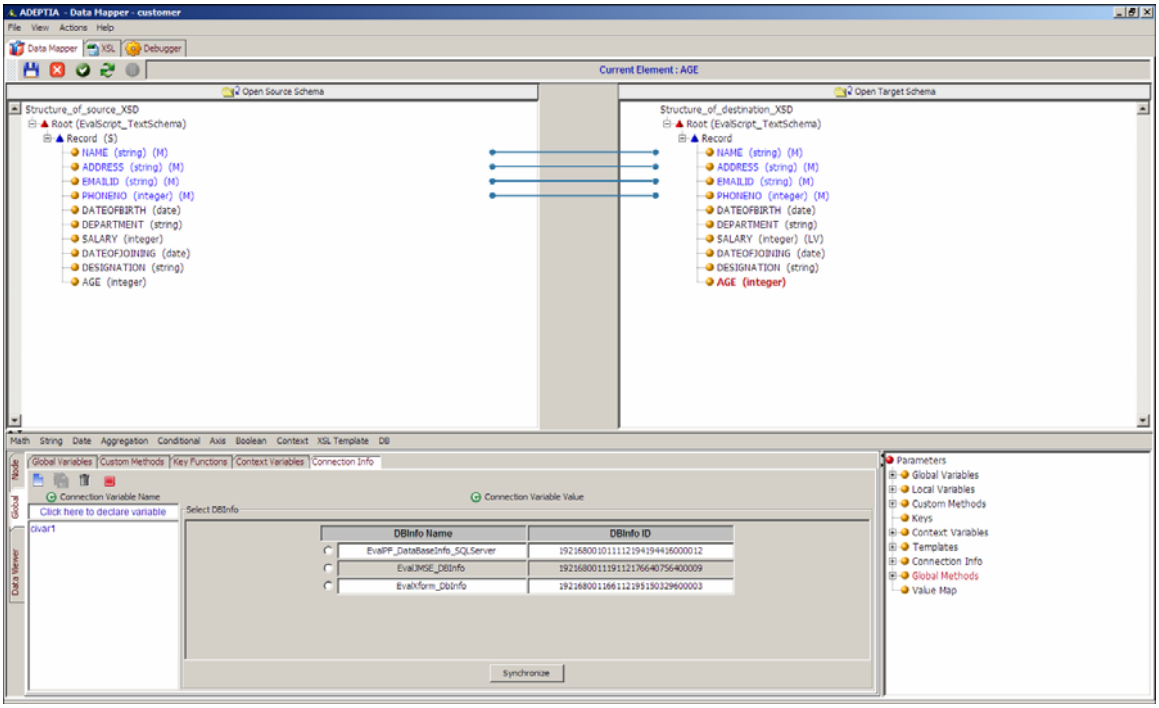


Figure 10.122: Connection Info Variable Pane

4. Click *Connection Variable Name* text field and enter the name of the *Connection Info* variable you want to create (e.g. Var1). Press the **Tab** or **Enter** key. This takes the control to the *Connection Variable Value* field.
5. The Connection Info variable accepts a **DBInfo** as its value. Thus, the *Connection Variable Value* field displays a list of existing **DBInfo Names** and their **IDs**. Select the **DBInfo** name that you want to assign for the *Connection Info* variable (see Figure 10.123).

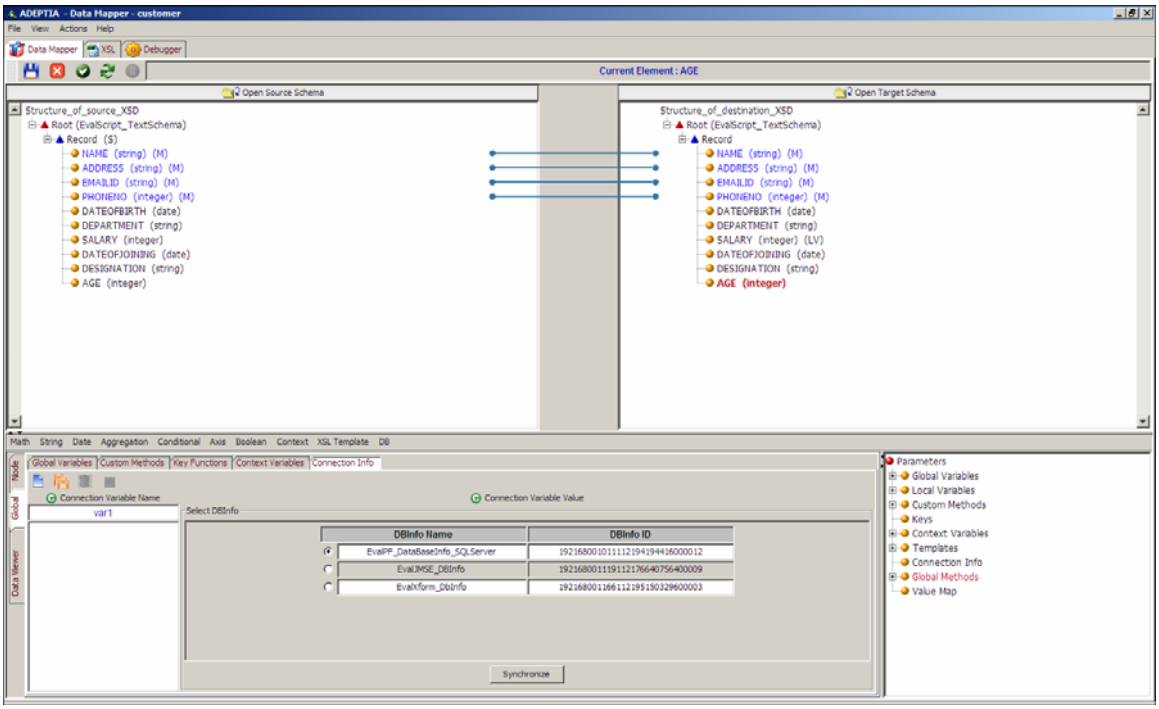


Figure 10.123: Enter Connection Variable Value

	You can click **Synchronize** to reload the **DBInfo** list.

6. Click **Save Connection Variable** ( ) button to save the *Connection Info* variable. This Connection Info variable is added to the list of existing variables in the *Connection Variable Name* field. It is also displayed under *Connection Info* in the Parameters Panel. If you shift the focus to another node, or click any of the *Mapping Rules, Textual Rules, Local Variables, Properties, XSL* or *Debugger* tabs, without saving the Connection Info variable, an alert message is displayed (refer to Figure 10.100).
7. Click **No** to save the Connection Info variable and shift the focus to the other object. If you click **Yes**, then the defined Connection Info variable is cleared and the focus is shifted to the other object.

	You can rearrange the Connection Info variables by dragging it up or down in the list.

	You can edit or delete a Connection Info variable from the Parameters Panel itself. For details, refer to the **Managing a Global Variable from Parameter Panel** section.

Setting Target Element Properties

You can set various properties of a target element.

Steps to set target element properties

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click the *target* element for which you want to set properties.
3. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
4. Click **Properties** tab. The Properties pane is displayed (see Figure 10.124)

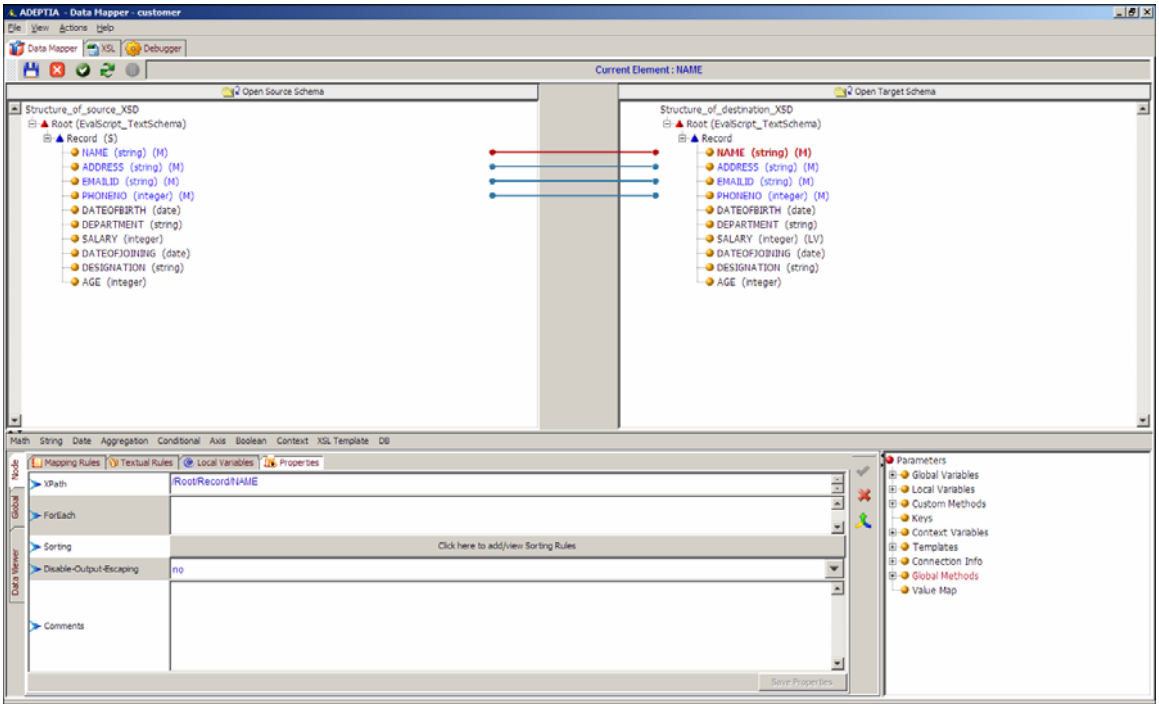


Figure 10.124: Properties Pane

The Properties pane comprises of various properties, which are outlined in the table below.

Table 10.22: Properties of a Target Element

Property	Description
XPath	Displays the XPath of the selected target element. It is a read-only field.
For Each	Sets the For Each property for the target element. It is used to repeat occurrences of a target element depending on the total occurrences of a source element in the source file. To set the For Each property refer to the section Setting the For Each Property.
Sorting	Sets the Sorting rules for the target element. It also allows you to view or remove sorting rules for the target element. To set the Sorting Rules, refer to the

	section Setting Sorting Rules.
Disable-Output-Escaping	Sets the disable-output –escaping property for the target element. It is used to include this property under the 'value-of' element in the generated XSL. By default, this property is set as 'no'. For details on this property, refer to the section Using XSL Property.
Comments	Displays comments entered for the target element. This is a data entry field and allows you to enter comments for the target element. To enter comments, refer to the section Adding Comments for Target Element.

5. Enter the properties required for the selected target element.
6. Click **Save Properties** to save the properties entered for the target element. If you shift focus to another node, or click any of the *Mapping Rules, Textual Rules, Global Variables, Properties, XSL* or *Debugger* tabs, without saving the properties, an alert message is displayed (refer to Figure 10.100).
7. Click **No** to save the properties and shift focus to the other object. If you click **Yes**, then the defined properties are cleared and the focus is shifted to the other object.

Setting For Each Property

The *For Each* property is used to repeat the occurrences of a target element or node, depending on the total occurrences of a source element or a node in the source file.

For example, an XML schema with a node customer has three elements, first_name, last_name, and state. This schema is used both at the source and at the target end.

Supposing, the source file has 10 occurrences of the customer. Applying the For Each property on the target node customer, for the source node customer, generates an output file containing 10 occurrences of the customer in it.

If *For Each* property is not used, an output file is generated with only one occurrence of customer in it.

For Each and *Apply Template* both can not be used simultaneously on one node.

Steps to set the For Each property

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click the *target* element for which you want to set the For Each property.
3. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
4. Click **Properties** tab. The Properties pane is displayed (refer to Figure 10.124).

5. Click the *For Each* property field and then double-click the *source* element using which you want to apply the For Each property. This displays the entire path of the source element in the *For Each* field (see Figure 10.125).

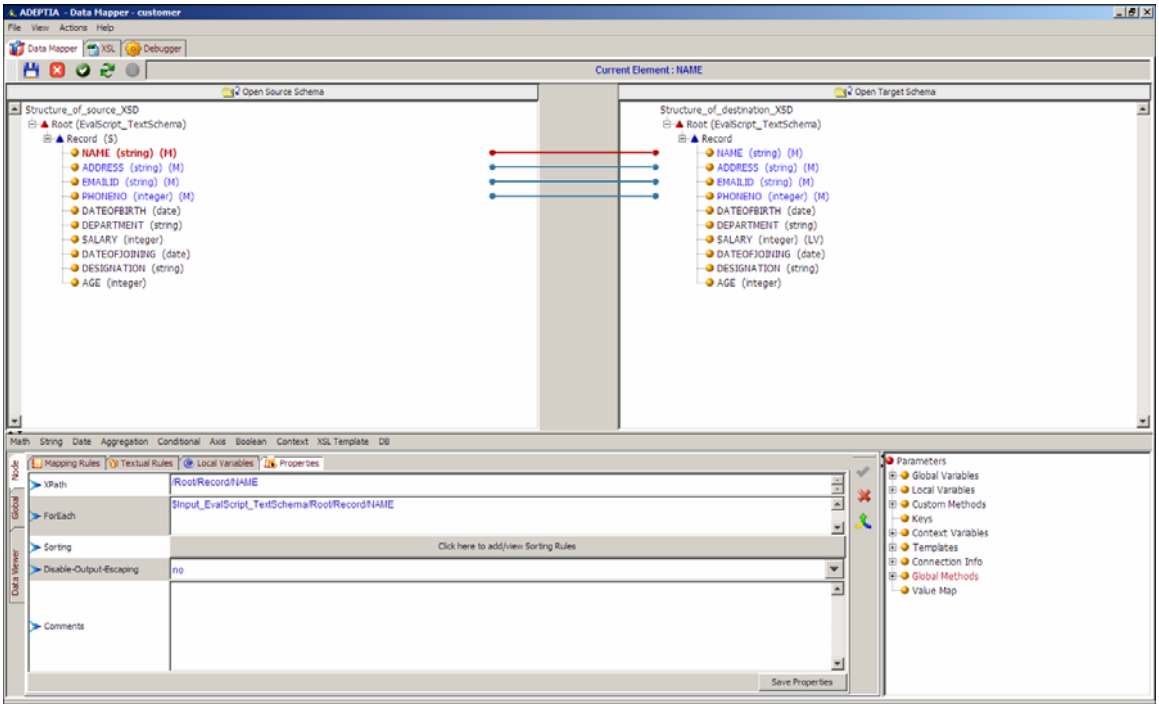


Figure 10.125: Source Element in For Each Property

	You can set the *For Each* property on more than one source element. You can use the | character as a separator between the XPath of each source element.

6. Click **Save Properties**. This applies the For Each property for the selected target element. If you shift focus to another node, or click any of the *Mapping Rules, Textual Rules, Global Variables, Properties, XSL* or *Debugger* tabs, without saving the For Each property, an alert message is displayed (refer to Figure 10.100).
7. Click **No** to save the For Each property and shift focus to the other object. If you click **Yes**, then the defined For Each property is cleared and the focus is shifted to the other object.

	Once the For Each property is applied to a target element, the letter **(F)** is displayed next to the target element. Refer to **Table of Suffixes** for details on suffixes displayed next to a target element.

	To remove the *For Each* property from the target element, right-click *target* element and select **Remove Mappings** option. This displays the Remove Mapping Options dialog box. Select the *Remove For Each* checkbox and click **OK**.

Adding Comments to Target Schema Elements

Comments are used to provide additional information to target nodes and elements. Comments are displayed as tool tips and are reflected in the XSL generated under the *XSL* tab.

Steps to add a comment to the target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click the *target* element for which you want to add a comment.
3. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
4. Click **Properties** tab. The Properties pane is displayed (refer to Figure 10.124).
5. Click the *Comments* property field and enter the comment for the selected target element (see Figure 10.126)

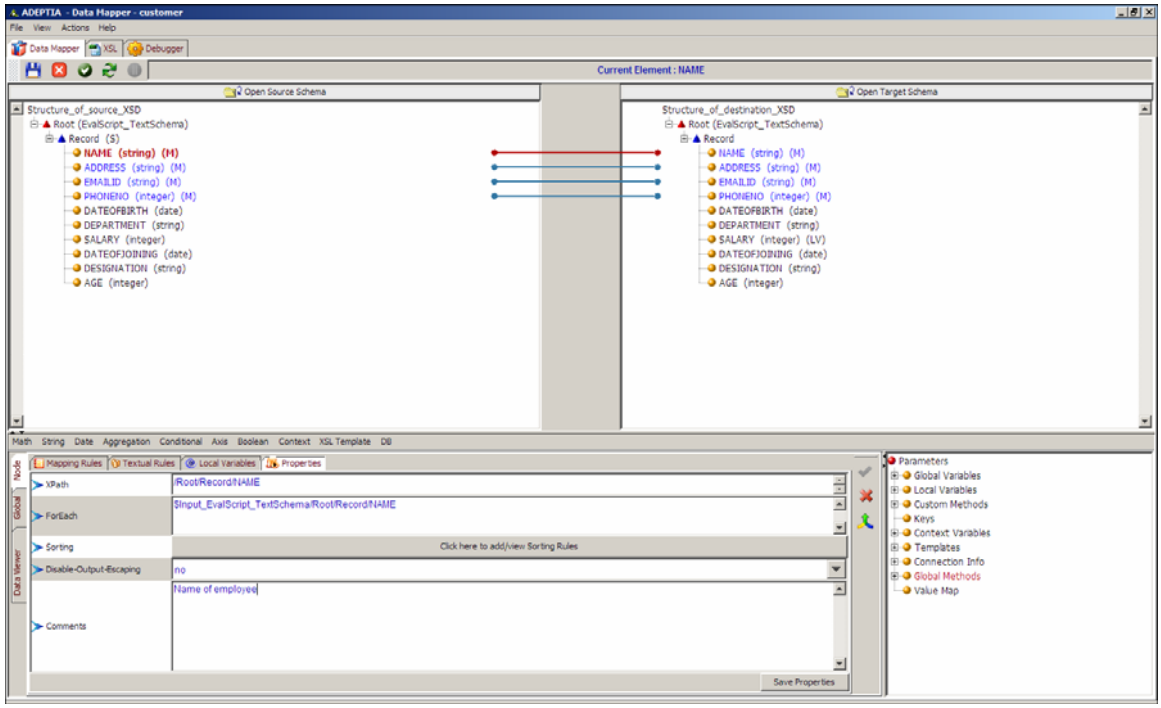


Figure 10.126: Enter Comment

6. Click **Save Properties**. This adds the comment and displays it next to the target element. If you shift focus to another node, or click any of the *Mapping Rules, Textual Rules, Global Variables, Properties, XSL* or *Debugger* tabs, without saving the comments, an alert message is displayed (refer to Figure 10.100).
7. Click **No** to save the comments and shift focus to the other object. If you click **Yes**, then the defined comments are cleared and the focus is shifted to the other object.

	Once a comment is added to a target element, the letter **(CM)** is displayed next to the target element. This signifies that a comment has been created for the target element. Refer to **Table of Suffixes** for details on suffixes displayed next to a target element.

Setting Sorting Rules

You can set sorting rules for a target element. It is used to sort value of elements in the generated output. While generating the output XML, the value of elements can be sorted either in ascending or descending order. For example, records of the employee can be sorted on the basis of their salaries. Figure 10.127 shows input XML before sorting.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <employees>
- <employee attribute="14" attribute1="15">
  <TESTCASENO>TESTCASENO1</TESTCASENO>
  <DESCRIPTION>DESCRIPTION1</DESCRIPTION>
  <NAME>John</NAME>
  <ADDRESS>Address1</ADDRESS>
  <EMAILID>EMAILID1</EMAILID>
  <PHONENO>PHONENO1</PHONENO>
  <DOB>DOB1</DOB>
  <DEPT>DEPT1</DEPT>
  <SALARY>10000</SALARY>
  <DOJ>DOJ1</DOJ>
  <DESIGNATION>DESIGNATION1</DESIGNATION>
  <AGE>34</AGE>
  </employee>
- <employee attribute="14" attribute1="15">
  <TESTCASENO>TESTCASENO2</TESTCASENO>
  <DESCRIPTION>DESCRIPTION2</DESCRIPTION>
  <NAME>David</NAME>
  <ADDRESS>ADDRESS2</ADDRESS>
  <EMAILID>EMAILID2</EMAILID>
  <PHONENO>PHONENO2</PHONENO>
  <DOB>DOB2</DOB>
  <DEPT>DEPT2</DEPT>
  <SALARY>8000</SALARY>
  <DOJ>DOJ2</DOJ>
  <DESIGNATION>DESIGNATION2</DESIGNATION>
  <AGE>45</AGE>
  </employee>
- <employee attribute="14" attribute1="15">
```

```
<TESTCASENO>TESTCASEN3</TESTCASENO>
<DESCRIPTION>DESCRIPTION3</DESCRIPTION>
<NAME>Ricky</NAME>
<ADDRESS>ADDRESS3</ADDRESS>
<EMAILID>EMAILID3</EMAILID>
<PHONENO>PHONENO3</PHONENO>
<DOB>DOB3</DOB>
<DEPT>DEPT3</DEPT>
<SALARY>15000</SALARY>
<DOJ>DOJ3</DOJ>
<DESIGNATION>DESIGNATION3</DESIGNATION>
<AGE>36</AGE>
</employee>
</employees>
```

Figure 10.127: Sample Input XML

Figure 10.128 shows the Output XML after sorting.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <employees xmlns:java="http://xml.apache.org/xslt/java"
xmlns:str="http://exslt.org/strings">
- <employee attribute="" attribute1="">
<TESTCASENO>TESTCASEN3</TESTCASENO>
<DESCRIPTION>DESCRIPTION3</DESCRIPTION>
<NAME>Ricky</NAME>
<ADDRESS>ADDRESS3</ADDRESS>
<EMAILID>EMAILID3</EMAILID>
<PHONENO>PHONENO3</PHONENO>
<DOB>DOB3</DOB>
<DEPT>DEPT3</DEPT>
<SALARY>15000</SALARY>
<DOJ>DOJ3</DOJ>
<DESIGNATION>DESIGNATION3</DESIGNATION>
<AGE>36</AGE>
</employee>
- <employee attribute="" attribute1="">
<TESTCASENO>TESTCASENO1</TESTCASENO>
<DESCRIPTION>DESCRIPTION1</DESCRIPTION>
<NAME>John</NAME>
<ADDRESS>Address1</ADDRESS>
<EMAILID>EMAILID1</EMAILID>
```

```
<PHONENO>PHONENO1</PHONENO>
<DOB>DOB1</DOB>
<DEPT>DEPT1</DEPT>
<SALARY>10000</SALARY>
<DOJ>DOJ1</DOJ>
<DESIGNATION>DESIGNATION1</DESIGNATION>
<AGE>34</AGE>
</employee>
- <employee attribute="" attribute1="">
<TESTCASENO>TESTCASENO2</TESTCASENO>
<DESCRIPTION>DESCRIPTION2</DESCRIPTION>
<NAME>David</NAME>
<ADDRESS>ADDRESS2</ADDRESS>
<EMAILID>EMAILID2</EMAILID>
<PHONENO>PHONENO2</PHONENO>
<DOB>DOB2</DOB>
<DEPT>DEPT2</DEPT>
<SALARY>8000</SALARY>
<DOJ>DOJ2</DOJ>
<DESIGNATION>DESIGNATION2</DESIGNATION>
<AGE>45</AGE>
</employee>
</employees>
```

Figure 10.128: Output XML

The Sorting rules can be set only for record where *For Each Mapping* or *Apply Template* is used. The *<xsl:apply-templates>* element applies a template to the current element or to the child nodes of the current element, when the parent elements of source and target schemas are mapped.

Steps to set Sorting Rules

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Ensure that the **source and target elements to be mapped are selected and displayed** in the Mapping Graph Area.
3. To use Apply template, click *Employee* (parent) element in the source panel and drag the mouse pointer to the *Employee* (parent) element in the target panel. A line is displayed between the source and target panels indicating the mapping between source and target elements (see Figure 10.129).

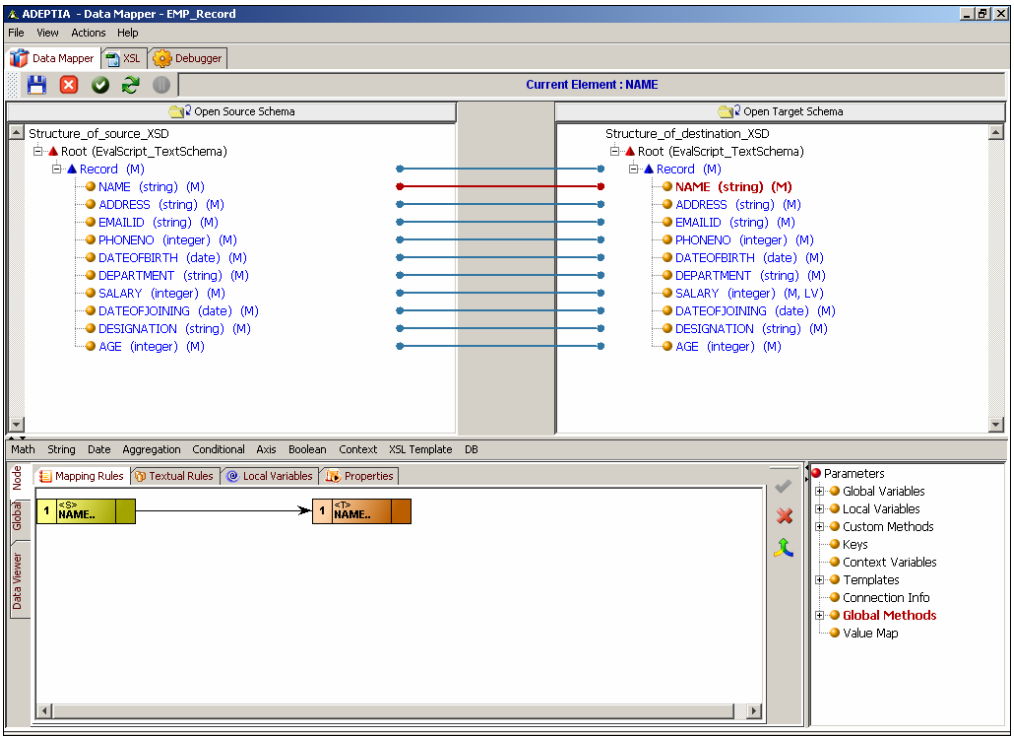


Figure 10.129: Apply Template

	To learn how to set For Each property on employee element refer to the section **Setting For Each Property**.

4. Click **Node** tab displayed in the Mapping Graph Area. All tabs of the Node tab are displayed.
5. Click **Properties** tab. The Properties pane is displayed (refer to Figure 10.124).
6. Click the **Click here to add/view Sorting Rules** option displayed against the Sorting property field. This displays the Add New Sorting Rules dialog box (see Figure 10.130)

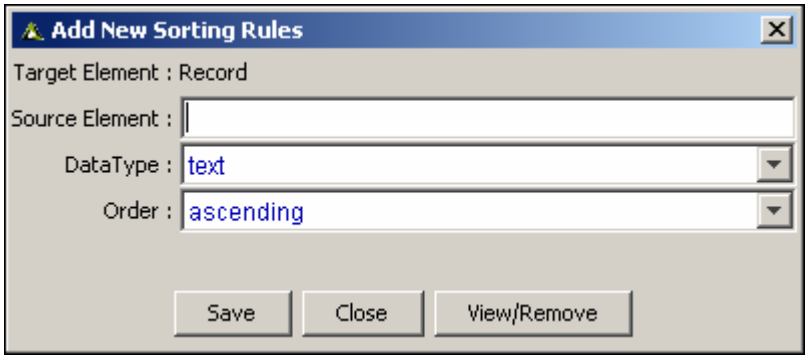


Figure 10.130: Sorting Rules Dialog Box

7. This screen displays the name of the selected target record to be sorted. Type the name of the source element on the basis of which you want to sort output (e.g. SALARY), in the *Source Element* field.
8. Select the data type as either text or numeric from the *Data Type* drop-down list. This specifies how the input value is to be interpreted. By default, text is selected.
9. Select the sorting order as ascending or descending from the *Order* drop-down list. By default, ascending is selected.
10. Click **Save** to save the sorting rule.
11. Click C**lose** to return the Data Mapper screen. The letter **(SR)** is displayed next to the sorted target element. This signifies that the target element has been sorted based on the specified source element. Refer to the **Table of Suffixes** for details on suffixes displayed next to an element.

	More than one sorting can also be applied to a single target element. For example, if salary of two employees is same, they can be further sorted out based on their age. To delete sorting, click **View/Remove** button. The Remove Sorting Rules dialog box is displayed with a list of existing sorting rules. Select sorting rule and click **Remove/Remove All** to delete sorting rules.

Once the sorting is applied based on the set rules, the output XML is displayed as displayed in Figure 10.128.

Filtering of Elements/Attributes in Target Data

You can filter elements or attributes from appearing in the target XML. There are two ways to filter data:

- Using IFF Condition
- Using Apply Filter checkbox from right-click popup menu

	The Apply Filter checkbox method is recommended as it is easy to use and can be done individually on multiple nodes, directly from the data mapper applet.

Using Apply Filter Checkbox

Steps to filter using Apply Filter Checkbox

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the *target* element that you want to filter, and mark the *Apply Filter* checkbox as checked (see Figure 10.131). By default, this checkbox is unchecked.

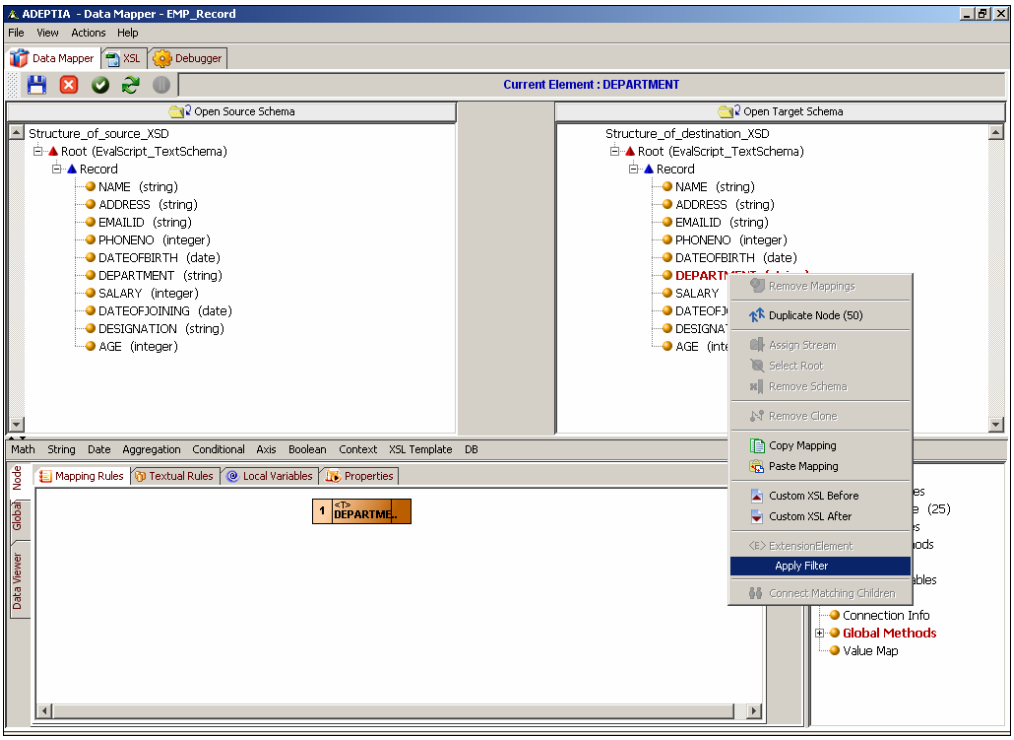


Figure 10.131: Checking Apply Filter Checkbox

3. This selection will filter the selected element from the target output XML data. You can uncheck this checkbox anytime to display the element in the target data.

	You can apply this filtering option on the element or attribute levels.
	Once the filter is applied on the target element, the letters (FL) are displayed next to each filtered target element. This signifies that the filter has been applied for the target element. Refer to **Table of Suffixes** for details on suffixes displayed next to an element.

Creating Clone of Target Element

Cloning means generating replicas of the target elements or nodes. You can create clones of all target elements.

Steps to create a clone

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the *target* element and select the **Duplicate Node** option.

	You can create a clone of all target nodes except the root node. Thus, the *Duplicate Node* sub-option is displayed as inactive for the root node.

3. The element which is created using cloning is displayed with the suffix [1] after its name. Subsequent clones that are created for that element will have the suffix [2] and so on. You can create a maximum of 50 occurrences of an element, with one main element and 49 clones.

Removing Clone of Target Element

Steps to remove a clone

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the cloned element that you want to remove and select the **Remove Clone** option. A screen is displayed confirming the removal of the selected clone (see Figure 10.132).

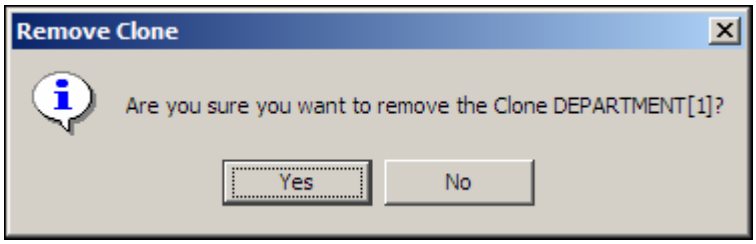


Figure 10.132: Confirm Remove Clone

3. Click **Yes** to remove the clone. This removes the selected clone from the target schema.

	If more than one clone has been created for an element, then you need to remove the clones in descending order.

Removing Schema

You can remove a schema from the source or target panels. If a source or target schema is removed, everything associated with the mapping of the schema such as Mapping Graph Area, Connection Lines or For Each property is removed.

Steps to remove a schema

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the *root* element of the schema that you want to remove and select the **Remove Schema** option. A screen is displayed confirming the removal of the selected schema (see Figure 10.133).

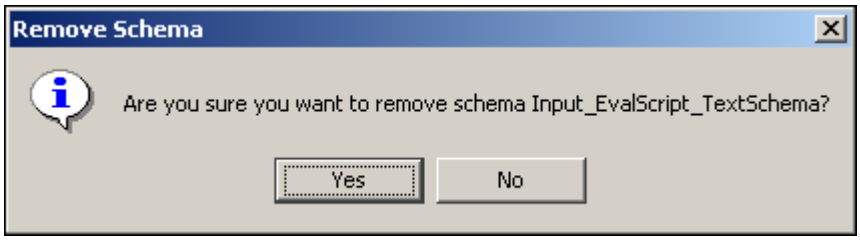


Figure 10.133: Confirm Remove Schema

3. Click **Yes** to remove the schema. This removes the selected clone from the target schema.

	A schema can be removed only from the root element. Thus, the *Remove Schema* option is displayed as active only for the root element.

Adding Custom XSL

At times, you may want to add some additional specialized XSL to a target element, or at the top or bottom of the mapping XSL. You can do this by adding custom XSL code.

Steps to add custom XSL code for a Target Element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the *target* element and select the **Custom XSL Before** option (see Figure 10.134).

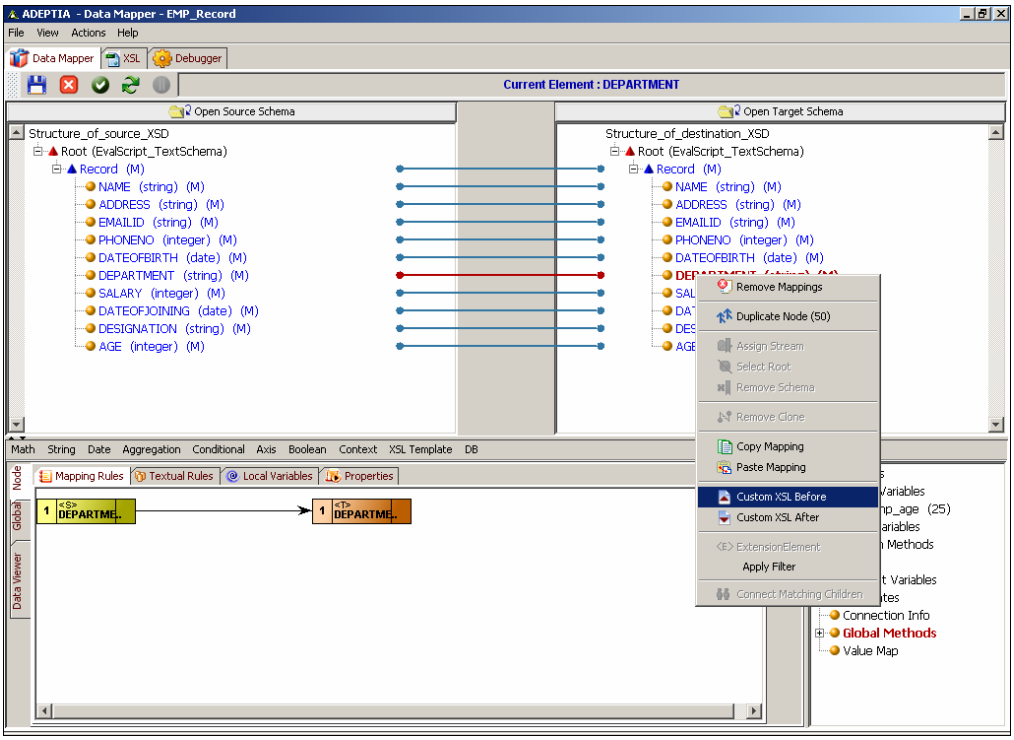


Figure 10.134: Select Custom XSL Before Option

3. The Add Custom XSL Before screen is displayed for the target element. Enter the custom XSL code for the target element (see Figure 10.135).

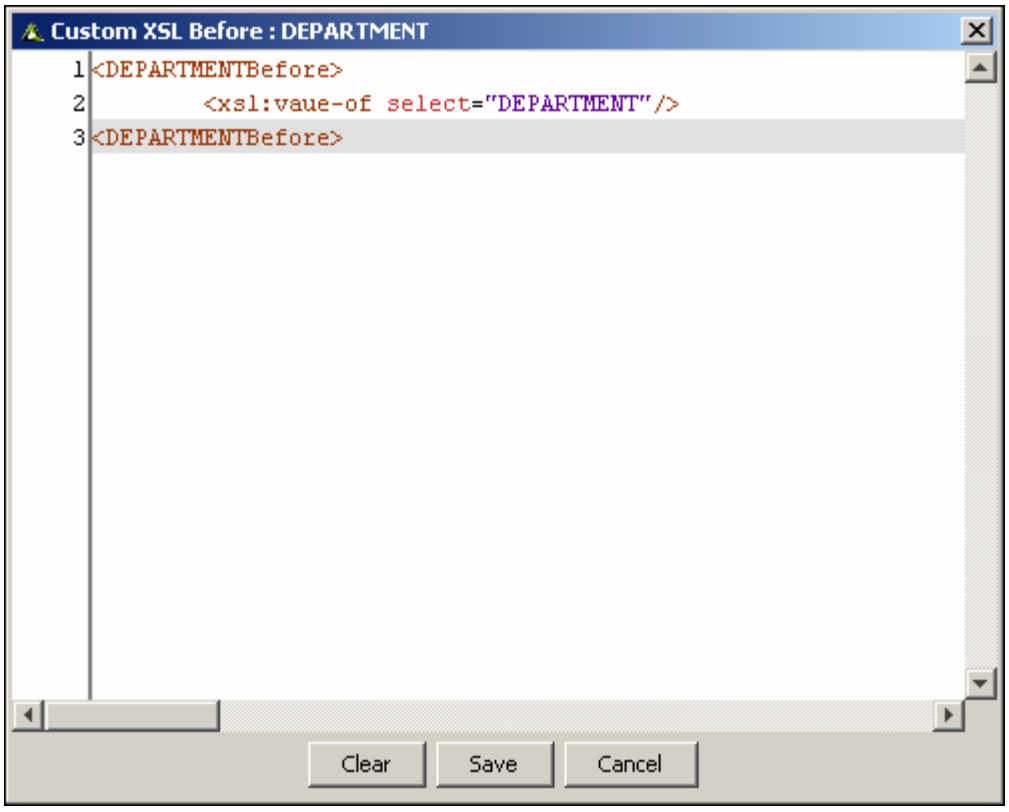


Figure 10.135: Enter Custom XSL Code

4. Click **Save** to save the XSL code and close the screen. Alternately, click **Clear** to clear the XSL code, or **Cancel** to close the screen without saving the changes.

	Similarly, you can add custom XSL code after a target element, by selecting **Custom XSL After** option from the right menu (refer to Figure 10.134).

Steps to add global custom XSL code

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Click the **Actions** menu and select **Global Custom XSL Before** option (see Figure 10.136).

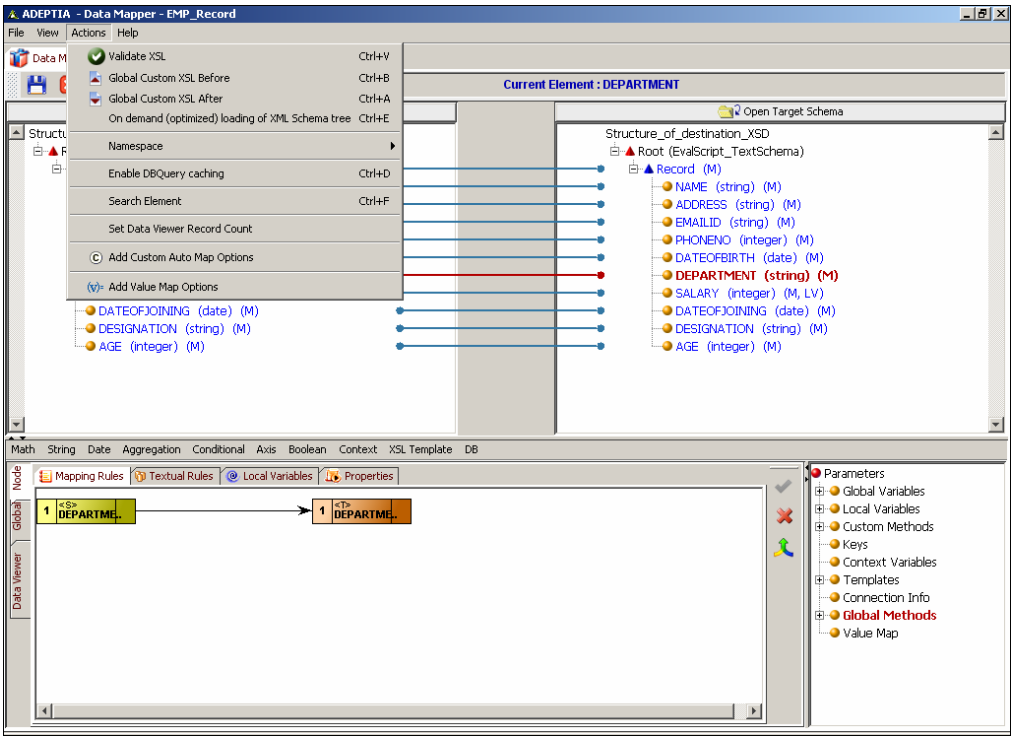


Figure 10.136: Select Global Custom XSL Before Option

3. The Add Global Custom XSL Before screen is displayed. Enter the global custom XSL code (see Figure 10.137).

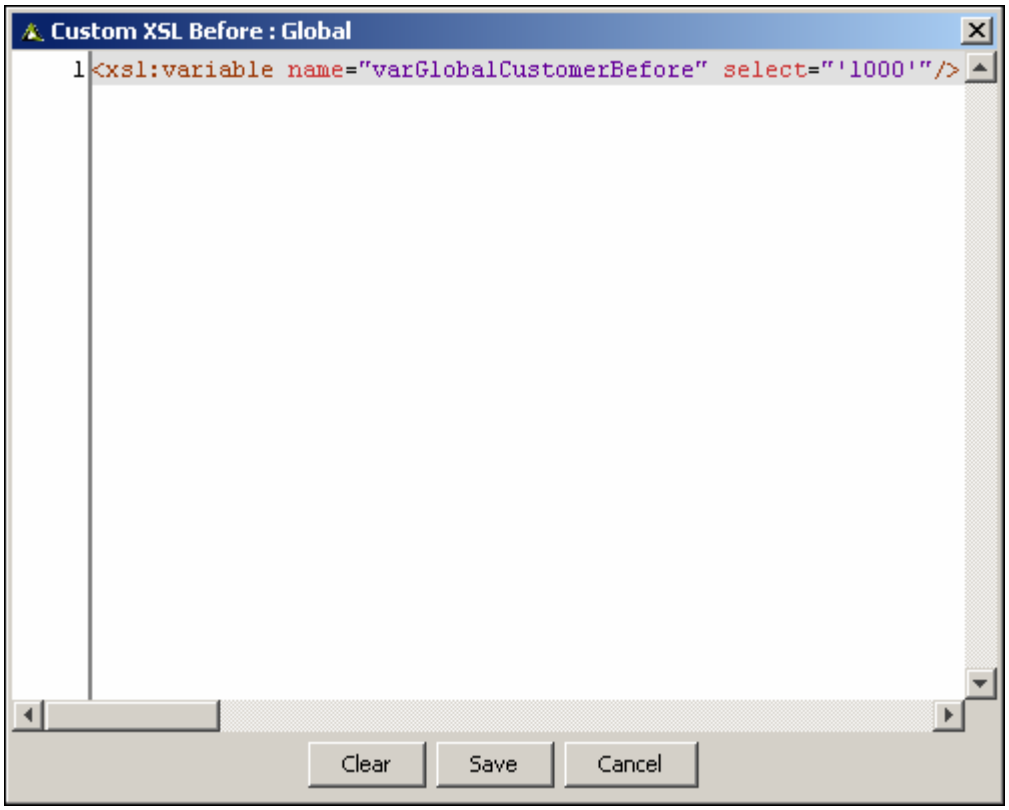


Figure 10.137: Enter Global Custom XSL Code

4. Click **Save** to save the XSL code and close the screen. Alternately, click **Clear** to clear the XSL code, or **Cancel** to close the screen without saving the changes.

	Similarly, you can add global custom XSL code after mapping XSL, by selecting **Global Custom XSL After** option from the Actions menu (refer to Select Global Custom XSL Before Option screen).

5. Once you have entered the custom XSL code, it is saved in the Mapping XSL screen (see Figure 10.138).

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.1" xmlns:java="http://xml.apache.org/xslt/java" xmlns:xalan="
    <xsl:output method="xml" version="1.0" encoding="ISO-8859-1" indent="yes"/>
    <xsl:param name="_userName"/>
    <xsl:param name="_password"/>
    <xsl:param name="_class"/>
    <xsl:param name="_identifier"/>
    <xsl:param name="_subject"/>
    <xsl:param name="_repositoryPath"/>
    <xsl:param name="_Input1"/>
    <xsl:variable name="Input_Demo_EmployeeSchema" select="document($_Input1)"/>
    <xsl:variable name="apos">'</xsl:variable>
<!--Global PreCustom XSL starting-->
    <xsl:variable name="varGlobalCustomBefore" select="'1000'"/>
<!--Global PreCustom XSL ending-->
    <xsl:template match="/">
        <employees>
            <employee>
<!--PreCustom XSL starting for element EMPID-->
                <EMPIDBefore>
                    <xsl:value-of select="EMPID"/>
                </EMPIDBefore>
<!--PreCustom XSL ending for element EMPID-->
                <EMPID>
                    <xsl:value-of select="$Input_Demo_EmployeeSchema/employees/employee/EMPID"/>
                </EMPID>
<!--PostCustom XSL starting for element EMPID-->
                <EMPIDAfter>
                    <xsl:value-of select="EMPID"/>
                </EMPIDAfter>
<!--PostCustom XSL ending for element EMPID-->
                <FIRSTNAME>
                    <xsl:value-of select="$Input_Demo_EmployeeSchema/employees/employee/FIRSTNAME"/>
                </FIRSTNAME>
                <LASTNAME>
                    <xsl:value-of select="$Input_Demo_EmployeeSchema/employees/employee/LASTNAME"/>
                </LASTNAME>

        </employees>
    </xsl:template>
<!--Global PostCustom XSL starting-->
    <xsl:variable name="varGlobalCustomBefore" select="'1000'"/>
<!--Global PostCustom XSL ending-->
```

Figure 10.138: Custom XSL Code in Mapping XSL

Select Root

You can change the root element when XML schemas are loaded. The root element always appears as a tag in the Output section on the *Debugger* screen. You can change this tag by selecting the desired root element.

Steps to select root element of a schema

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the *root element* of the schema whose root element you want to change and select the **Select Root** option.
3. The Select Root Element screen is displayed (see Figure 10.139). This screen displays a list of existing root elements for the selected schema in a drop-down list.

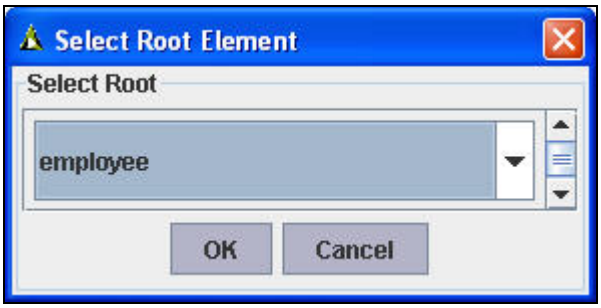


Figure 10.139: Select Root Element

	All elements that are declared as global in the XSD are listed in this drop-down list.

4. Select the *root element* that you want to change from the *Select Root* drop-down list and click **OK**. A screen is displayed confirming the change of the root element (see Figure 10.140).

Figure 10.140: Confirm Change of Root Element

5. Click **Yes** to change the root element. This changes the root element and displays the selected root element.

	The *Select Root* sub-option is displayed as active only for the root element.

Adding Namespace Prefix

You can add a namespace prefix to a target element. This prefix is displayed next to the target element in the target schema, and is automatically displayed in the Output section on the *Debugger* screen.

Steps to add a namespace prefix to a target element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.

2. Right-click the element of the target schema for which you want to add a namespace prefix and select the **Add Namespace Prefix** option. The Add Namespace Prefix screen is displayed (see Figure 10.141).

Figure 10.141: Add Namespace Prefix

3. Enter the namepace prefix and click **OK**. This adds the prefix and displays it next to the selected target element in the target schema.

	Once you add a namespace prefix for a target element, you need to add a namespace for it. To do this, refer to **Adding Namespace** section.

Populating extension type element

If complex type of element is an extension type i.e. other complex types are extending from this type, then you have the option to load the hierarchy from the list of complex types, which are extending from original element type.

Steps to select extension element

1. Ensure that the **source and target schemas are loaded** and all their elements are listed under their respective nodes.
2. Right-click the extension element of the schema whose element hierarchy you want to change and select the **Extension Element** option.

	The *Extension Element* sub-option is displayed as active only for the extension element.

3. The Select Extension Element screen is displayed (see Figure 10.142). This screen displays a list of extension element type.

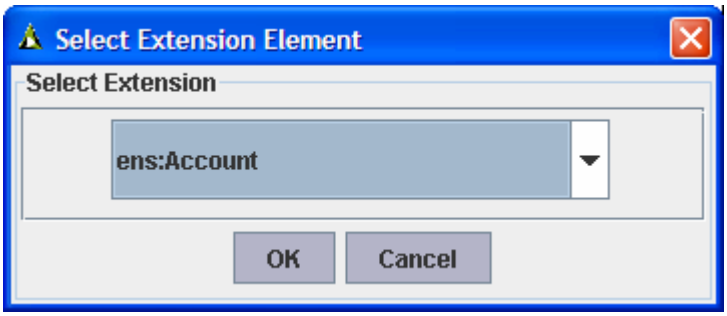


Figure 10.142: Select Extension Element

4. Select the required extension type that you want to load from the *Select Extension Element* drop-down list and click **OK**. A screen is displayed confirming the change of extension element (see Figure 10.143).

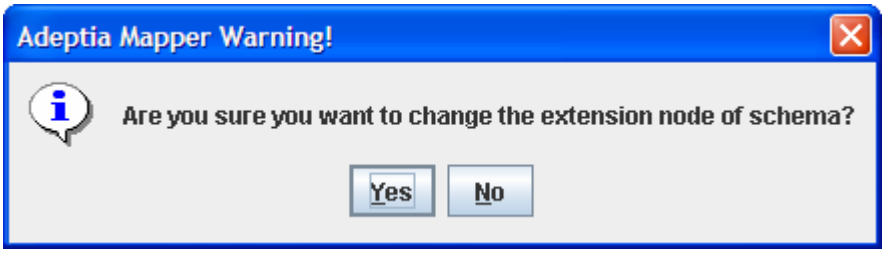


Figure 10.143: Confirm Change of Extension Element

5. Click **Yes** to change the extension element. The selected hierarchy will be loaded.

USING RECORD TO RECORD SERVICE

Record to Record Service takes inputs record by record and processes them according to the defined logic using java programming construct, and gives the output one record at a time. User can write Java logic that will be executed in this service. The Java logic has access to Record to Record script service, context of the process flow this service belongs to, input and output Stream handlers, which allow Java logic to access and manipulate input data to generate output and pass it to another service in the process flow.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Prerequisites:

- *Schema* activity must be created before creating *Record to Record Service*.

Steps to create Record to Record activity:

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Data Transform** to expand the tree, and then click **Record to Record**. The Manage Record to Record screen is displayed (see Figure 10.144).

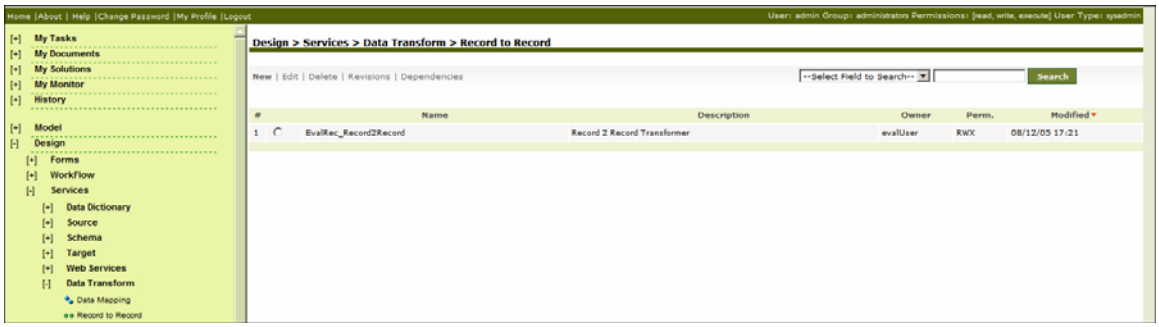


Figure 10.144: Manage Record to Record Service

4. Click the **New** link. The Create Record to Record screen is displayed (see Figure 10.145).

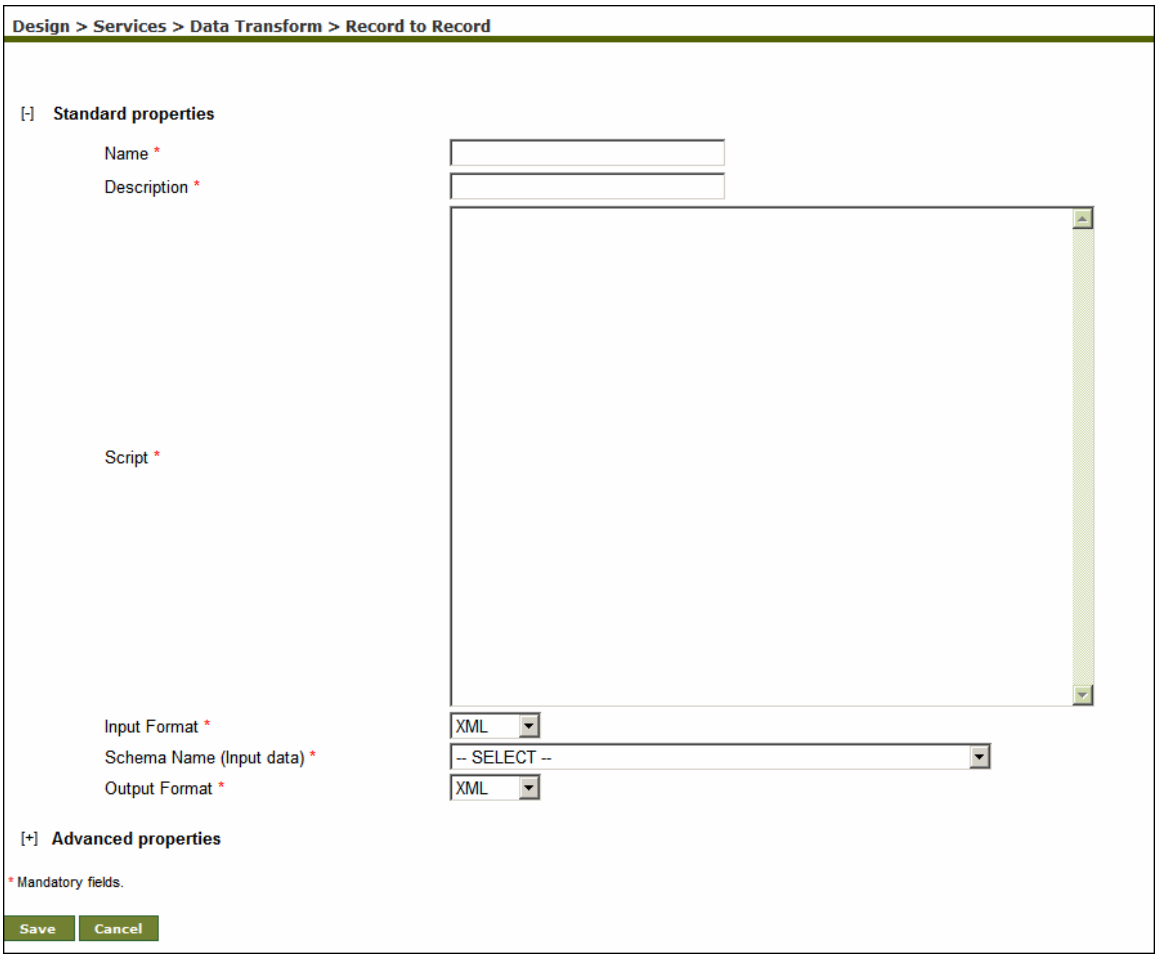


Figure 10.145:Create Record to Record Service

5. Enter the name and description of the new Record to Record service in the *Name* and *Description* fields respectively.

6. Enter the sample Java script displayed in Figure 10.146 to perform the desired transformation.

Template Script

```
Note: the complete script is executed for each record.

// setting data into process flow context
  // context - This object represents the process flow context  which is
implicitly available to Script
    context.put("Variable name", "Object value") ;

// reading data from process flow context
  Object value=context.get("Variable name");

//  Reading the input Record
```

```
// record - represents the input data record object, which is implicitly available to Script
  String name = record.getField("name");
  String email = record.getField("email");

// Creating an Empty record
 com.adeptia.indigo.io.Record outputRecord = com.adeptia.indigo.io.IoObjectPool.borrowRecord();

// Populating output Record
 outputRecord.addField(name);
 outputRecord.addField(email);

// Writing output record to output stream read by another activity
 // service- This is the "Record to Record" Service object which is also available implicitly to the script.
 // "default"- this is the name of the output stream.keep this as it is.while making process flow make sure you use the default stream only.
  service.write(outputRecord, "default");

// To wait for certain flag in process flow context so that next record can be processed based on the value of flag
 while(true)
 {
  String isRecievedAck = (String) context.get("recievedAck");
  if(isRecievedAck == null || (isRecievedAck.equals("false")))
  {
   try
   {
    Thread.sleep(300);
   }
   catch(Exception e)
   {
   }
  }
  else
  {
   break;
  }
 } // end
```

Figure 10.146: Sample Java Script

7. Select input and output format as either XML or Native (non XML) from the *Input Format* and *Output Format* drop-down lists respectively.

8. Select the source schema activity from the *Schema Name* drop-down list.

	If any XML Schema is selected in the *Schema Name* drop-down list, then the *Input Format* must be selected as XML. To learn how to create Schema activity, refer to Creating Schema Activity section. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click the **Save** button. A screen is displayed confirming that the Record-to-Record activity has been created successfully. If the comments property is enabled, then clicking Save will display a screen where you need to enter comments related to creating the Record-to-Record service (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the Record-to-Record service has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

11 CREATING EXTENSIONS

This section describes the process of **creating a custom plugin activity**.

CREATING CUSTOM PLUGIN

A custom plugin is a scripted service that can process data in a customized manner using java-programming constructs and provides an extension point for adding any customized data processing logic. It takes the input as a stream and generates the output as a stream. You can write a Java logic that will be executed in this service. Java logic has access to the script service, context of the process flow this service belongs to, input and output stream handlers that allow Java logic to access and manipulate input data to generate the output and pass it to another service in process flow.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create custom plugin activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Extensions** to expand the tree, and then click **Custom Plugin**. The Manage Custom Plugin screen is displayed (see Figure 11.1).

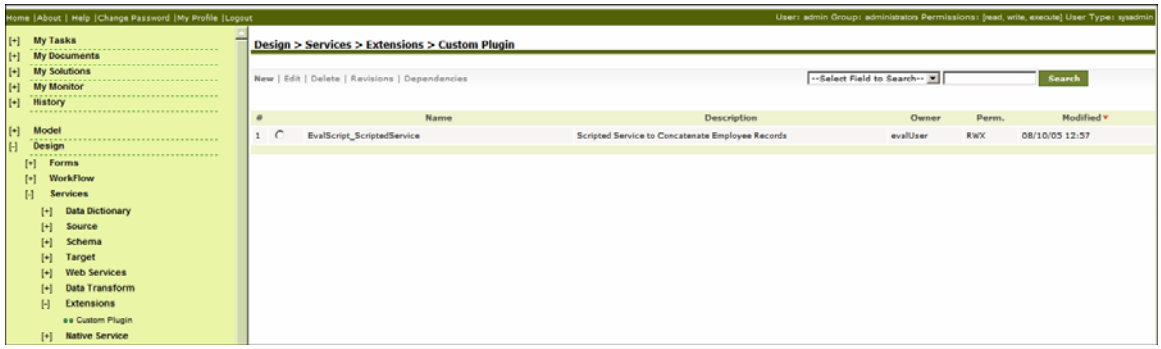


Figure 11.1: Manage Custom Plugin

4. Click the **New** link. The Create Custom Plugin screen is displayed (see Figure 11.2).

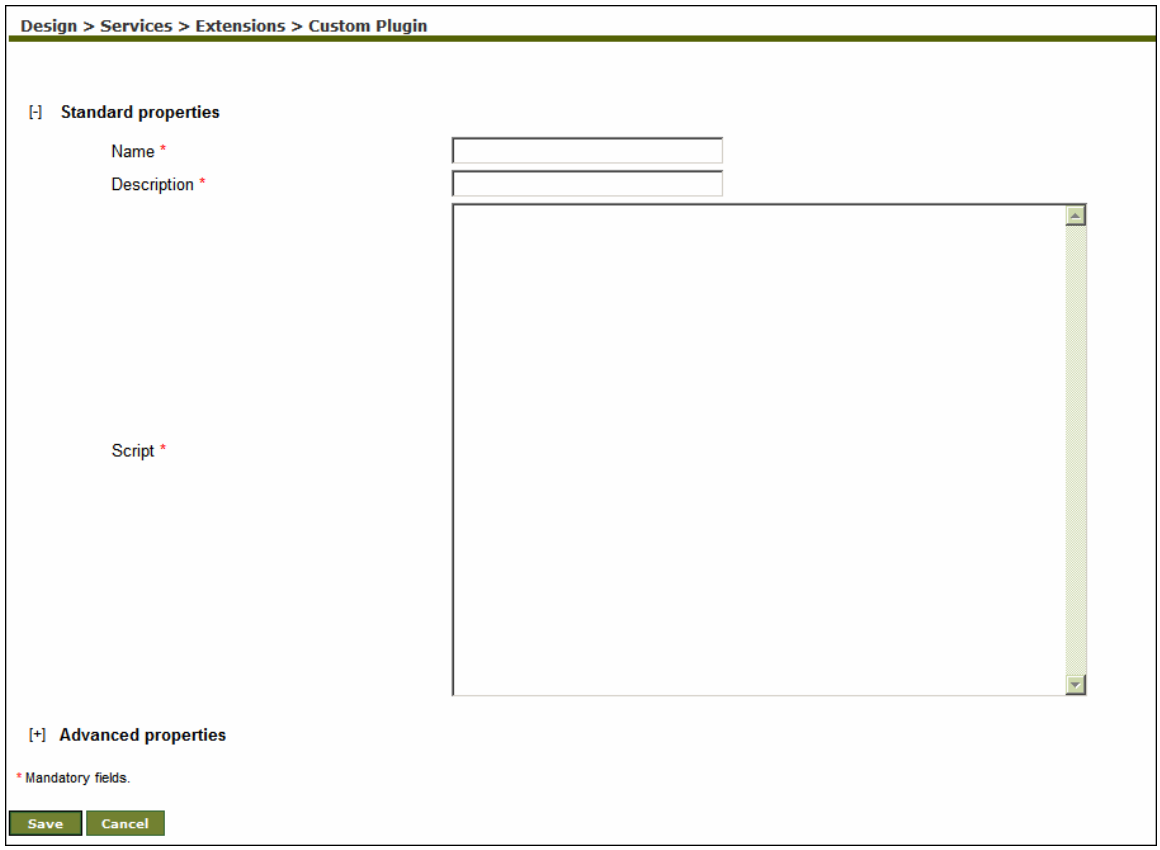


Figure 11.2: Create Custom Plugin

5. Enter the name and description of the new custom plugin in the *Name* and *Description* fields respectively.

6. Enter the sample Java script displayed in Figure 11.3 to perform the desired transformation.

Template Script:

```
// Setting data into process flow context
  // context - This object represents the process flow context,
which is implicitly available to Script
    context.put("Variable name", "Object value");

// Reading data from process flow context
    Object  value = context.get("Variable name");

// Reading the input data
  // inputStream - represents the input data Stream, which is
implicitly available to Script
    BufferedInputStream bstream = new
BufferedInputStream(inputStream);

// Any data processing logic
```

```
    ZipInputStreamzin = new ZipInputStream(new
BufferedInputStream(inputStream));
    byte[] data = zin.compress().getBytes()];

// Writing output data to output stream read by another activity
  // service - this is the "Scripted Service" Service object, which
is also available implicitly to the script.
  // "default" - this is the name of the output stream. Keep this
as it is. While making process flow make sure you use the default
stream only.
    service.write(data, "default");
```

Figure 11.3: Sample JAVA Script

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

7. Click the **Save** button. A screen is displayed confirming that the custom plugin activity has been created successfully. If the *comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the custom plugin (refer to Figure 3.6).

8. Enter the 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 custom plugin has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

12 CREATING NATIVE SERVICE ACTIVITY

This section describes the process of **creating a Native Call activity**.

CREATING NATIVE CALL ACTIVITY

Native Call is used to run the .EXE, .BAT or .SH files asynchronously during the execution of a process flow.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to create Native Call activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Native Service** to expand the tree, and then click **Native Call**. The Manage Native Call screen is displayed (see Figure 12.1).

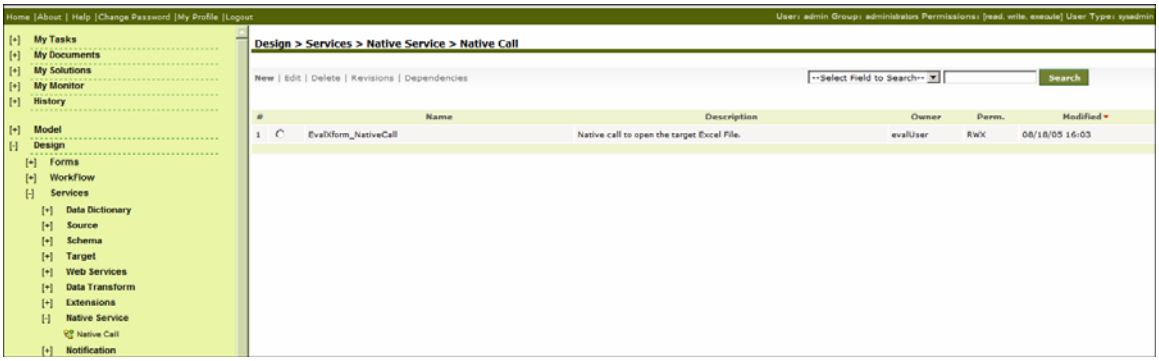


Figure 12.1: Manage Native Call

4. Click the **New** link. The Create Native Call screen is displayed (see Figure 12.2).

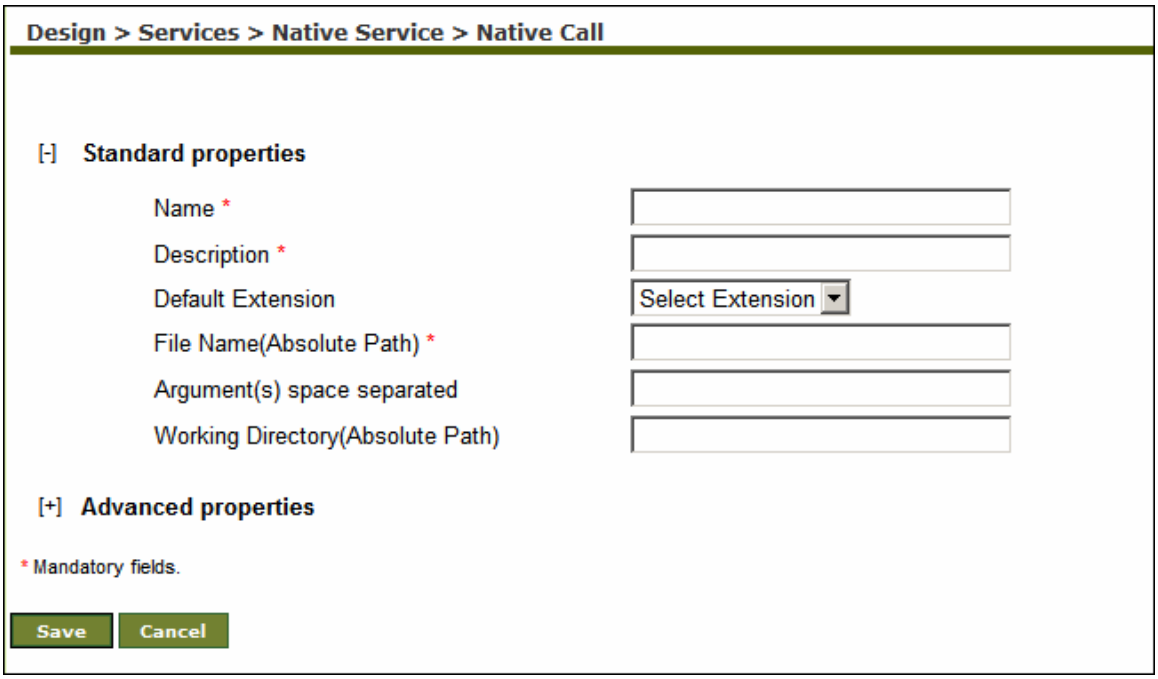


Figure 12.2: Create Native Call

5. Enter the name and description of the new Native Call activity in the *Name* and *Description* fields respectively.
6. Select the extension of the file i.e. .exe, .bat or .sh from the *Default Extension* drop-down list.
7. Enter the filename with full path in the *File Name(Absolute Path)* field.
8. To specify any arguments for selected batch or executable file enter the arguments in the *Argument(s) space separated* field.
9. Enter the path of directory, where you want the run the specified batch or executable file, in the *Working Directory (Absolute Path)* field.

	All the files specified into batch or executable file must be placed on the server on which the Adeptia Server is running.

	If Working Directory is not specified, the home directory of Adeptia Server, (i.e. *../../AdeptiaServer/AdeptiaeServer-4.9*) becomes the working directory. All the files specified into batch or executable file will be looked into Adeptia Server home path. For example, there is batch file *Mybatch.bat* and it is placed on C drive. The contents of batch file are *Copy File1.txt File2.txt*. *File1.txt* is placed in C drive. Since the working directory is not specified the batch file will look for *File1.txt* into *../../AdeptiaServer/AdeptiaServer-4.9*and it will cause error. So you can either specify the working directory (i.e. C:\) or specify the absolute path (e.g. *copy C:\File1.txt C:\File2.txt*) in the batch file.

10. Click **Save** button. This displays a screen confirming that the native call activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a

screen where you need to enter comments related to creating the native call (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

	When using a native call in a process flow, the *waitForEnd* property is set to *False* by default. This implies that the native call and other activities in the process flow are executed simultaneously. If you want to wait till the native call is complete, before executing the next activity in the process flow, you need to set this property to *True*.

13 CREATING NOTIFICATION ACTIVITY

This section covers the following topics:

CREATING MAIL NOTIFICATION ACTIVITY

In the Adeptia Suite, Notification is used to send mail to users about the execution status of a process flow.You can add the Mail Notification activity anywhere in the process flow. You can add more than one Mail Notification activities in a process flow.

The Mail Notification activity is of two types:

- **User Defined Message:** This type of mail notification is used to send custom email message to any user during execution of a process flow. Any file can be attached to the mail.
- **Process Flow Summary:** This type of mail notification is used by the Adeptia Server to dynamically send summary of a process flow execution.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

CREATING NOTIFICATION TO SEND USER DEFINED MESSAGE

Steps to create User Defined Message Notification

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Notification** to expand the tree, and then click **Mail Notification**. The Manage Mail Notification screen is displayed (see Figure 13.1).

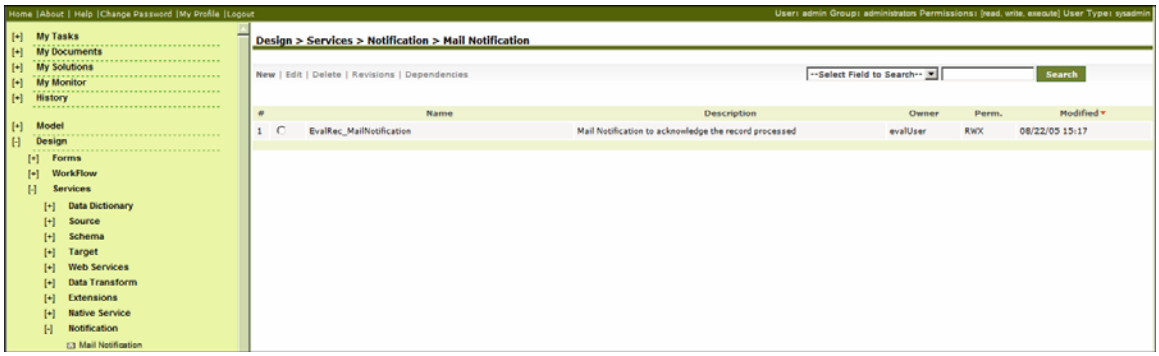


Figure 13.1: Manage Mail Notification

4. Click the **New** link. The Create Mail Notification screen is displayed (see Figure 13.2).

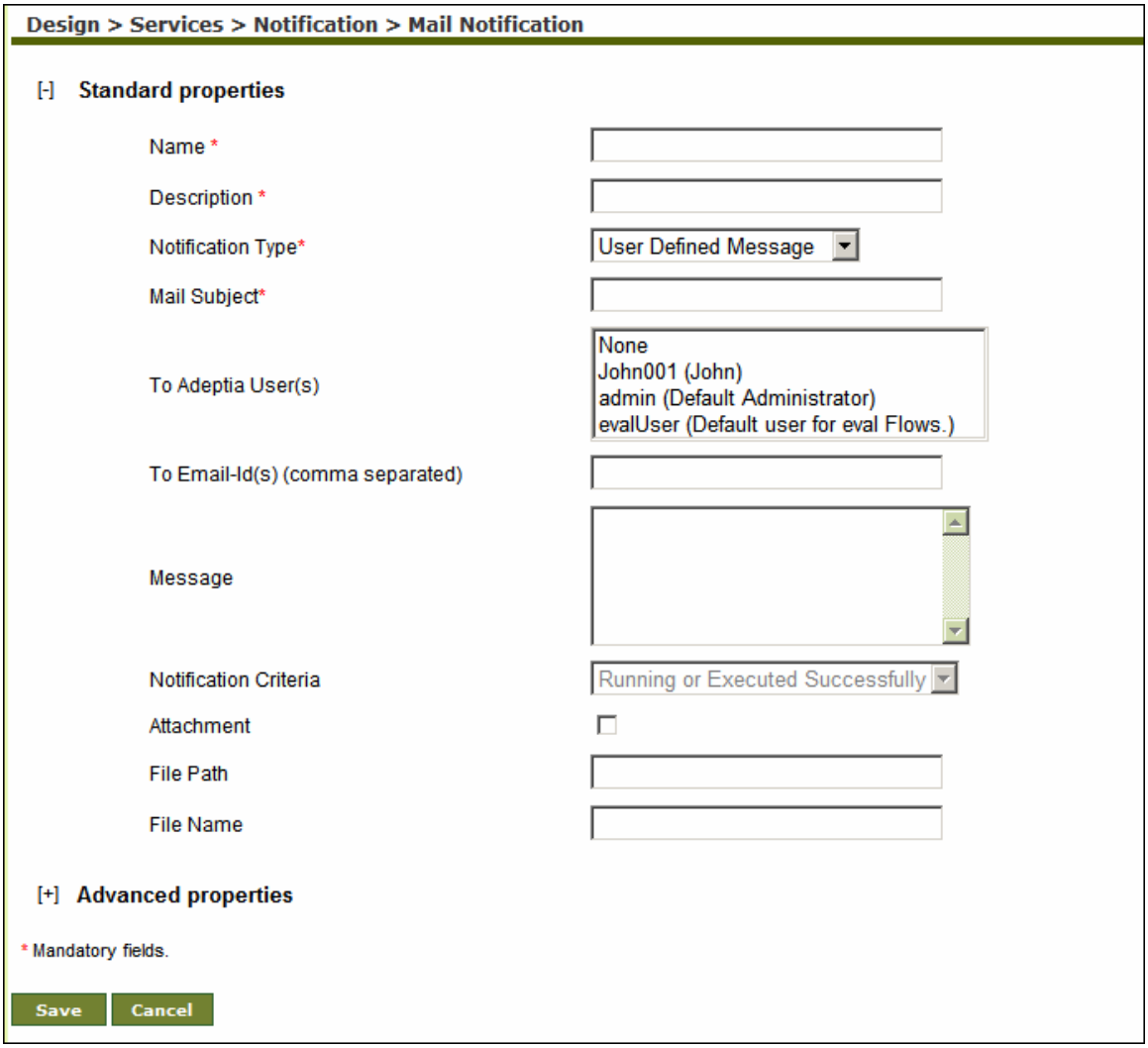


Figure 13.2: Create Mail Notification

5. Enter the name of the new Mail Notification in the *Name* field. Then, enter the description for the Mail Notification in the *Description* field.
6. Select User Defined Message from the *Notification Type* drop-down list.
7. Enter subject for the email message in the *Mail Subject* field.
8. Select the user(s) to whom you want to send the notification email from *To Adeptia user(s)* field.
9. Enter the recipient(s) email address in *To Email Id(s) (comma separated)* field.

	You can select the recipient of the notification mail either by selecting user from *To Adeptia user(s)* or by specifying email address in *To Email Id(s) (comma separated)* field. When you select user, the notification mail is sent to the email address specified while creating the user.

10. Enter the email message in the *Message* field.
11. To send a file as attachment, check the *Attachment* checkbox and enter the file path and file name in the *File Path* and *File Name* fields respectively.

	To learn about Advanced Properties refer to section **Changing Advanced Properties** section.

12. Click **Save** button. This displays a screen confirming that the mail notification activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the mail notification (refer to Figure 3.6).
13. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

14. Click **OK** to save the comments. This displays a screen confirming that the mail notification activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING NOTIFICATION TO SEND PROCESS FLOW SUMMARY

Steps to create Process Flow Summary Notification

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Notification** to expand the tree, and then click **Mail Notification**. The Manage Mail Notification screen is displayed (refer to Figure 13.1).

4. Click the **New** link. The Create Mail Notification screen is displayed (refer to Figure 13.2).
5. Enter the name of the new mail notification in the *Name* field. Then, enter the description for the mail notification in the *Description* field.
6. Select Process Flow Summary from the *Notification Type* drop-down list.
7. Enter subject for the email message in the *Mail Subject* field.
8. Select the user(s) to whom you want to send the notification email from *To Adeptia user(s)* field.
9. Enter the recipient(s) email address in *To Email Id(s) (comma separated)* field.

You can select the recipient of the notification mail either by selecting user from *To Adeptia user(s)* or by specifying email address in *To Email Id(s) (comma separated)* field.

When you select user, the notification mail is sent to the email address specified while creating the user.

10. Select one of the criteria from the *Notification Criteria* drop-down list. All criteria are explained in the table below.

Table 13.1: Notification Criteria

Notification Criteria	Description
Running or Executed Successfully	Email is sent only when the process flow is running or executed successfully.
Failure	Email is sent only when the process flow execution is failed. However, if any activity before the mail notification fails, then all activities (including Mail Notification activity) are skipped and no email is sent. Thus, it is recommended to use the Mail Notification activity with the End Event of process flow. To know how to use Mail Notification Activity with End Event of the Process Flow, refer to **Attaching End Process (Mail Notification) to Process Flow** section.
Always	Email is sent whether the process flow execution is successful or failed. However, if any activity before the mail notification fails, then all activities (including Mail Notification activity) are skipped and no email is sent. Thus, it is recommended to use the Mail Notification activity with the End Event of process flow. To know how to use Mail Notification Activity with End Event of the Process Flow, refer to **Attaching End Process (Mail Notification) to Process Flow** section.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

11. Click **Save** button. A screen is displayed where you need to enter comments related to creating the mail notification (refer to Figure 3.6).
12. Enter comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the mail notification activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

14 CREATING POLLING SERVICE ACTIVITY

Polling Services allow the process flow to 'wait' and 'listen' to a defined location, at which specific file, mail or data is to arrive or is to be modified before the execution of next activity. The Polling Services perform the 'listen' action at a frequency specified while creating the Polling activity.

There are four types of Polling Service:

- **Database Polling Service**
- **File Polling Service**
- **FTP Polling Service**
- **Mail Polling Service**

Polling Status

When a polling service is used in a process flow, it creates a *PollingStatus* activity variable. This variable is assigned value based on the execution of the polling service in the process flow. The Polling Status activity variable can take the following values:

- Timeout
- Aborted
- Success

CREATING DATABASE POLLING SERVICE ACTIVITY

The Database Polling Service activity is used to check any changes in the data stored in a database table.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Prerequisites:

- *Database Info* activity must be created before creating *Database Polling Service* activity.

Steps to create a Database Polling Service activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Polling** to expand the tree, and then click **Database**. The Manage Database Polling Service screen is displayed (see Figure 14.1).

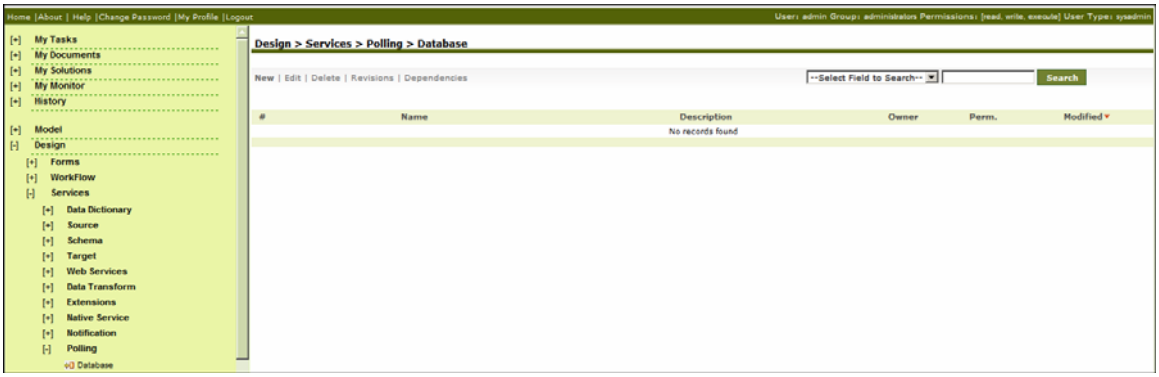


Figure 14.1: Manage Database Polling Service

4. Click the **New** link. The Create Database Polling Service screen is displayed (see Figure 14.2).

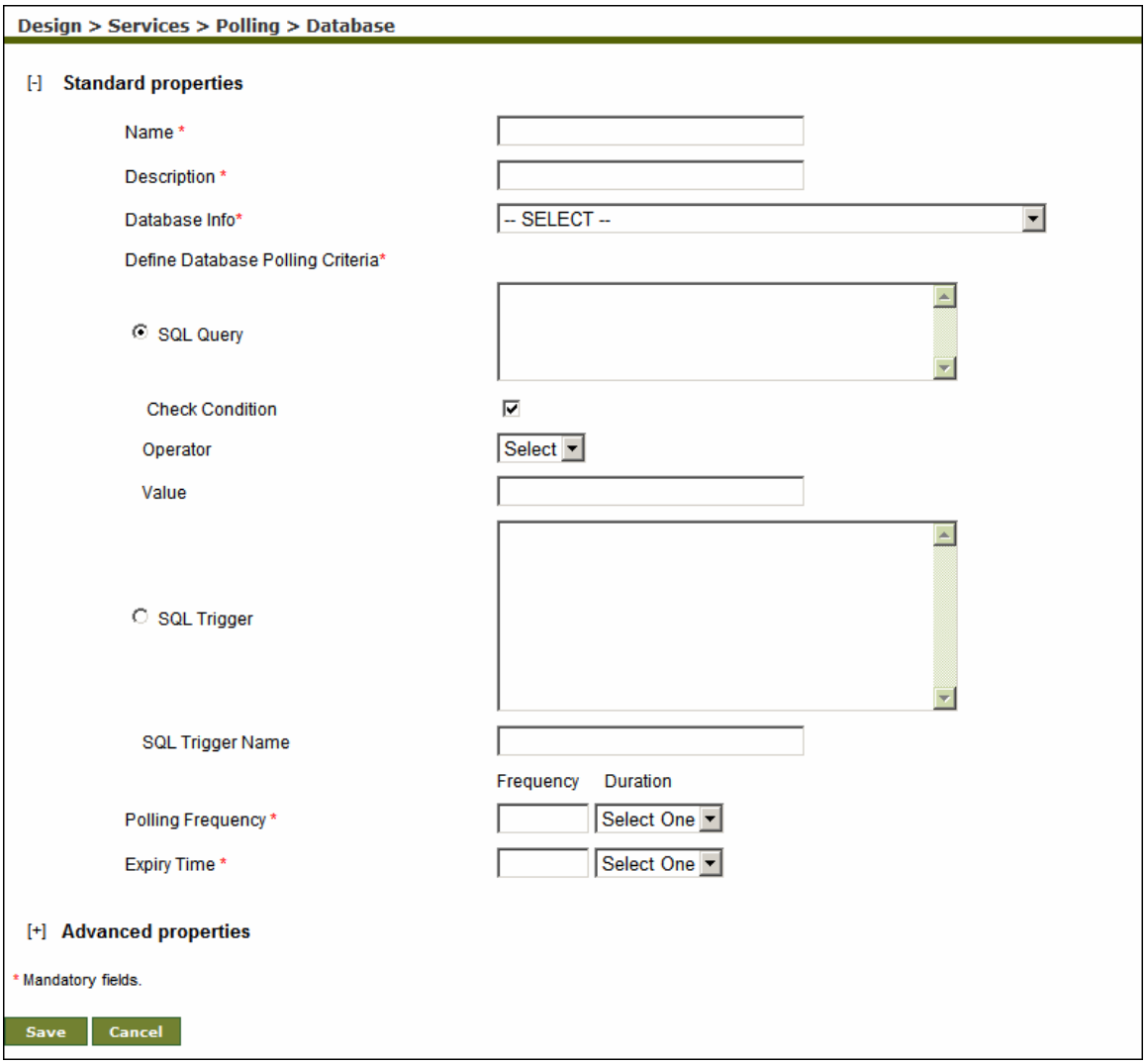


Figure 14.2: Create Database Polling Service

5. Enter the name and description of the new Database Polling Service in the *Name* and *Description* fields respectively.

6. Select the Database Info Id activity from the *Database Info Id* drop-down list.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administrator Guide*.

7. You can create the database event definition by entering an SQL Query or a database trigger command. By default, *SQL Query* option is selected. Enter the query in the *SQL Query* field. Select the operator for the query from the *Operator* drop-down list. Enter the value to be compared in the query in the *Value* field. The query should return only one record. If the query returns multiple records, then only the first record is accepted. If the query returns one record, then it will compare the value of the first field with the value specified in the *Value* field.

8. Alternately, enter the database trigger command in the *SQL Trigger* field.

Following is the format of SQL trigger

<Trigger Text>
INSERT INTO dbpollingtable VALUES ('Query =<WHERE CLAUSE>');
END <trigger name> ;

Edit the parts, which are within < >. You can define a 'Where' clause that indicates the row that is updated. When the command is parsed, it will return the updated row from the database source.

Do not delete the Insert query.

<trigger name> after the END tag should be used for Oracle only. In case of SQL server, <trigger name> is not needed.

Following is the example of the trigger used for SQL Server :

```
create trigger Trigger_test on emp for
insert,update
as
declare @empname varchar(20)
begin
set @empname=(select empname from inserted);
INSERT INTO dbpollingtable VALUES ('Query =WHERE empname='''
+@empname+''');

 END  ;
```

Following is the example of the trigger used for Oracle:

```
CREATE OR REPLACE TRIGGER Trigger_test
AFTER INSERT OR UPDATE ON Emp FOR EACH ROW  BEGIN
INSERT INTO dbpollingtable VALUES ('Query = where rowid= ' || :new.rowi
d );  END Trigger_test;
```

Here :

Trigger_test is name of the trigger.

Emp is the name of the user table on which insert or update operation has to be done.

dbpollingtable is the name of the temporary table used. Do not change it.

9. Enter the name of Trigger in the *SQL Trigger Name* field.

10. Enter the time interval for Polling in the *Polling Frequency* field. Enter the digit in the *Frequency* field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.
11. Enter the expiry time in the *Expiry Time* field. After expiry time process flow does not poll for data.

	Recommended minimum Polling Frequency is 30 seconds. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

	When a database polling service is used in a process flow, it creates a *Polling Status activity* variable. For details, refer to **Polling Status** section.

12. Click **Save** button. This displays a screen confirming that the database polling service has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the database polling service (refer to Figure 3.6).
13. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

14. Click **OK** to save the comments. This displays a screen confirming that the database polling service has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING FILE POLLING SERVICE ACTIVITY

The Polling Service activity is used to check the arrival or modification of file(s) on the Local LAN location.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a File Polling Service

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

3. Click **[+] Polling** to expand the tree, and then click **File**. The Manage File Polling Service screen is displayed (see Figure 14.3).

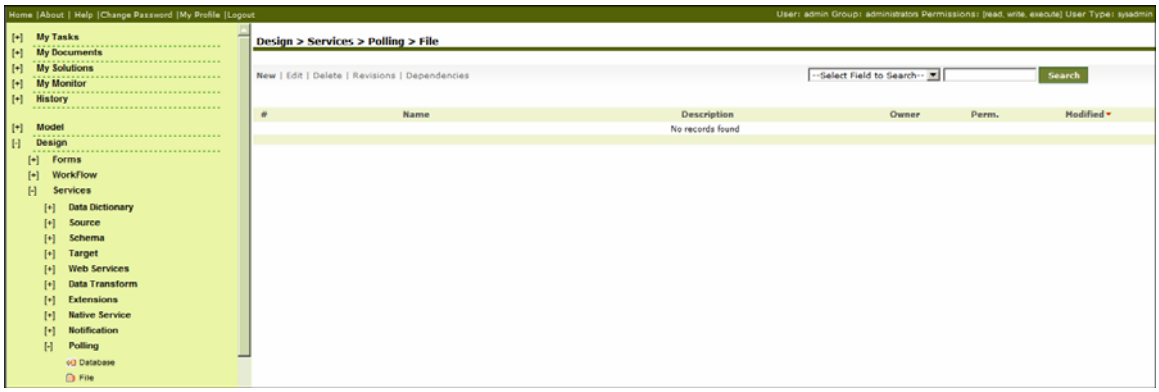


Figure 14.3: Manage File Polling Service

4. Click the **New** link. The Create File Polling Service screen is displayed (see Figure 14.4).

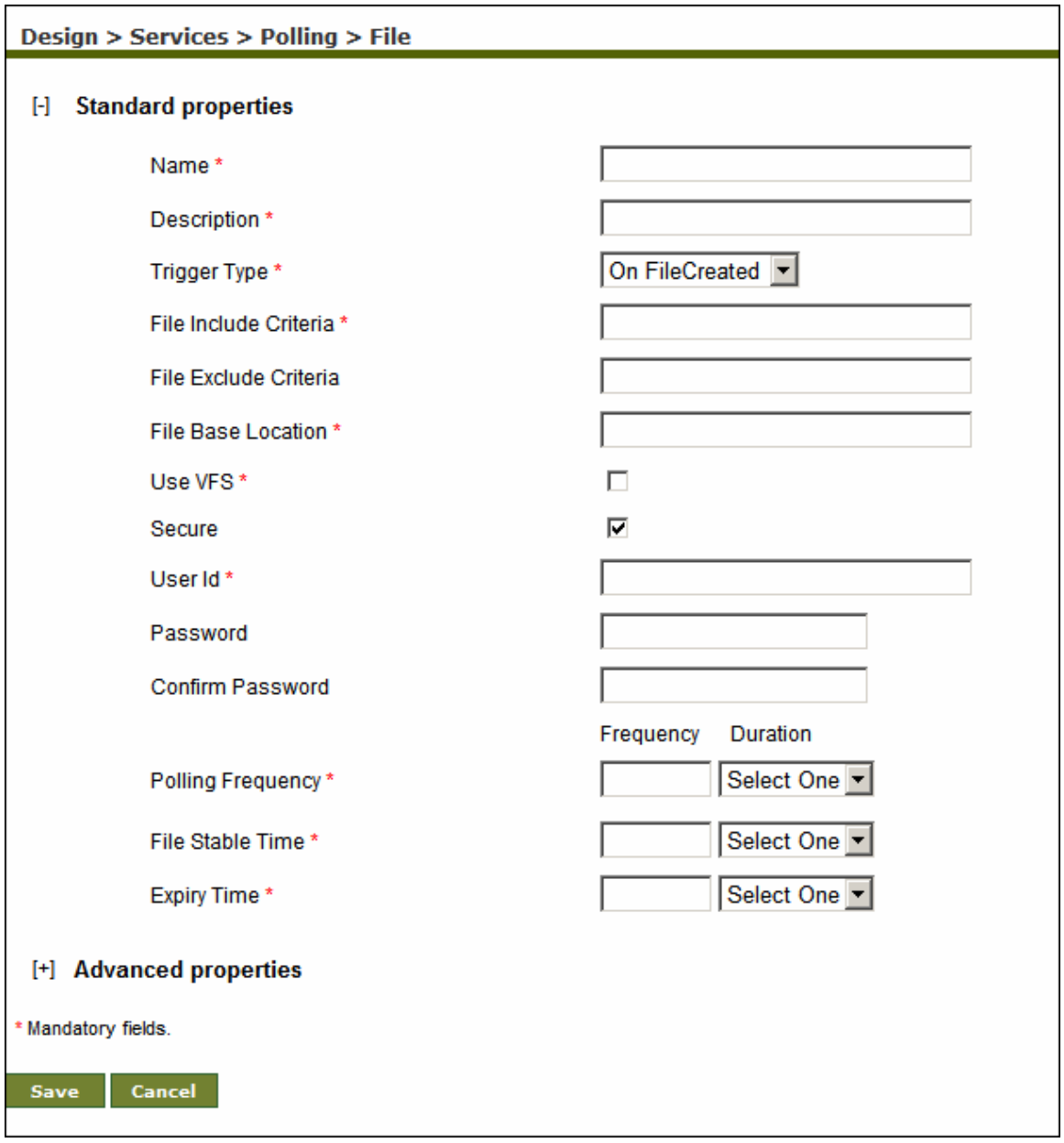


Figure 14.4: Create File Polling Service

5. Enter the name and description of the new File Polling Service in the *Name* and *Description* fields respectively.

6. Select the trigger type from the *Trigger Type* drop-down list. The effect on the selection is listed in the table below.

Table 14.1: Trigger Type Selection Values

Trigger Type Selection	Description
On FileCreated	To configure the File Polling service to check for the

	creation of a new file(s).
On FileExists	To configure the File Polling service to check for the existence of the file(s)
On FileModified	To configure the File Polling service to check for any modification in file(s)

7. Enter the name of file that File Polling service needs to verify, in *File Include Criteria* field.

8. Enter the name of file that File Polling service does not need to verify, in *File Exclude Criteria* field. For example **.txt* is entered in *File Include Criteria*, but two files *Gdata.txt* and *Gdata1.txt* are not required to be verified by File Polling service. Then *Gdata.txt* and *Gdata1.txt* file name need to be entered separated by comma in *File Exclude Criteria* field. You can use regular expressions listed in the table below.

Table 14.2: Expressions used in File Include Criteria and File Exclude Criteria

Expression	Description
.	For all files with some extension
*	For all files in a directory
a*.txt	For files starting with a and having extension txt (e.g. arch.txt)
a??????.txt	For files starting with a and have 6 more character followed by txt extension (e.g. archive.txt)
a[1-9]	For a1, a2 ,a3 ……..a9
b[aiu]t	For bat, bit or but
a.txt, a.doc	For two files named as a.txt and a.doc

9. Enter the path of file in *File Base Location*. Example c:/Gmdata.

You can also use regular expression for folders, in *File Include Criteria* and *File Exclude Criteria* fields. For example, if you enter *h*/*.txt* in *File Include Criteria* field and *C:/Gmdata* in *File Base Location* field, it will search for all .txt file inside all directories which starts from h under C:\Gmdata.

10. When Adeptia Server is installed on Windows Operating System, File Polling uses windows service to connect to remote machine to access any file. It just connects once and uses the same connection with the same User ID and Password (which is stored in the cache) every time. If you want to enforce the validation of User ID and Password every time while accessing the file on a remote machine, check *Use VFS* checkbox.

11. Enter username and password in the *User ID* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

12. Enter the time interval, the file polling service will check for the arrival of any file or modification of existing file in the *Polling Frequency* field. Enter the digit in the *Frequency* field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

Recommended minimum Polling Frequency is 30 seconds.

13. Enter the file stable time in the *File Stable Time* field. This is applicable only when user selects On *FileModified* in trigger type. Polling will wait for the above specified time to become a file stable.
14. Enter the expiry time in the *Expiry Time* field. After expiry time process flow does not poll for the file.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

	When a file polling service is used in a process flow, it creates a *Polling Status activity* variable. For details, refer to **Polling Status** section.

15. Click **Save** button. This displays a screen confirming that the file polling service has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the file polling service (refer to Figure 3.6).
16. Enter **the** comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

17. Click **OK** to save the comments. This displays a screen confirming that the file polling service has been **created** successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING FTP POLLING SERVICE ACTIVITY

The FTP Polling Service activity is used to check the arrival or modification of a file(s) on a FTP location.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a FTP Polling Service

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

3. Click **[+] Polling** to expand the tree, and then click **FTP**. The Manage FTP Polling screen is displayed (see Figure 14.5).

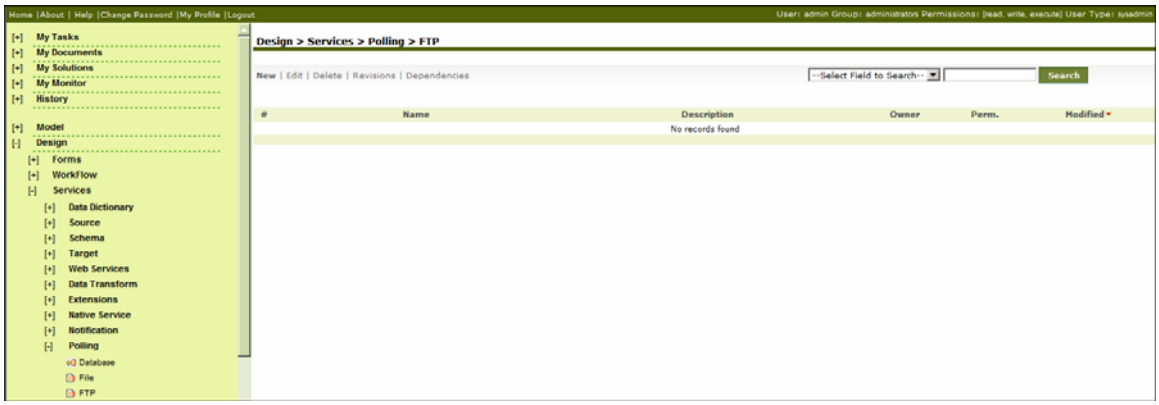


Figure 14.5: Manage FTP Polling Service

4. Click the **New** link. The Create FTP Polling Service screen is displayed (see Figure 14.6).

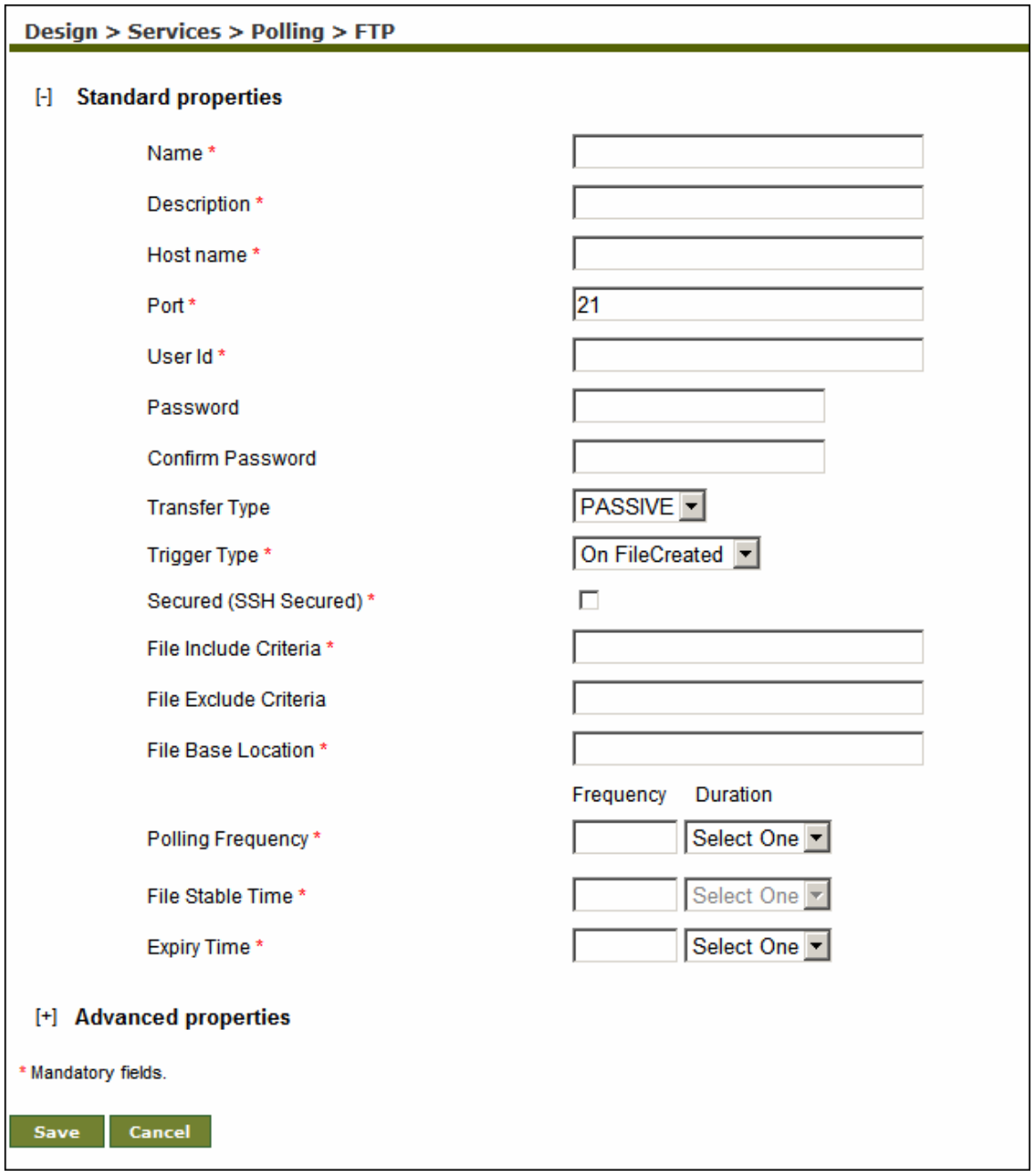


Figure 14.6: Create FTP Polling Service

5. Enter the name and description of the new FTP Polling Service in the *Name* and *Description* fields respectively.
6. Enter the name and port number of the FTP Server in the *Host Name* and *Port* fields respectively.
7. Enter username and password required to access FTP Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

8. Select the transfer type as either *Active* or *Passive* from the *Transfer Type* drop-down list. *Active* transfer is more secure since the client only initiates communication to the Server on one port whereas in case of *Passive* transfer the client initiates communication to the Server over two ports. Passive mode is useful when you are behind a firewall or a proxy.
9. Select the trigger type from the *Trigger Type* drop-down list. For selection of values, refer to Table 14.1.
10. Check the *Secured (SSH Secured)* checkbox if the specified FTP server is a secured server.
11. Enter the name of file that FTP event needs to verify in the *File Include Criteria* field.
12. Enter the name of file that file event does not need to verify in the *File Exclude Criteria* field. For example **.txt* is entered in *File Include Criteria*, but two files *Gdata.txt* and *Gdata1.txt* are not required to be verified by FTP Polling Service. Then *Gdata.txt* and *Gdata1.txt* file name need to be entered separated by comma in *File Exclude Criteria* field. You can use regular expressions listed in Table 14.2.
13. Enter the path of file in *File Base Location*. Example c:/Gmdata.

You can also use regular expression for folders, in *File Include Criteria* and *File Exclude Criteria* fields. For example, if you enter *h*/*.txt* in *File Include Criteria* field and *C:/Gmdata* in *File Base Location* field, it will search for all .txt file inside all directories which starts from h under C:\Gmdata.

14. Enter the time interval, the FTP Polling Service will check for the arrival of any file or modification of existing file in the *Polling Frequency* field. Enter the digit in the *Frequency* field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

Recommended minimum Polling Frequency is 30 seconds.

15. Enter the file stable time in the *File Stable Time* field. This is applicable only when user selects On *FileModified* in trigger type. Polling will wait for the above specified time to become a file stable.
16. Enter the expiry time in the *Expiry Time* field. After expiry time process flow does not poll for the file.

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

When a FTP polling service is used in a process flow, it creates a *Polling Status activity* variable. For details, refer to **Polling Status** section.

17. Click **Save** button. This displays a screen confirming that the FTP polling service has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the FTP polling service (refer to Figure 3.6).
18. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

19. Click **OK** to save the comments. This displays a screen confirming that the FTP polling service has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING MAIL POLLING SERVICE ACTIVITY

The Mail Polling activity is used to 'listen' for the arrival of any mails on the mail Server.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a Mail Polling activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Polling** to expand the tree, and then click **Mail**. The Manage Mail Polling screen is displayed (see Figure 14.7).

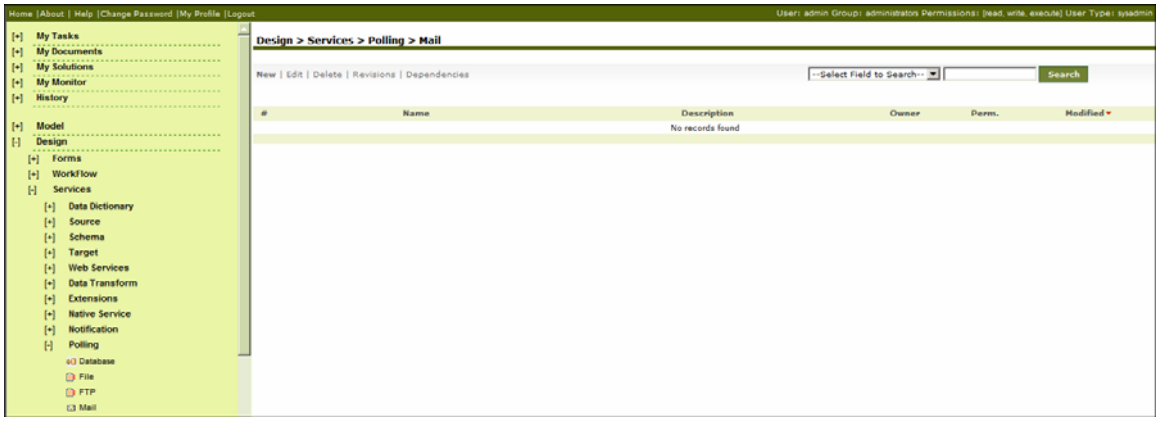


Figure 14.7: Manage Mail Polling Service

4. Click the **New** link. The Create Mail Polling screen is displayed (see Figure 14.8).

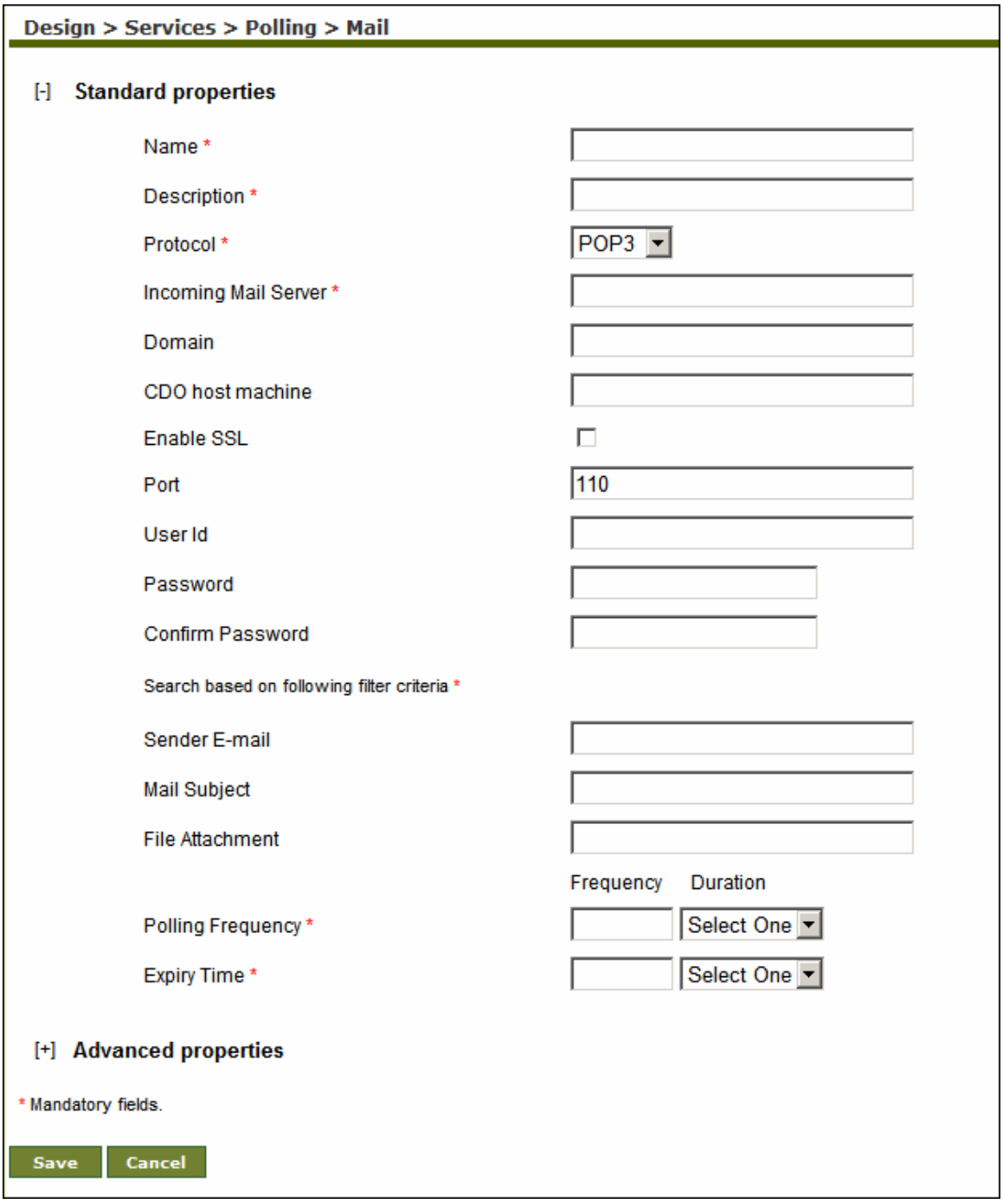


Figure 14.8: Create Mail Polling Service

5. Enter the name and description of the new Mail Polling activity in the *Name* and *Description* fields respectively.

6. Select the Internet standard protocol to be used for retrieving incoming mails, from the *Protocol* drop-down list. You can select either the *POP3*, *IMAP4* or the *MAPI* protocol. Based

on the selected protocol, the default port number for that protocol is displayed in the *Port* field.

7. Enter the address of the incoming mail server in the *Incoming Mail Server* field.

	To access mails from Microsoft Exchange Server use *MAPI* in the *Protocol* drop-down list. To connect Adeptia Server with Microsoft Exchange Server, you need to buy a third party tool called *J-Integra for Exchange*. J-Integra for Exchange is a high performance middleware bridge that enables Java Exchange interoperability. If you want to retrieve mails from an Exchange Server using J-Integra, select *MAPI* in the *Protocol* drop-down list. If *MAPI* is selected in the *Protocol* drop-down list: ▪ Enter name of the exchange server in *Incoming Mail Server* field ▪ Enter the domain name in the *Domain* field ▪ Enter the name of the CDO host machine in *CDO host machine* field. CDOConfig.exe is a tool that comes with the J-Integra for Exchange SDK and is used for configuring CDO. Host where CDO is configured is called CDO host machine. For detailed information about Jintegra for Exchange, refer to **http://j-integra.intrinsyc.com/products/exchange/**.

8. Check the *Secure* checkbox, if the specified incoming mail server is SSL enabled.
9. The default port number of the selected protocol is displayed in the *Port* field. If you want to change this port number, enter the new port number in the *Port* field.
10. Enter the user Id and password of mail server in the *User ID* and *Password* fields respectively. Then re-enter the password in the *Confirm Password* field.
11. Select any of the following filter criteria:
    - Sender E-mail
    - Mail Subject
    - File Attachment

    You may select more than one filter criteria.
12. Enter the sender's email address and subject of email in the *Sender Email* and *Mail Subject* fields respectively.
13. Enter the name of the file attached with mail, in the *File Attachment* field.
14. Enter the time interval for Polling in the *Polling Frequency* field. Enter the digit in the Frequency field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

	Recommended minimum Polling Frequency is 30 seconds.

15. Enter the expiry time in the *Expiry Time* field. After expiry time process flow does not poll for the mail.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

	When a mail polling service is used in a process flow, it creates a *Polling Status activity* variable. For details, refer to **Polling Status** section.

16. Click **Save** button. This displays a screen confirming that the mail polling service has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the mail polling service (refer to Figure 3.6).
17. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

18. Click **OK** to save the comments. This displays a screen confirming that the mail polling service has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

15 MISCELLANEOUS

This section allows you to create the following activities:

- Context Download
- Context Upload
- Stored Procedure

CREATING CONTEXT DOWNLOAD ACTIVITY

Context Download activity is used to generate XML from a context variable. This is required when the context variable's information is to be stored in some target. This information can be passed as stream to other activities. For example, if you want to store variables and their values present in the context as per schema defined to any target activity like database target, file target, then context download activity can be used. It will generate XML of context variables and their values, which can be further used.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to Create Context Download activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Miscellaneous** to expand the tree, and then click **Context Download**. The Manage Context Download screen is displayed (see Figure 15.1)

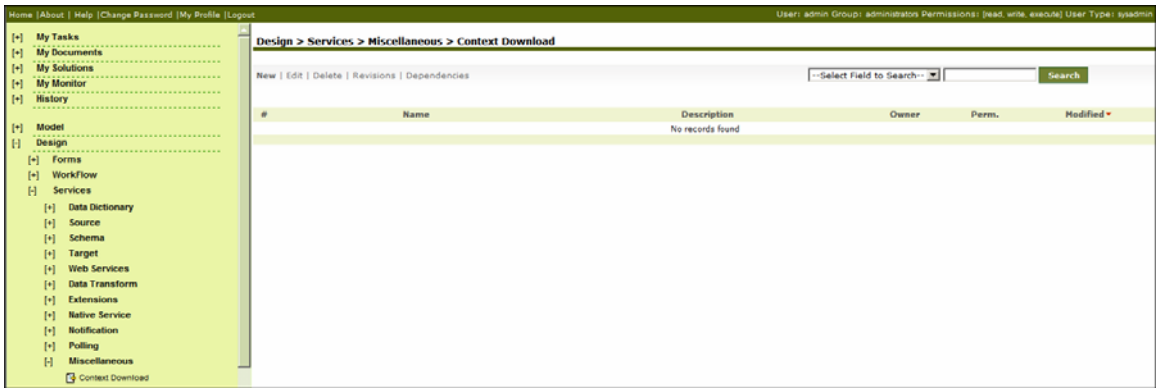


Figure 15.1: Manage Context Download

4. Click **New** link. The Create Context Download screen is displayed (see Figure 15.2).

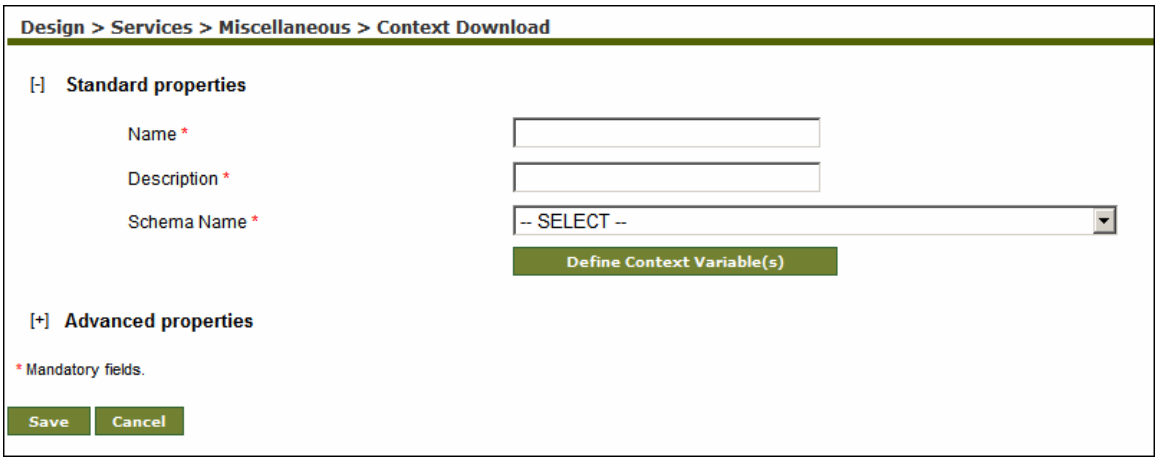


Figure 15.2: Create Context Download

5. Enter the name and description for Context Download in the *Name* and *Description* fields respectively.
6. Select the schema activity from the *Schema Name* drop-down list.
7. To define the context variable, click **Define context variable(s)** button. The Map Context Variable screen is displayed (see Figure 15.3).

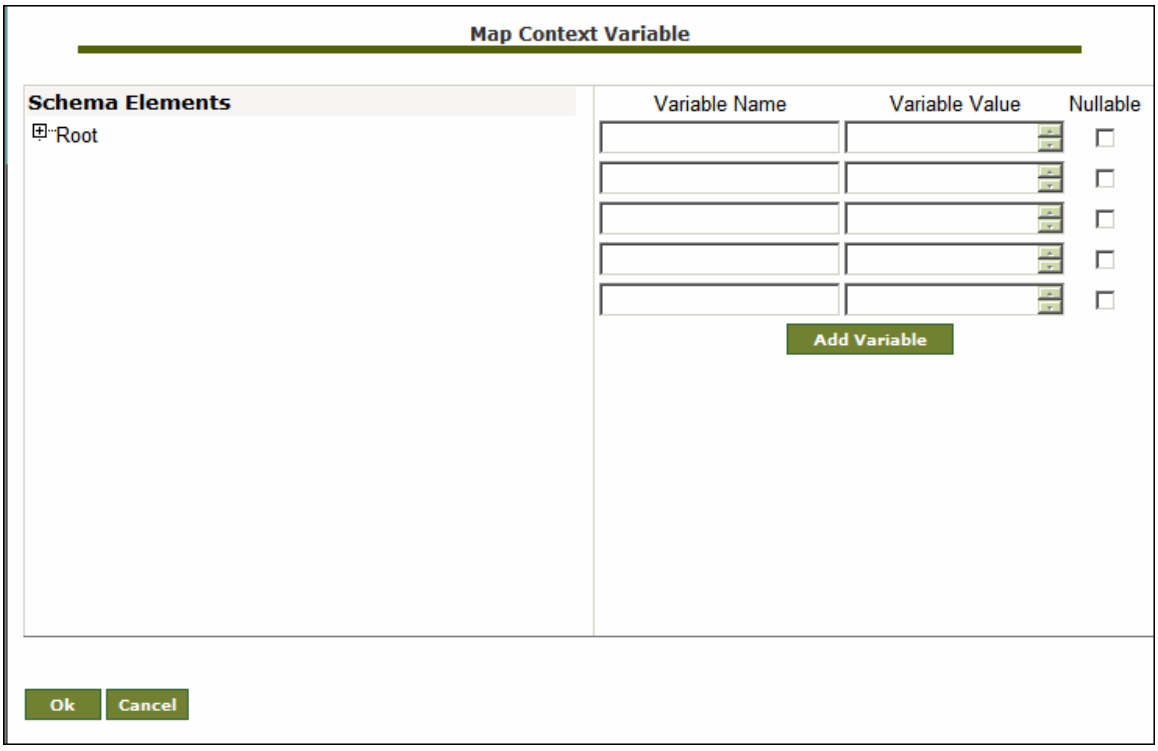


Figure 15.3: Map Context Variable

8. Expand the selected schema by clicking **[+]**. All fields of selected schemas are displayed (see Figure 15.4).

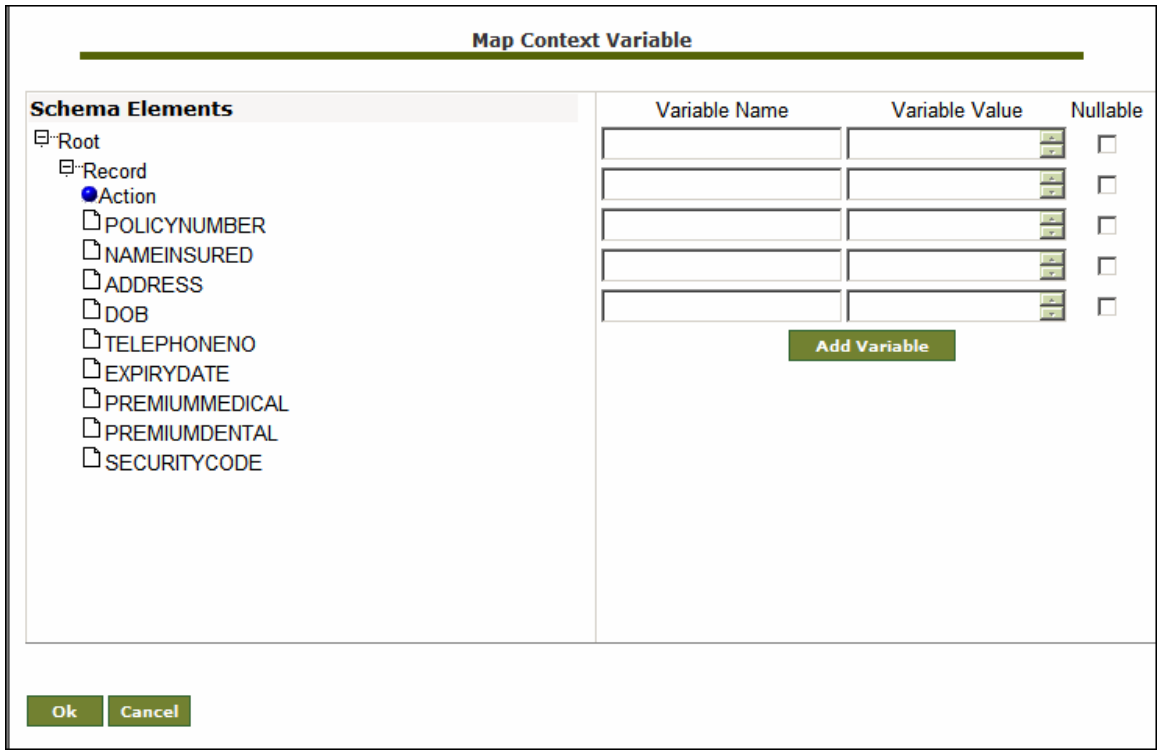


Figure 15.4: Expand Schema

9. Enter a name for the context variable in the *Variable Name* field (see Figure 15.5).

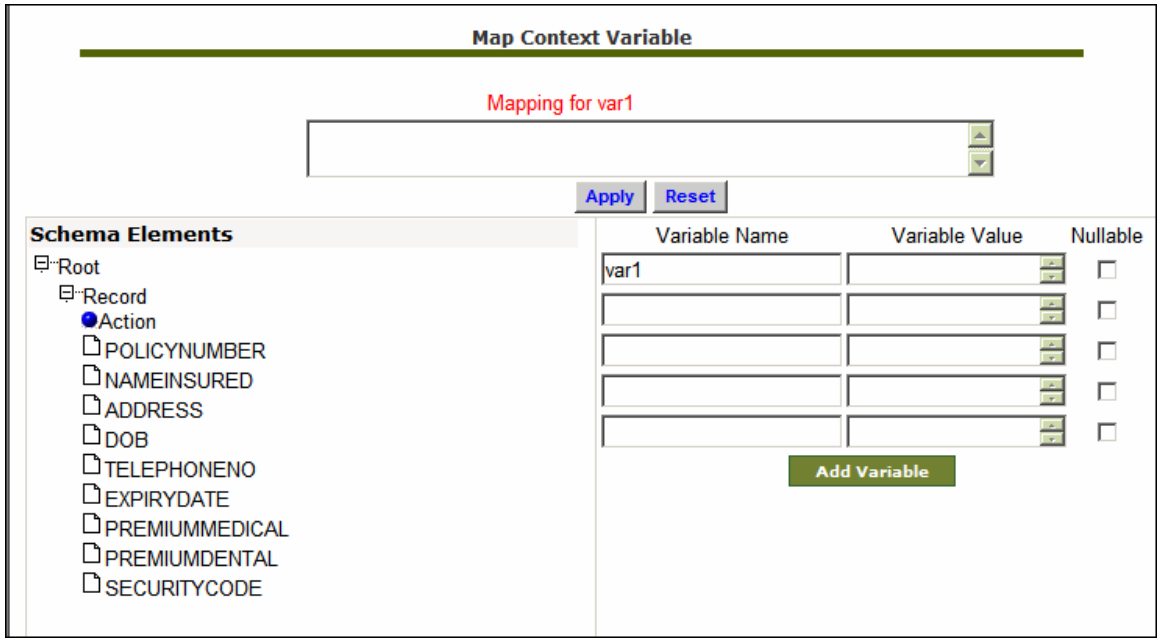


Figure 15.5: Enter Variable Name

10. Select the schema element, which you want to map with the above defined variable. The XPath of the selected element is displayed in the Mapping field (see Figure 15.6).

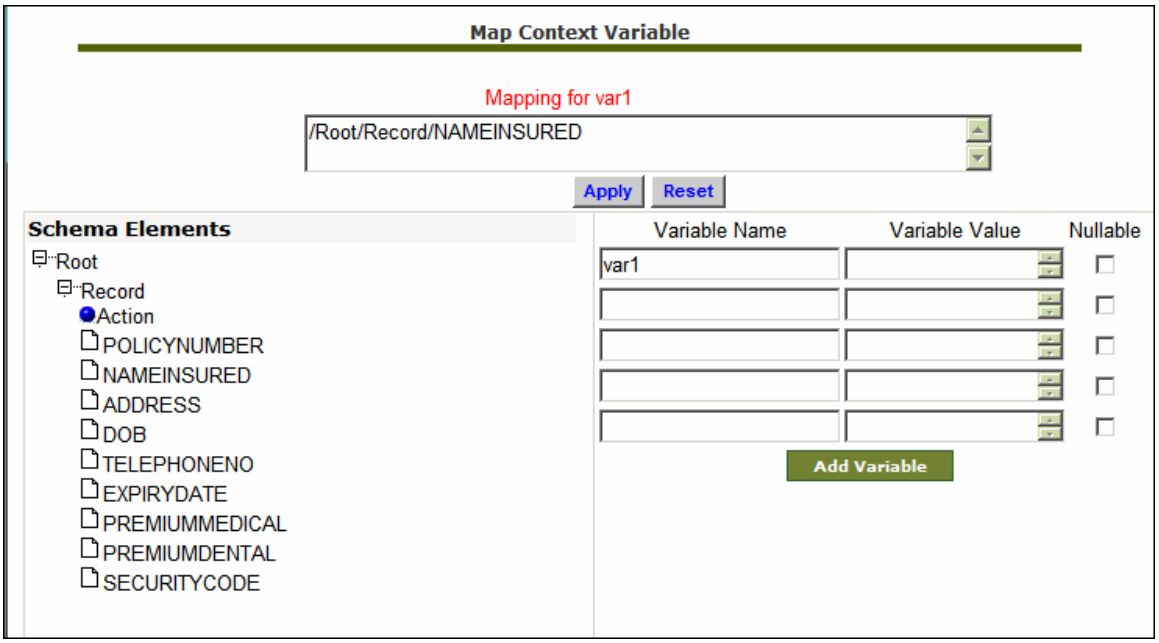


Figure 15.6: Map Schema Element with Variable

11. Click **Apply** button. Value of the selected schema element is displayed in the *Variable Value* field (see Figure 15.7).

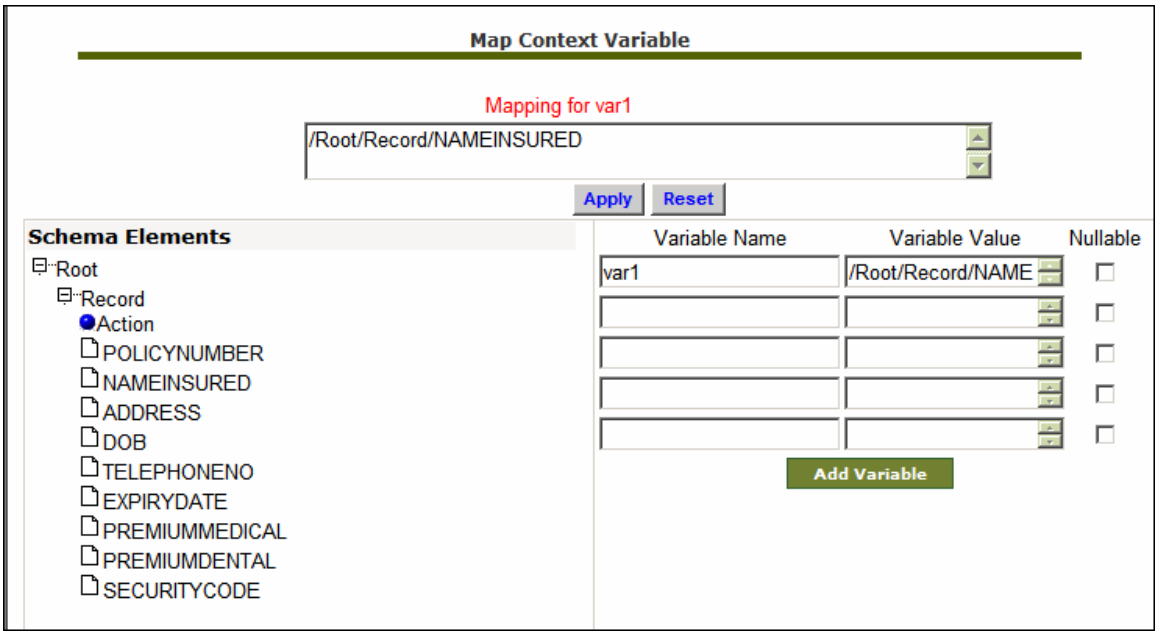


Figure 15.7: Apply Map

12. Repeat steps from 8 to10 to map the context variable with other elements.
13. Check the *Nullable* checkbox in case you want this variable to allow blank value.
14. Click **OK** button to return to the Create Context Download screen.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

15. Click **Save** button. This displays a screen confirming that the Context Download activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the context download activity (refer to Figure 3.6).
16. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

17. Click **OK** to save the comments. This displays a screen confirming that the Context download activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING CONTEXT UPLOAD ACTIVITY

Context Upload is used to map any field of a schema to the Process Flow Context Variable. Once the variable is set in process flow context, it can be used by any activity in the Process Flow.

Context Upload variables can be created for all types of schemas. In case of XML Schema, only those XML schemas, which are created by uploading XSD file, or DTD file (with Convert to XSD option enabled) can be used to create Context Upload variables.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Steps to Create Context Upload activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Miscellaneous** to expand the tree, and then click **Context Upload**. The Manage Context Upload screen is displayed (see Figure 15.8)

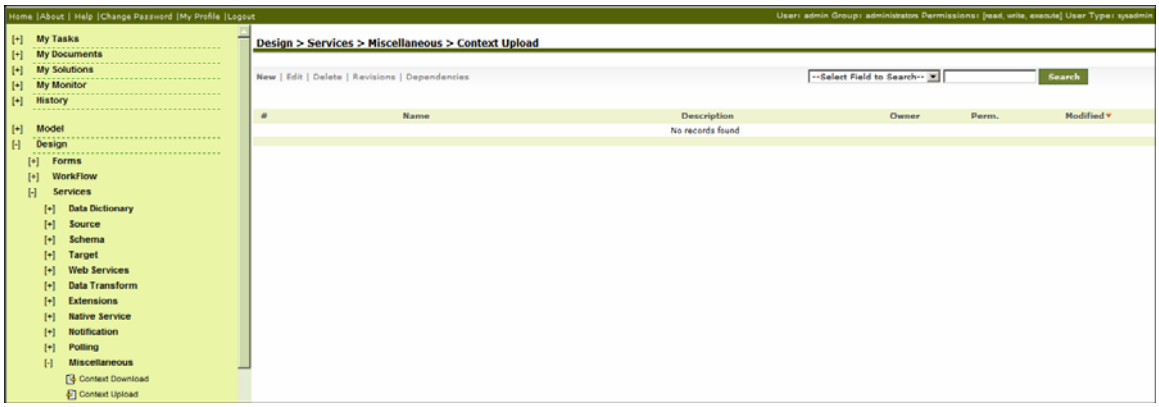


Figure 15.8: Manage Context Upload

4. Click **New** link. The Create Context Upload screen is displayed (see Figure 15.9).

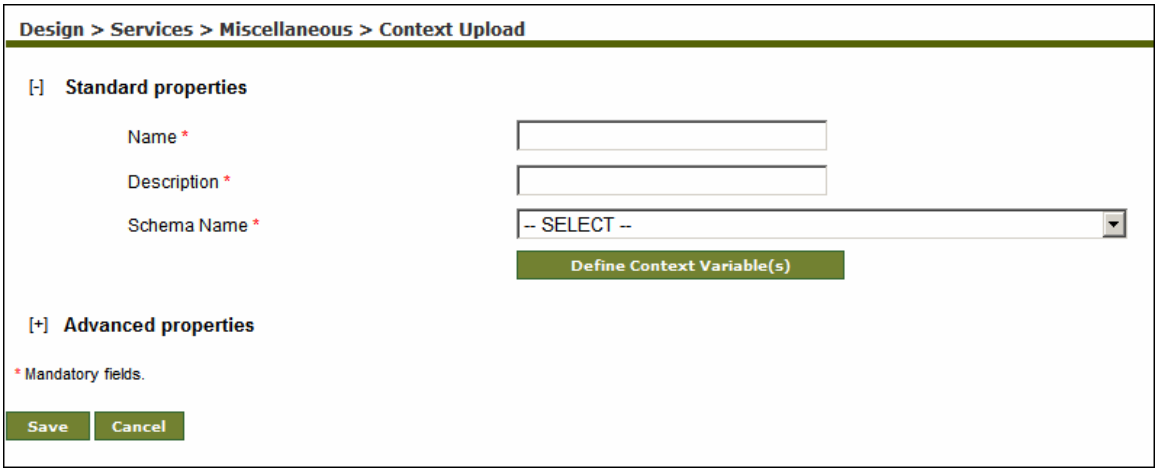


Figure 15.9: Create Context Upload

5. Enter the name and description for Context Upload in the *Name* and *Description* fields respectively.
6. Select the schema activity from the *Schema Name* drop-down list.
7. To define the context variable, click **Define context variable(s)** button. The Map Context Variable screen is displayed (refer to Figure 15.3).
8. Expand the selected schema by clicking **[+]**. All fields of selected schemas are displayed (refer to Figure 15.4).
9. Enter a name for the context variable in the *Variable Name* field (refer to Figure 15.5).
10. Select the schema element, which you want to map with the above defined variable. The XPath of the selected element is displayed in the Mapping field (refer to Figure 15.6).

	The Context Upload variable accepts only string and list as parameters. When it is used in a process flow, then you need to specify the XPath value. If you select 'String' as the Data Type property, then the first XPath value is used.

11. Click **Apply** button. Value of the selected schema element is displayed in the *Variable Value* field (refer to Figure 15.7).
12. Repeat step 8 to11 to map the context variable with other elements.
13. Click **OK** button to return to the Create Context Upload screen.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

14. Click **Save** button. This displays a screen confirming that the Context Upload activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the context upload activity (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the Context Upload activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

CREATING STORED PROCEDURE

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. Stored procedures are used to encapsulate a set of operations or queries to execute on a database server. Stored procedures can be compiled and executed with different parameters and they may have any combination of input, output, and input/output parameters.

Using Adeptia Server's Stored Procedure activity, you can execute a database stored procedure. Stored Procedure activity takes IN parameters from process flow context. So you need to set the value of IN parameters in the process flow context. For Example if the selected stored procedure has IN parameters as *empid*, you have to set an input variable in process flow context with the name *Service.<ActivityName>.InParams.empid,* where *<Activity Name>* is the name of the stored procedure activity, which is used in the process flow. You can set this input variable using put-context-var, Process flow Variable or custom plugin.

Similarly OUT parameters of the stored procedure are set to process flow context. For example if the selected stored procedure has OUT parameters as *salary*, the output variable is set in the process flow context as *Service.<ActivityName>.OutParams.salary*, where *<Activity Name>* is the name of the activity, which is used to execute the process flow. You can further use this variable and its value is the process flow. Value of output variable set by stored procedure activity, is always in string format. You need to type cast it in required format. Stored procedure doesn't generate a stream. So you have to use context source activity after stored procedure activity. To know how to create a context source, refer to the section **Using Context Source and Context Target.**

Variable name for INOUT parameter will be *Service.<ActivityName>.InOutParams.count,* where *<Activity Name>* is the name of the activity, which is used to execute the process flow and count is the INOUT parameter of the stored procedure.

Limitation:

- Adeptia Server stored procedure activity is supported for MS SQL, Oracle and Sybase version 9.0.2.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Prerequisites:

- *Database Info* activity must be created before creating *Stored Procedure* activity.

Steps to create Stored Procedure activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Miscellaneous** to expand the tree, and then click **Stored Procedure**. The Manage Stored Procedure screen is displayed (see Figure 15.10).

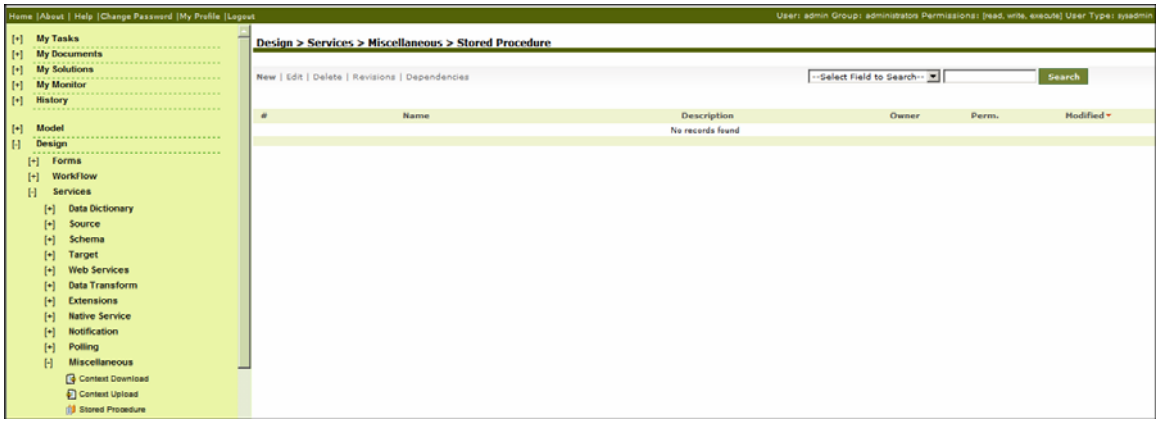


Figure 15.10: Manage Stored Procedure

4. Click the **New** link. The Create Stored Procedure screen is displayed (see Figure 15.11).

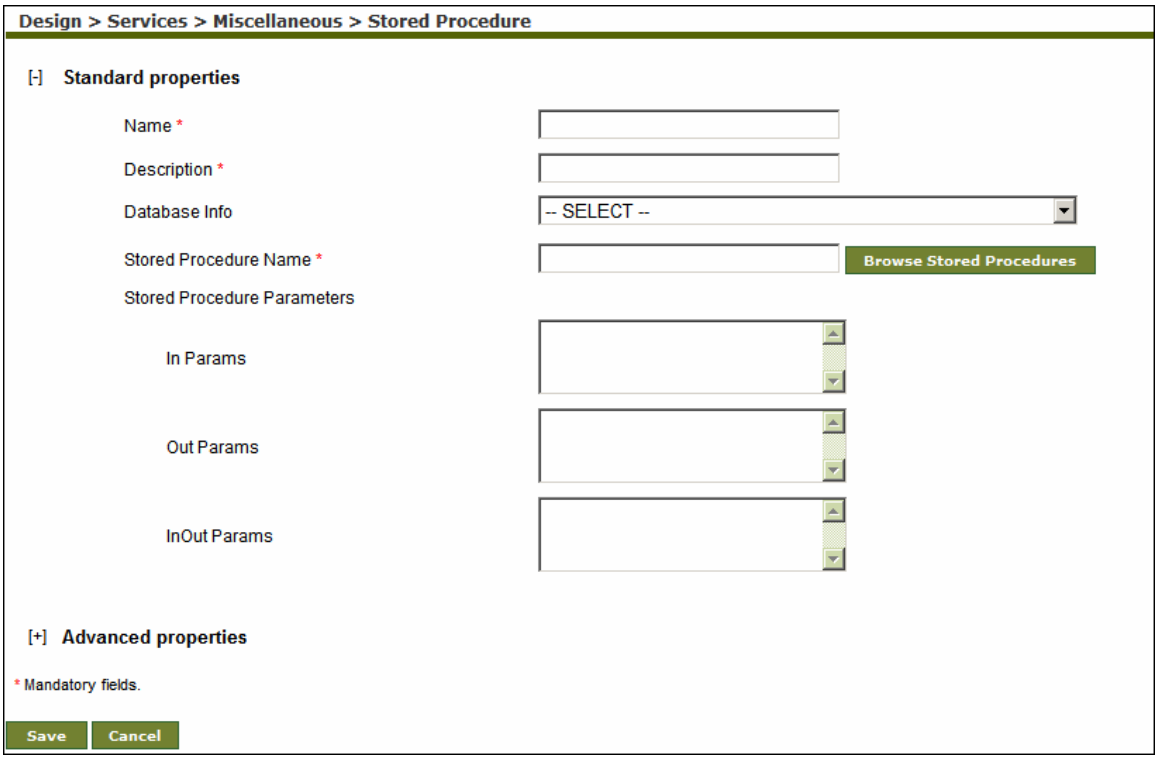


Figure 15.11: Create Stored Procedure

5. Enter the name and description for new stored procedure activity in the *Name* and *Description* fields respectively.

6. Select the *database* info activity from the *Database Info Id* drop-down list.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administrator Guide*.

7. To select the database stored procedure, click **Browse Stored Procedure** button. The Select Stored Procedure screen with list of stored procedure is displayed (see Figure 15.12)

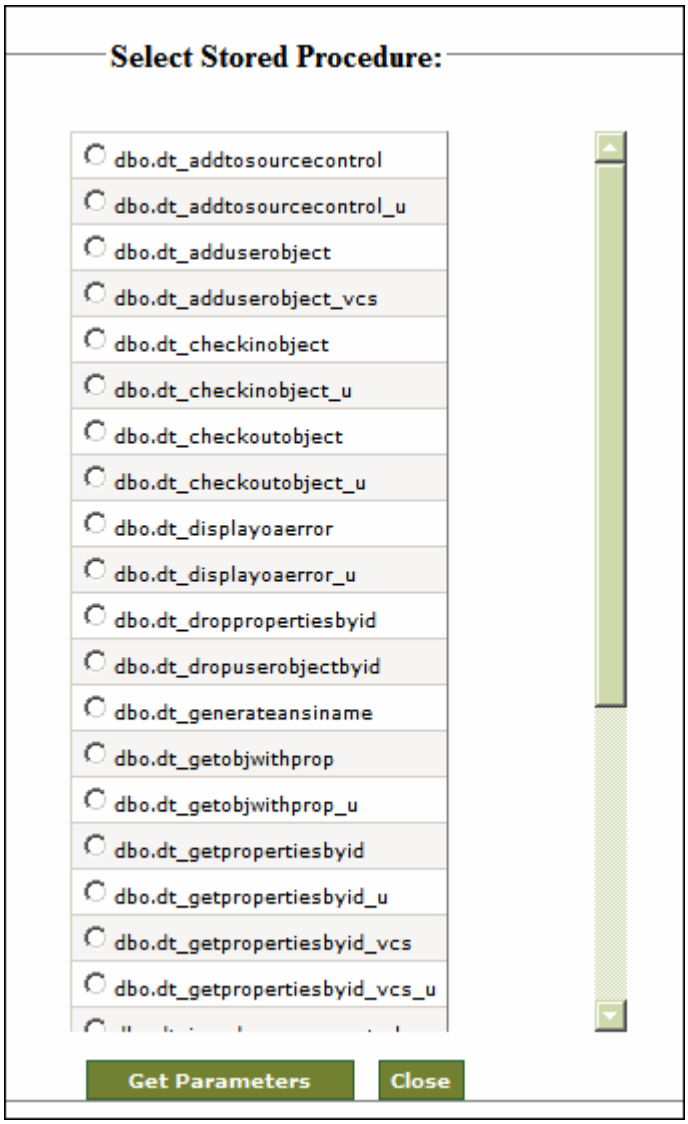


Figure 15.12: Select Stored Procedure

8. Select the required stored procedure and click **Get Parameters** button. The stored procedure parameter screen is displayed with list of parameters (see Figure 15.13).

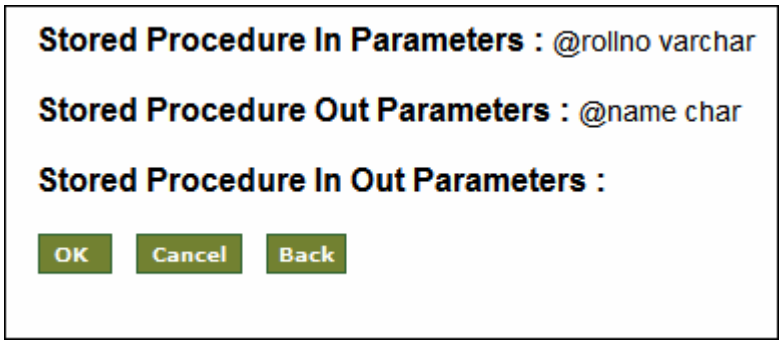


Figure 15.13: View Parameters

9. Click **OK** to return to Stored Procedure activity screen. Parameters of the selected procedure are populated in *Stored Procedure Parameters* field (see Figure 15.14).

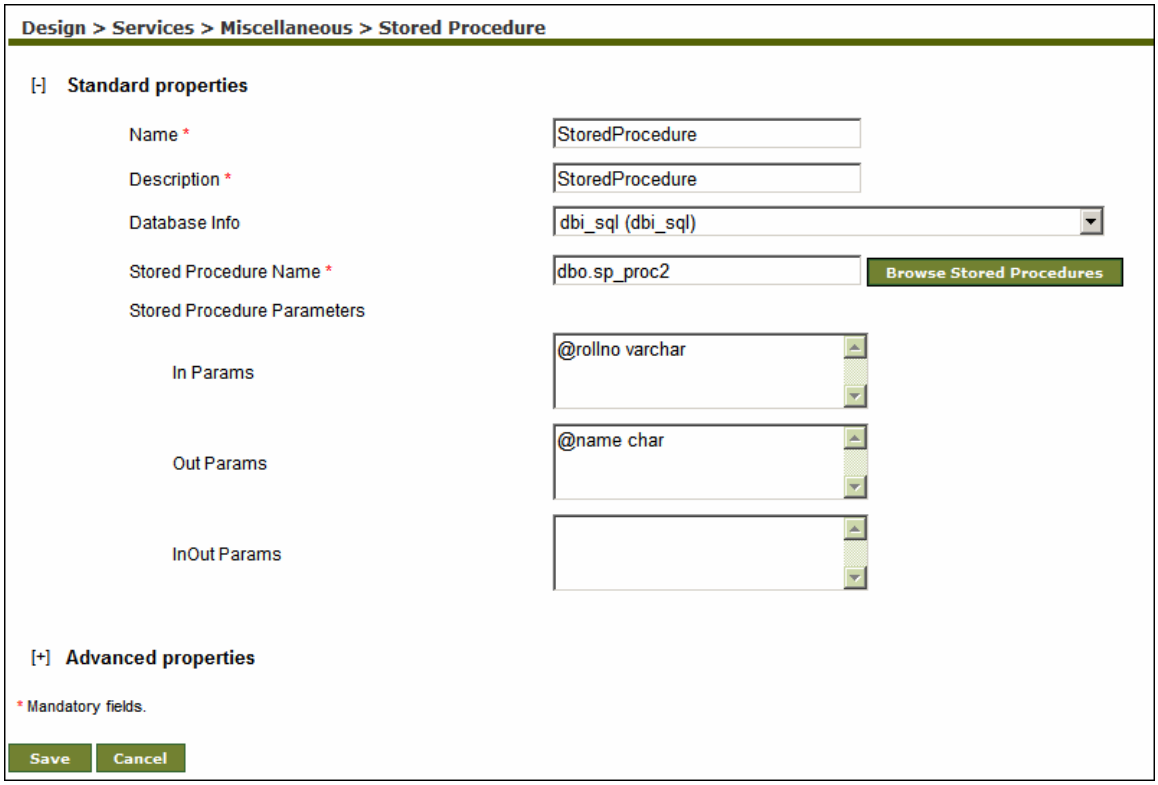


Figure 15.14: Manage Stored Procedure

10. In the Stored Procedure activity screen click **Save** button. This displays a screen confirming that the Stored Procedure activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the database schema (refer to Figure 3.6).

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

	The comment should be at least 1 character in length.

12. Click **OK** to save the comments. This displays a screen confirming that the Stored Procedure activity has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

16 WORKING WITH PROCESS FLOW

A process flow is a set of activities arranged in a sequence to perform a specific task(s). Combining various activities i.e. Source, Target, Schema or Transformer etc. creates a process flow. The execution of a process flow is controlled by the Process Engine. The Process Engine starts the execution of process flow instances and all the activities present in a process flow. When the process flow is executed, data from the source is converted to the intermediate form and then it is dispatched to the target. The transformer does the conversion of data. In the Adeptia Server two types of transformers are used:

- **Stream2XMLStream/XMLStream2stream:** This transformer converts the source data to XML (i.e. Stream2XMLStream) and then XML to target data (i.e. XMLStream2Stream).
- **SchemaStream2Record/SchemaRecord2Stream:** This transformer converts source data to intermediate format (i.e. SchemaStream2Record) and then intermediate format to target data (i.e. SchemaRecord2Stream).

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√	√	√

Repository

When the process flow is executed, data from the source is converted to the intermediate form and then it is dispatched to the target. The intermediate data is stored in a repository folder. The path of the repository folder is *.../../AdeptiaServer-5.0/ServerKernel/web/repository*. Another folder is created representing the group which executed the process flow. All files that are created during the activity, design and running of the process flow, are stored in the group folder and this group folder is stored inside the repository folder. While creating a process flow, the user can specify whether to keep or delete the repository folder.

Activities of a process flow are executed in two ways:

- **Synchronous Activity:** If an activity is synchronous, process flow initiates the execution of that activity and waits for its completion. The execution of subsequent activity is initiated only after the completion of first activity.
- **Asynchronous Activity:** If an activity is asynchronous, process flow initiates the execution of that activity and subsequent activities as defined in the process flow without waiting for the completion of the first activity.

Repository file is not generated for the activities which are used in **Asynchronous** mode.

Controlling Synchronization

In a process flow, if an activity is synchronous, no signal is required to acknowledge the Process Engine about its completion as it runs in the main thread. If an activity is asynchronous, running parallel to main thread, some mechanism is required to notify the Process Engine about its completion. This is achieved by signaling.

Signaling

Signaling facilitates coordination of asynchronous activities in a process flow. A signal is raised as an asynchronous activity completes. The Synch Node in the process flow waits for signal and ensures that all the activities raising that signal are completed before control moves further in the process flow. This is done using the Synch Node option while defining a process flow.

Time Out

When an activity is used as asynchronous, a new property TimeOut is added to it. By default, its value is 300 seconds. If an asynchronous activity cannot connect to the next activity within this time limit, it gets aborted. To know how to use an activity in asynchronous mode and to changes its property, refer to the section **Creating Process Flow**.

UNDERSTANDING PROCESS DESIGNER

Process Designer allows users to design business processes and business rules. It enables comprehensive designing of a process flow by the simple *Point and Click* method. The Business Process Modeling Notation (BPMN) standard is used to graphically depict business processes. The Process Designer is easy to use and enables both technical and non-technical users to design processes. The Process Flow Designer applet is displayed in Figure 16.1.

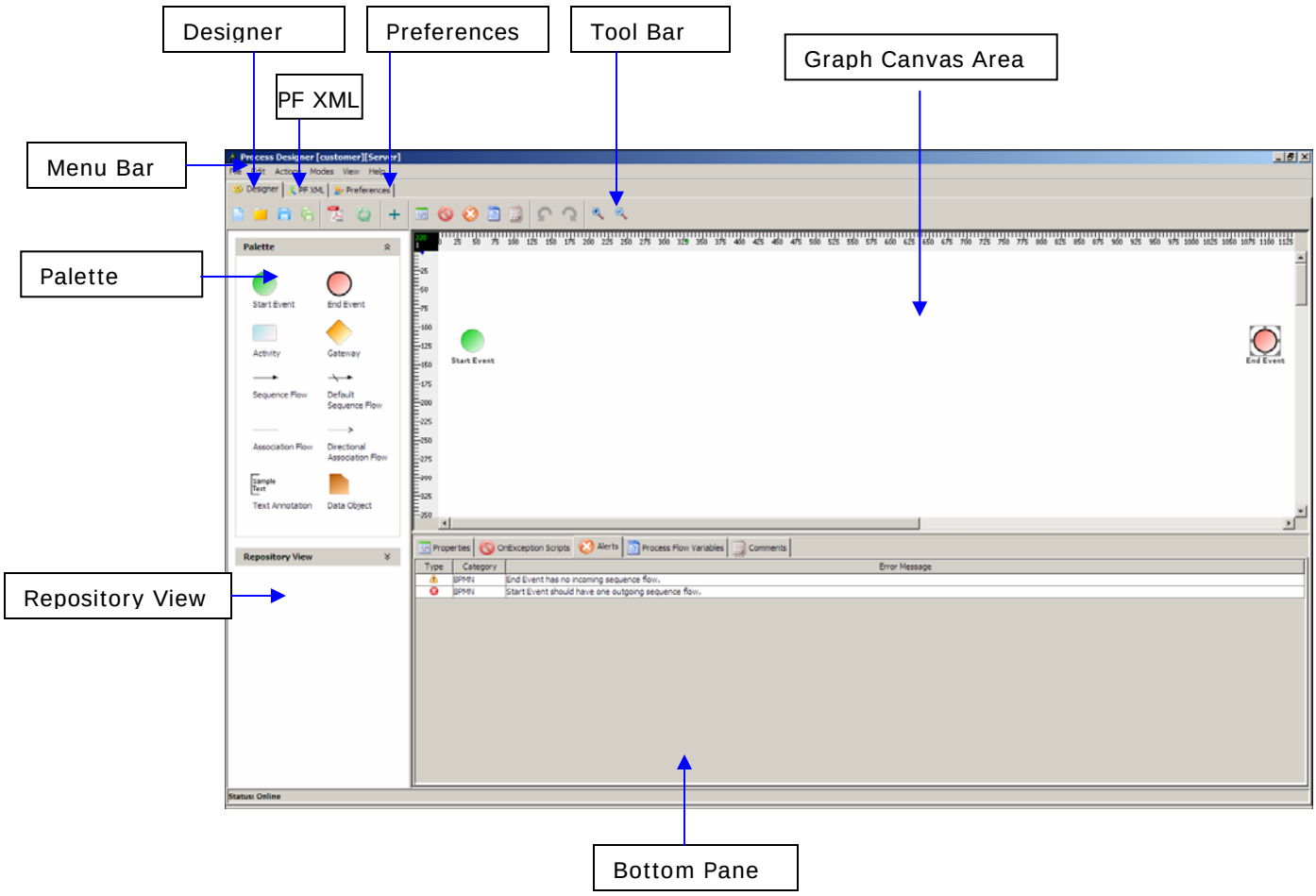


Figure 16.1: Process Designer Applet

The Process Flow Designer window is divided into eight sections:

- **Menu Bar**
- **Tool Bar**
- **Palette**
- **Repository View**
- **Graph Canvas**
- **Bottom Pane**
- **PF XML (Process Flow XML)**
- **Preferences**

Menu Bar

Options of the Menu Bar are listed in the table below.

Table 16.1: Menu Bar Options

Menu Option	Sub-Option	Function
File	New	Open new Graph Canvas to create a new process flow.
	Open locally saved Process Flow	Open a process flow saved to a local machine.
	Save to Server	Save process flow to the Adeptia Server.
	Save Process Flow locally	Save process flow to a local machine.
	Import Process Model	Import a process model and create the process flow based on that model
	Exit	Close the Process Designer applet.
Edit	Undo	Undo the previous action.
	Redo	If you later decide you didn't want to undo an action, click the Redo button
	Zoom In	Zoom and magnify the current selection.
	Zoom Out	Zoom and minimize the current selection.
Actions	Synchronize PD with Server	Synchronize the process flow with a list of Adeptia Server objects such as activities and process flow.
	Enter Login Information	Enter login information for accessing process designer.
	Maximize Graph Canvas	Maximize the Graph Canvas workspace.
	Show Properties Tab	Display and activate the Properties tab at the bottom pane of the Graph Canvas.
	Show Exception Handler Tab	Display and activate the On-Exception Scripts tab at the bottom pane of the Graph Canvas.
	Show Alerts Tab	Display and activate the Alerts tab at the bottom pane of the Graph Canvas.

	Show Process Flow Variables Tab	Display and activate the Process Flow Variables tab at the bottom pane of the Graph Canvas.
	Show Comments Tab	Display and activate the Comments tab at the bottom pane of the Graph Canvas.
	Process Flow Attributes	Allow you to enter the properties of the process flow.
Modes	Online/Offline	Toggle between online and offline mode.
View	Show Flow Object Labels	Display labels of process flow objects.
	Show Connectivity Object Labels	Display labels while connecting objects.
	Show Artifacts and Associations	Display artifacts and associations.
	Show Control Flows	Display control flows.
	Grid	Display grid in Graph Canvas.
Help	Help	Displays help for the Process Designer applet.

The sub-options of the File, Edit, Actions and Help menus can also be accessed by their keyboard shortcuts. To view a list of the defined shortcuts, refer to the **Keyboard Shortcuts** section.

Tool Bar

Options of the Tool bar are explained in the table below.

Table 16.2: Tool Bar Options

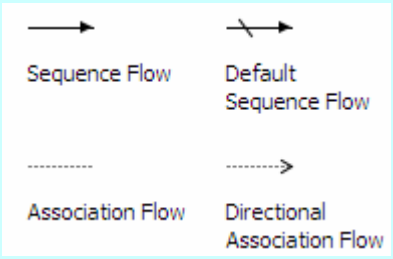
Button	Name	Function
	New Process Flow	Open new graph canvas to create a process flow
	Open Locally Saved Process Flow	Open process flow file saved on local hard disk.
	Save Process Flow to Adeptia Server	Save Process flow to the Adeptia Server.
	Save process Flow Locally	Save process flow on local hard disk.

	Generate PDF	Generate PDF file of the process flow diagram. You can generate a *Graph PDF*, S*ummary PDF* or *Entire Flow PDF*. A *Graph PDF* includes all the rules applied on all activities in the process flow. A *Summary PDF* includes only the activity details, and is implemented only on mapping and schema. The *Entire Flow PDF* includes all details of the process flow.
	Synchronize with Adeptia Server	Synchronize a list of Adeptia Server objects i.e. activities and process flow from the Adeptia Server.
	Maximize/Restore Flow Canvas	Maximize and restore graph canvas.
	Show Properties Panel	Show the Properties panel in the bottom pane.
	Show Exception Handler Panel	Show the Exception Handler Script in the bottom pane.
	Show Error Panel	Show the Error panel in the bottom pane.
	Show Process Flow Variable Panel	Show the Process Flow Variable Panel in the bottom pane
	Show Comments Panel	Show the Comments panel in the bottom pane.
	Undo	Undo the last action.
	Redo	If you later decide you didn't want to undo an action, click the Redo button.
	Zoom In	Zoom In the Graph Canvas area.
	Zoom Out	Zoom Out the Graph Canvas area.

Palette

The Palette contains a list of BPMN graphical elements. BPMN graphical elements are used to define the flow of business processes. These BPMN graphical elements are listed in five different panels, listed in the table below.

Table 16.3: BPMN Graphical Elements

Element	Description	Notation
Events	An event is something that "happens" during the course of a business process. Events influence the flow of the process and usually have a cause (trigger) or an impact (result).	
Activities	An activity is work that is performed within a business process. The rectangle image displayed in the next column is used to depict an Activity in a Process Flow.	
Gateways	A Gateway is used to control the divergence and convergence of a sequence flow. Thus it determines branching, forking, merging, and joining of paths.	
Flows	A flow (control flow) is used to show the order that activities are performed in a business process. There are four types of flows: ▪ Sequence flow ▪ Default Sequence flow ▪ Association flow ▪ Directional Association	
Artifacts	Artifacts do not have any direct effect on the sequence flow or message flow of the process. They are used to provide additional information for the reader of the Process flow diagram. You can add any amount of information in this element. However, you cannot add any color to the text. To view the information, you can resize the frame in all directions.	 Data Object

	All the BPMN Graphical Elements can be resized.

To know more details about BPMN (Business Process Modeling Notations) visit the site:
http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf

Repository View

The Repository View lists the Adeptia Server objects, such as Activities, Process flows, Context variables and Actions.

	The Activities Panel has been renamed to Repository View. It is not reflected in the screenshots of this section.

Graph Canvas

The Graph Canvas is the area where a process flow is drawn. BPMN specification and Adeptia Server activities are dragged onto the Graph Canvas and arranged in a sequence in order to create a process flow. The Graph Canvas can also be resized if an activity is dragged beyond the default size.

Bottom Pane

The Bottom Pane is used to view properties of the activities, add comments to the activities, view error in a process flow and to create context variables. There are five panels in the Bottom Pane:

- Properties
- On Exception Script
- Errors
- Process Flow Variable
- Comments

PF XML (Process Flow XML)

The Process Flow window displays Process Flow XML that is generated while designing a process flow. To view the XML of a process flow click **PF XML** tab (see Figure 16.2).

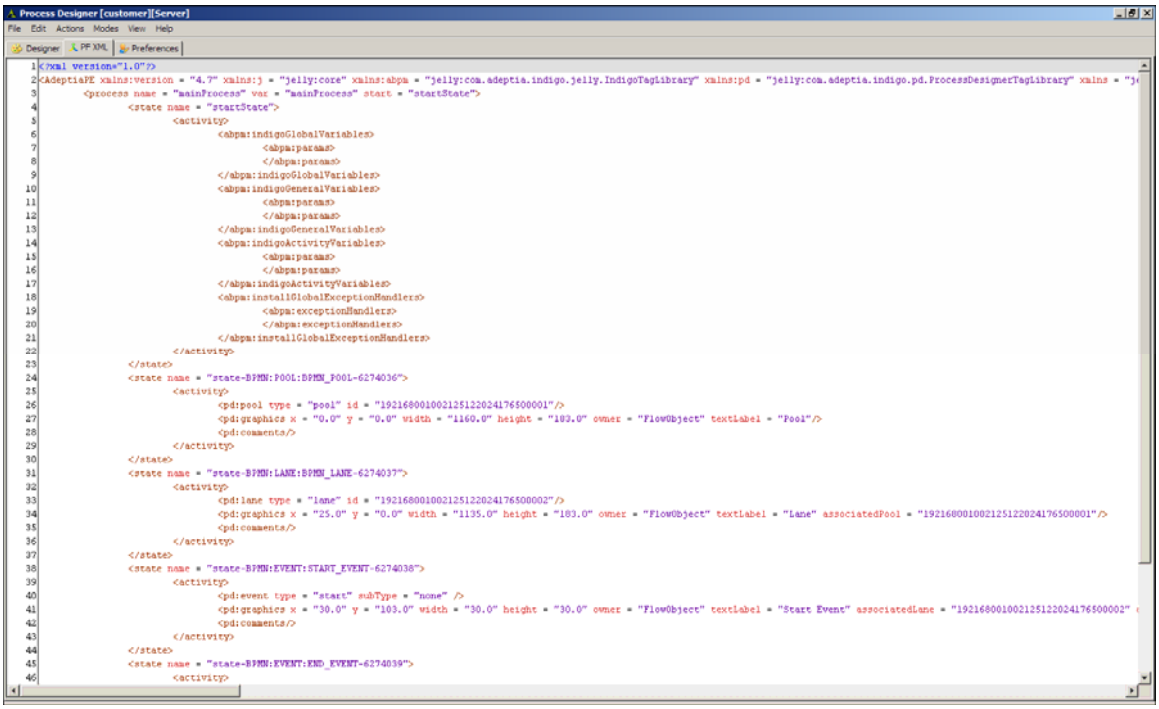


Figure 16.2: View Process Flow XML

Preferences

The Preferences window allows you to select the background color of graph canvas and labels etc. To view the Preferences window, click **Preferences** tab (see Figure 16.3).

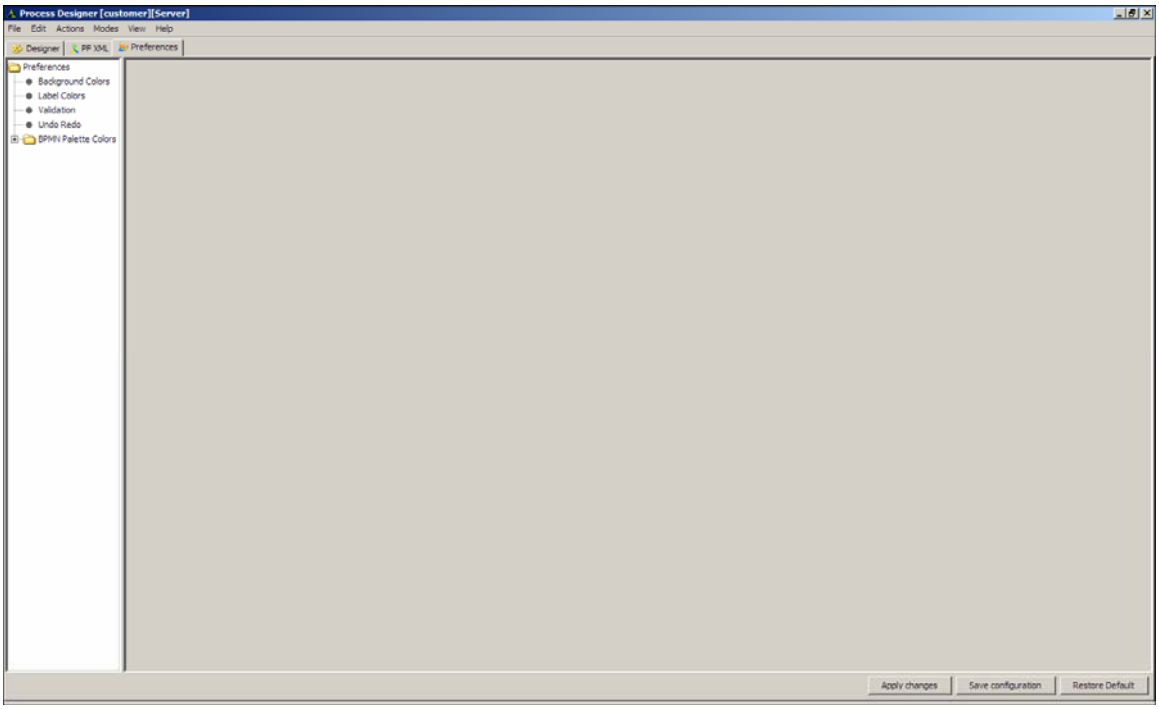


Figure 16.3: Preferences

You can modify various preferences. These are listed in the table below.

Table 16.4: Edit Preferences

Preferences	Description
Background Colors	Change the background color of the Graph Canvas, BPMN Events Panel or Activities Panel.
Label Colors	Change the colors of labels of activities displayed in the Graph Canvas.
Validation	Verifies that the process flow created in the Graph Canvas is correct as per the BPMN standard and Adeptia Server. You can enable/disable Validation in the Preferences window. By default, it is enabled.
Undo Redo	Reverts the action done by the user in the Graph Canvas. You can enable/disable Validation in the Preferences

	window. By default, it is disabled.
Set Undo and Redo Limit	Set the number of actions that you can undo or redo.
BPMN Palette Colors	Change colors of Events, Activities, Gateway and Artifacts.

Keyboard Shortcuts

Table 16.5: Keyboard Shortcuts

Menu Option	Sub-Option	Keyboard Shortcut
File	New	<Ctrl> + <N>
	Open locally saved Process Flow	<Ctrl> + <O>
	Save to Server	<Ctrl> + <S>
	Save Process Flow locally	<Ctrl> + <B>
Edit	Undo	<Ctrl> + <Z>
	Redo	<Ctrl> + <Y>
	Zoom In	<Ctrl> + <NumPad +>
	Zoom Out	<Ctrl> + <NumPad ->
Actions	Synchronize PD with Server	<F5>
	Enter Login Information	<Ctrl> + <L>
	Maximize Graph Canvas	<Ctrl> + <M>
	Show Properties Tab	<Ctrl> + <R>
	Show Exception Handler Tab	<Ctrl> + <G>
	Show Alerts Tab	<Ctrl> + <E>
	Show Process Flow Variables Tab	<Ctrl> + <F>
	Show Comments Tab	<Ctrl> + <H>
Help	Help	<F1>

CREATING PROCESS FLOW

Prerequisites:

- JRE 1.5 needs to be installed on your system to open the Process Designer applet.
- The *Pop-up Blocker* needs to be disabled in the web browser, to open the Process Designer applet. By default, the *Pop-up Blocker* is enabled.
- The windows user must have Administratoristrative rights on the PC, where Process Designer will be opened.

The Process Flow Creation using Process Designer comprises of two parts:

- **Designing Process Flow using BPMN Graphical Elements**
- **Attaching Adeptia Server activities with the BPMN elements**

Designing Process Flow using BPMN Graphical Elements

Steps to draw a Process Flow using Process Designer

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. A The Manage Process Flow screen is displayed (see Figure 16.4).

Figure 16.4: Manage Process Flow

3. Click **New** link. The Create Process Flow screen is displayed (see Figure 16.5).

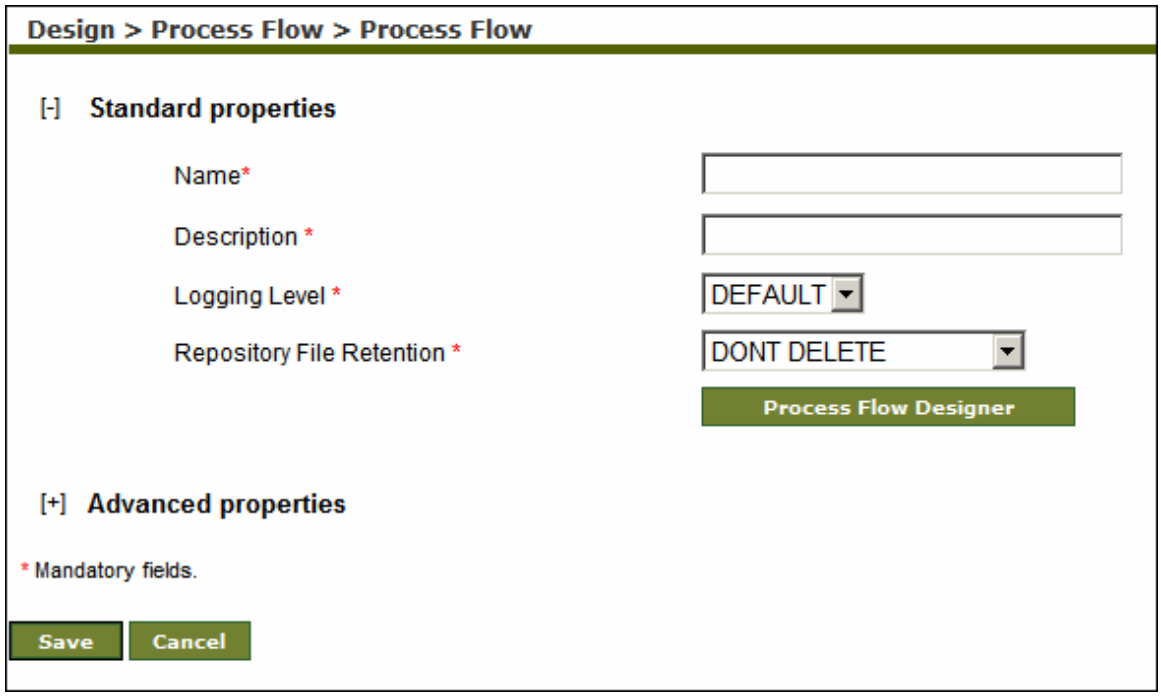


Figure 16.5: Create Process Flow

4. Enter the name and description of the new process flow in the *Name* and *Description* fields respectively.

	Description of the process flow can be overridden during execution of the process flow. To know how to override the process flow description, refer to the section **Overriding Process Flow Description at Runtime**.

5. Select the logging level from the *Logging Level* drop-down list. The various logging levels are depicted in the table below.

Table 16.6: Types of Logging Levels

Level	Description
DEBUG	The DEBUG level logs are fine-grained informational events that are most useful to debug any problem. Debug level is useful for programmers.
INFO	The INFO level logs are informational messages that highlight the progress of Process flow execution. In INFO, status (successful or failure) of each activity is displayed.
ERROR	In ERROR, possible cause of failure of an activity is displayed.
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 Server. You can also choose an option to delete them or to archive them in a different location. Options for Repository File Retention are outlined in the table below.

Table 16.7: Options for Repository File Retention

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. To change the location where archived file is stored, refer to the section to any other folder or to WebDAV repository, you need to change *abpm.transaction.repository.archive.server* property. Refer to the Administratoristrator Guide for details.
DELETE ON SUCCESS	Repository files are deleted only when the process flow is executed successfully and there is no error record.

7. Click **Process Flow Designer** button. The Process Designer window is displayed (see Figure 16.6).

If you are starting the Process Designer on your system for the first time, then a warning message is displayed that prevents you from starting this application. Just ignore this message and click **Start** to continue.

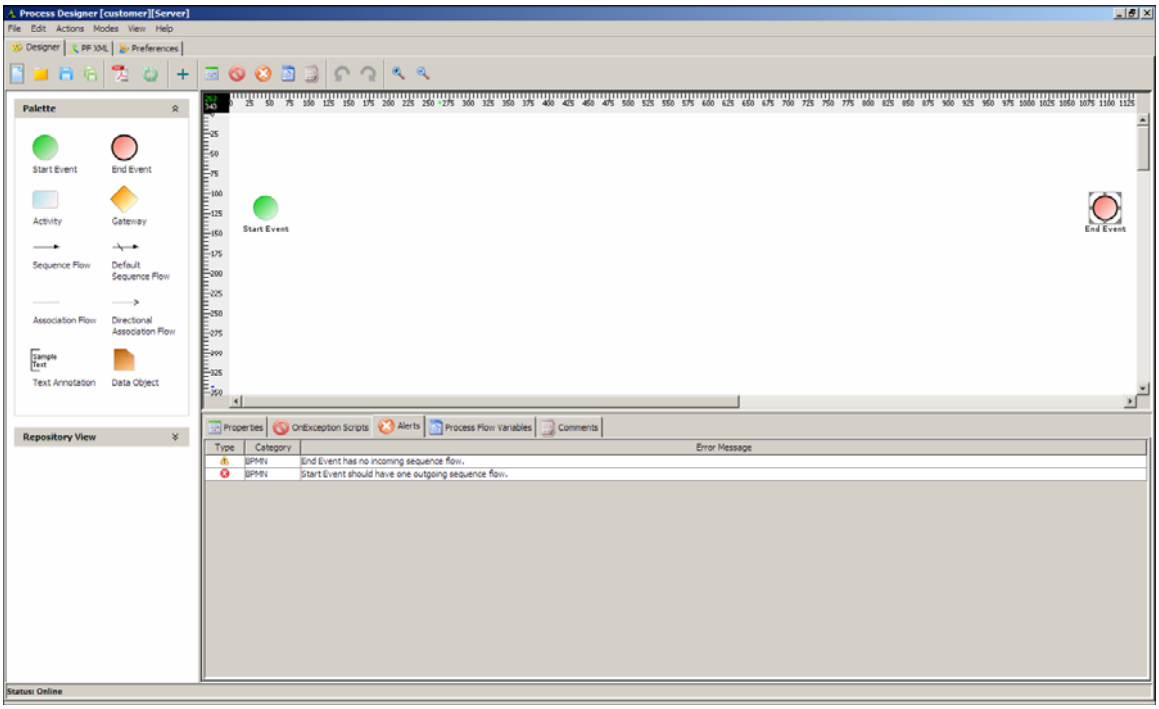


Figure 16.6: Process Designer

8. Click **Activity** element in the Palette and drag it to the Graph Canvas. The dragged activity element is displayed in the Graph Canvas (see Figure 16.7).

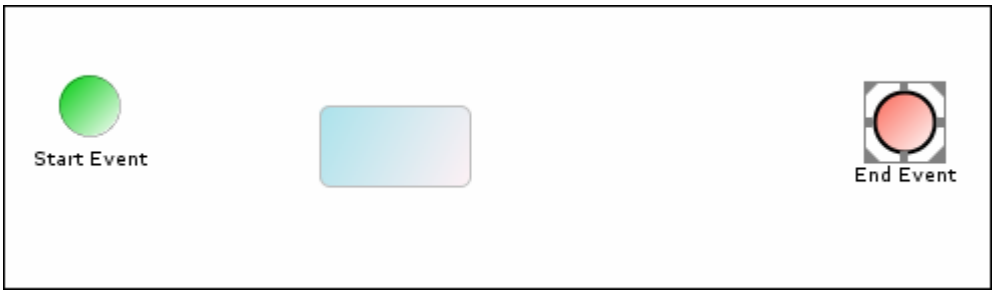


Figure 16.7: Dragging BPMN Activity into Graph Canvas

9. By default, all BPMN elements except Event objects are blank. You can add a label to the BPMN element, once you drag it to the Graph Canvas. You can do this by double-clicking the element. This displays the properties associated with the element in the Properties Panel in the Bottom Pane. Alternately, you can right-click the element and select **View Properties** option (see Figure 16.8).

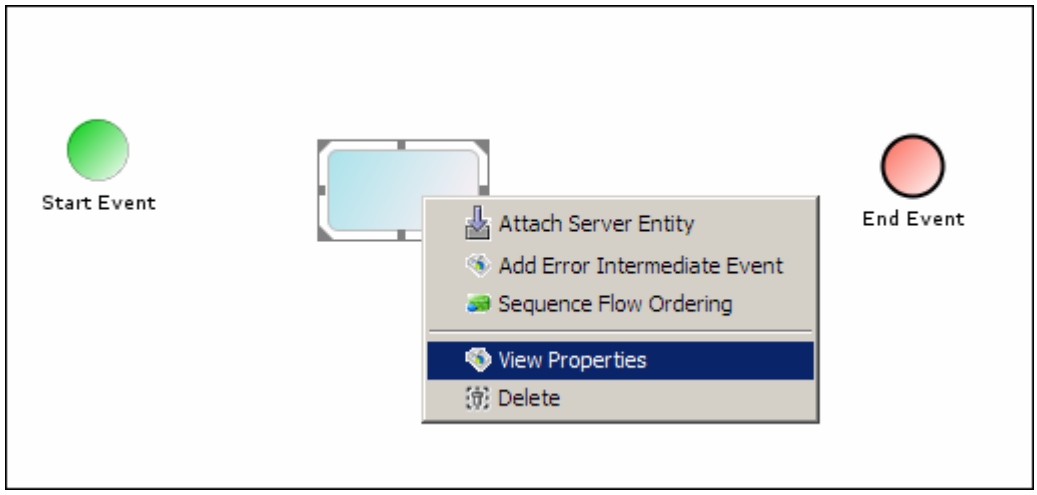


Figure 16.8: Right-Clicking an Activity

10. The Properties Panel is displayed in the Bottom Pane. Type the name of the element in the *Value* column of the *Label* field. Click the element again in the Graph Canvas to display the name in the element (see Figure 16.9).

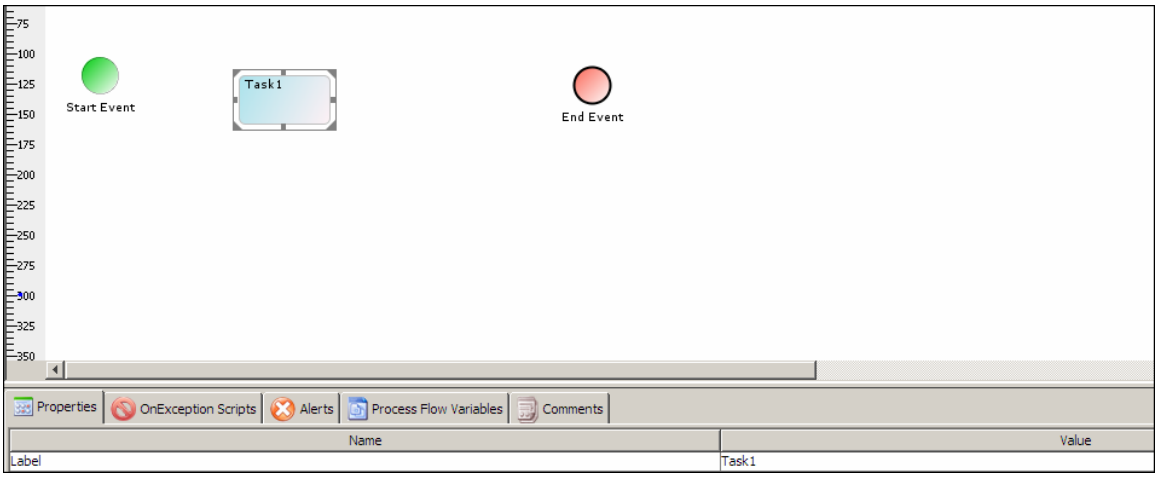


Figure 16.9: Displaying Name of Activity

	All BPMN elements can be labeled in the same way.

11. Repeat steps 9 and 10 to add more activities in Graph Canvas.
12. Click **Gateway** element and drag it to desired location in the Graph Canvas. The Gateway element is displayed in the Graph Canvas area (see Figure 16.10).

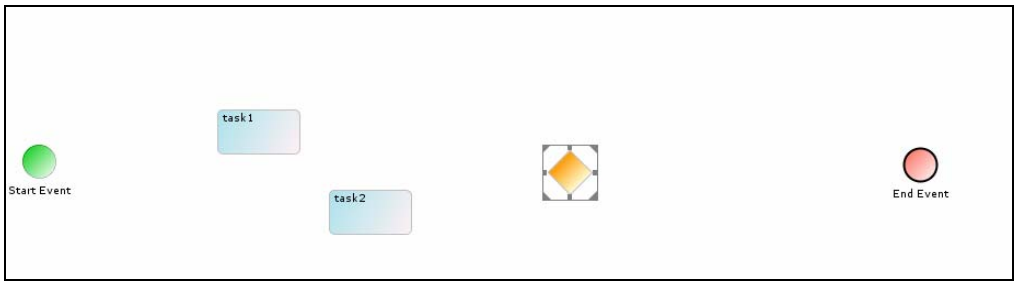


Figure 16.10: Dragging BPMN Gateway into Graph Canvas

	All similar elements can be dragged and dropped in the same way.

Once the required BPMN elements are dragged to the Graph Canvas and proper labeling is done using the Properties Panel, it is necessary to connect them in order to design a business flow.

13. Click required flow in the Palette. The control flow is selected.
14. Drag the cursor between two BPMN elements. Both BPMN elements are connected with the selected flows (see Figure 16.11).

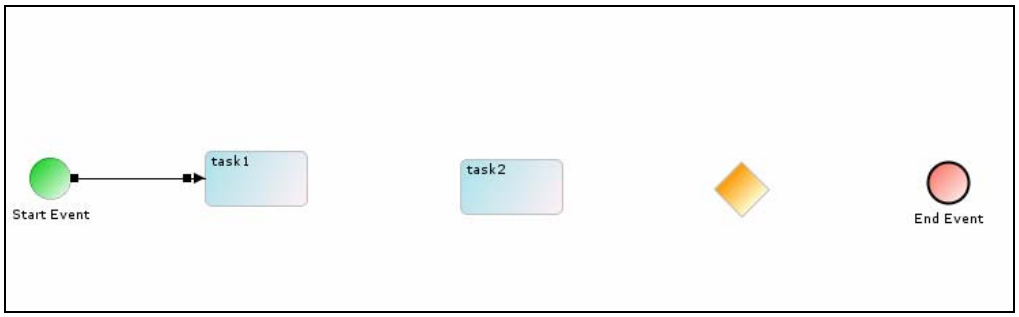


Figure 16.11: Connecting BPMN Elements

15. Connect all BPMN elements with appropriate control flow (see Figure 16.12).

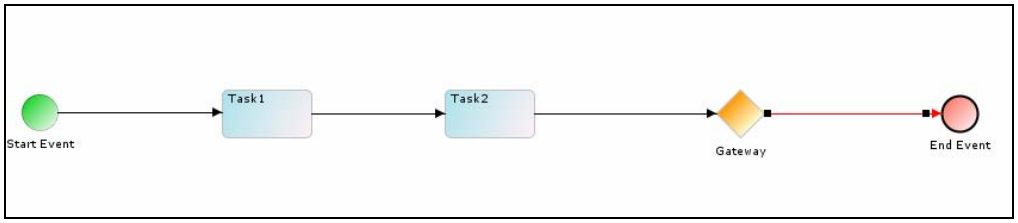


Figure 16.12: Connecting BPMN Elements

	You can reposition the BPMN elements by moving the arrow keys. All elements except Sequence Flow can be repositioned.

Once designing of business process is completed it is necessary to attach the Adeptia Server activities to BPMN elements of the business process.

Attaching Adeptia Server activities with the BPMN elements

Prerequisites

- Adeptia Server activities must be created before attaching them with BPMN elements in the Process flow.

Steps to attach Adeptia Server activities with BPMN elements

1. Expand the **Repository View** panel. All the items in the Server Entities category are displayed (see Figure 16.13).

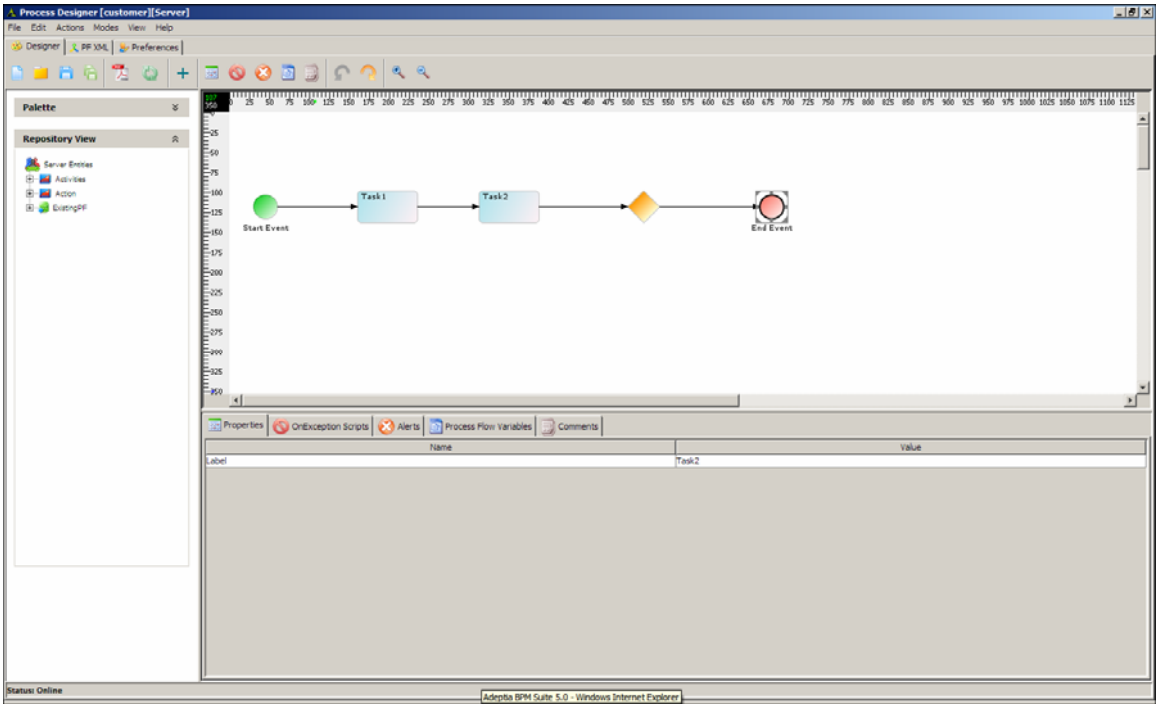


Figure 16.13: Selecting Adeptia Server Activity

2. Further expand the type of activities e.g. Human Workflow, Schema, Source, etc until you find the required activity (see Figure 16.14).

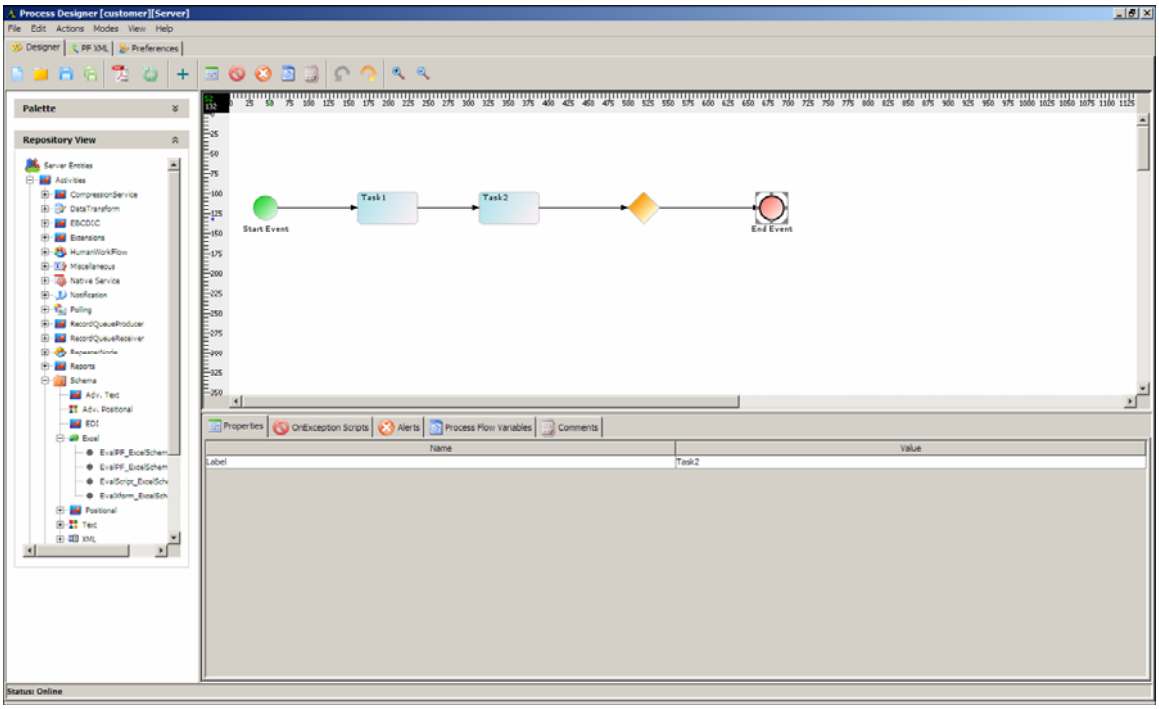


Figure 16.14: Selecting Adeptia Server Activity

3. Select the required activity, drag it to the Graph Canvas and drop it on the BPMN element with which you want to attach the activity. A small image of the activity is displayed at the top left corner of the BPMN element (see Figure 16.15).

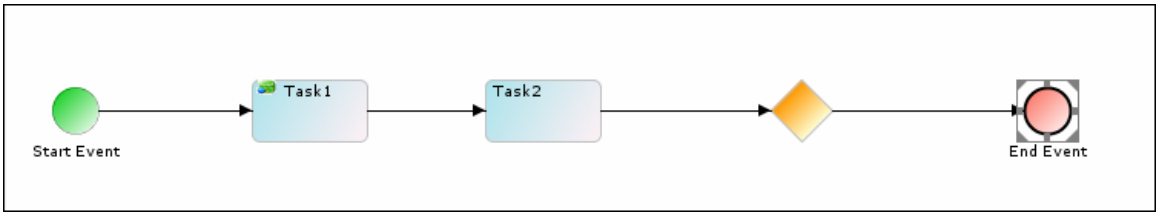


Figure 16.15: Attaching Adeptia Server activity

	An activity can also be attached by right-clicking the BPMN element. Right-click BPMN element and select **Attach Server Entity**. List of activities is displayed. Select the required activity and click **Ok** button. You can change the label of the element if desired. If you attach an activity to a blank BPMN element, then the activity name is displayed in the BPMN element (see Figure 16.16).

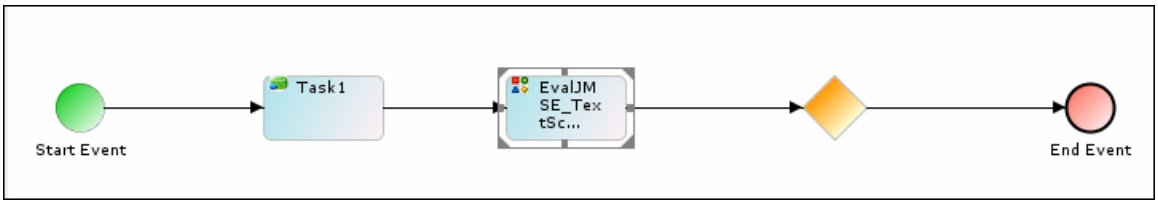


Figure 16.16: Adeptia Server activity name in BPMN element

4. Repeat step 1 to 3 to attach activities to other BPMN elements (see Figure 16.17).

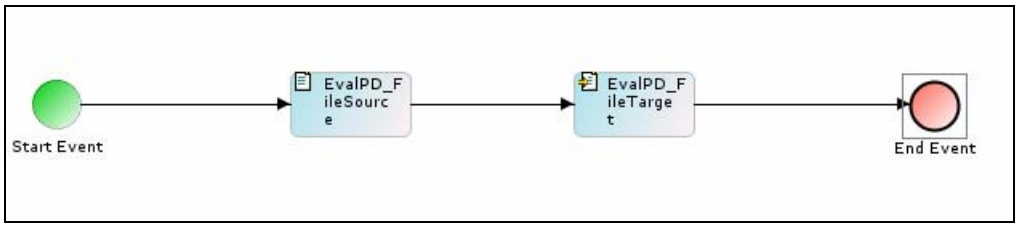


Figure 16.17: Attaching Adeptia Server activity

5. To view or edit the properties of the Adeptia Server activity attached with the BPMN elements, right-click activity and then select **View Properties**. The properties of the attached activity are displayed in the Properties Panel (see Figure 16.18).

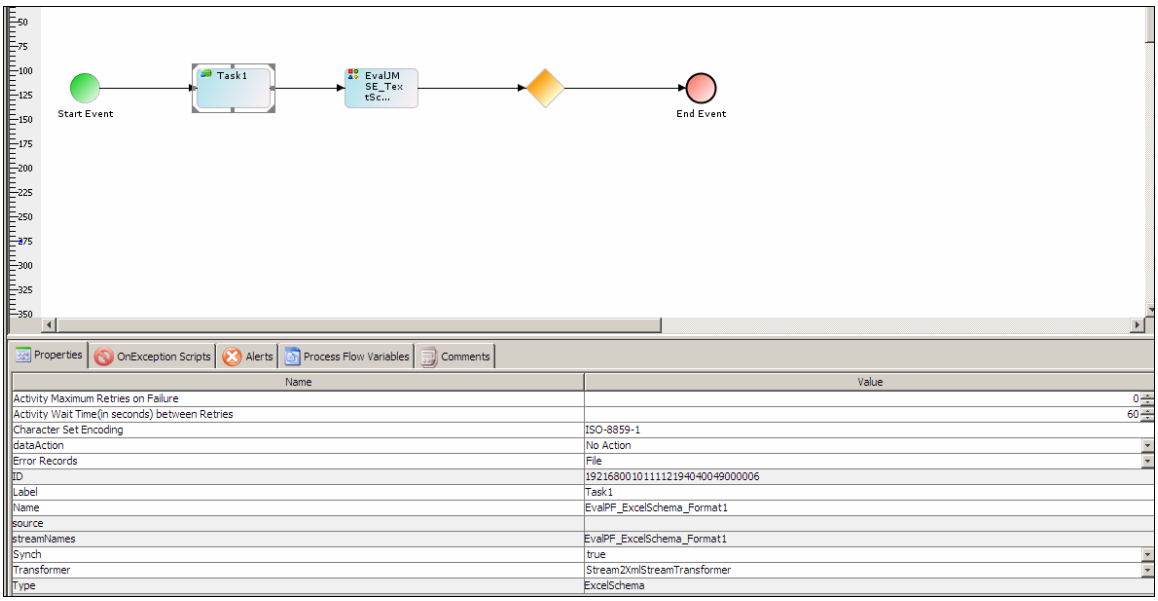


Figure 16.18: Edit Activity Properties

If mapping is used in a process flow, following things must be checked:

- Right-click source schema activity and select View Properties. *Transformer* property must be *Stream 2XMLStream*.

	▪ Right-click target schema activity and select View Properties. *Transformer* property must be *XMLStream2Stream*. ▪ For details on Transformer types and changing from one type to another, refer to the section Changing Transformer Type.

6. By default activities are *Synchronous*. If you want to make the activity *asynchronous*, go to the **Properties Panel**, change the value of *Synch* from *true* to *false*.
7. Click **File** menu and then select **Save to Adeptia Server**, to save the process flow on the Adeptia Server. This displays a screen confirming that the process flow has been created successfully. If the comments property is enabled, then clicking **Save to Adeptia Server** will display a screen where you need to enter comments related to creating the process flow (see Figure 16.19).

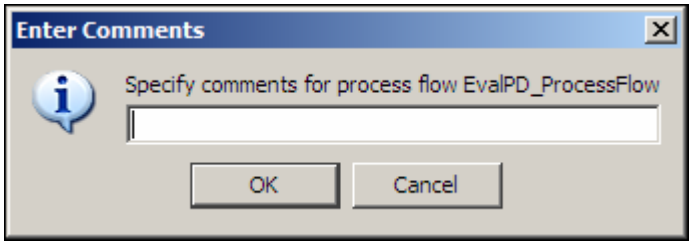


Figure 16.19: Enter Comments (Process Flow)

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

	▪ The comment should be at least 1 character in length. ▪ If you enable/disable the *Comments* property in the middle of a process flow, you need to restart the process flow applet.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator Guide*.

Save Process Flow on Local Hard Disk

Steps to save the Process Flow on local hard disk

1. Select **Save Process Flow Locally** from the **File** menu. The Save window is displayed (see Figure 16.20).

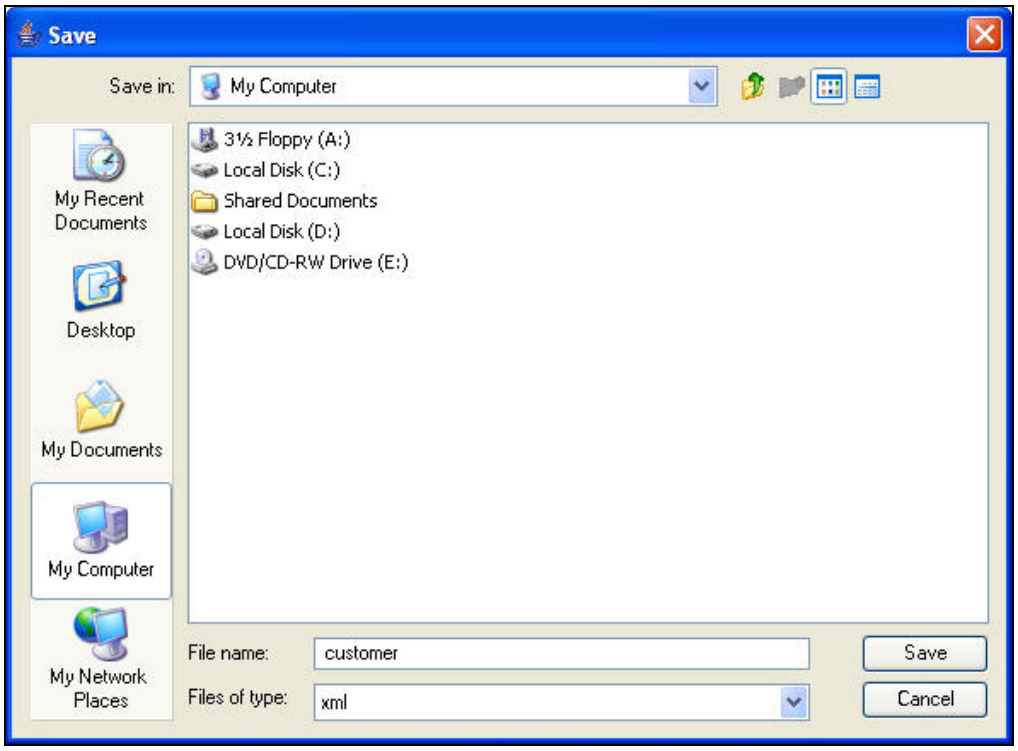


Figure 16.20: Save Process Flow

2. Enter the name of the file in *File Name* field and click **Save** button. The process flow is saved in XML format in the specified location.

	To create a Process flow, an IT user can simply drag the required Adeptia Server activities to the Graph Canvas and connect them using flow controls. In other words, an IT user does not need to draw process flow using BPMN elements and then attach Adeptia Server activities to the BPMN elements.
	You can view details of an event associated with a process flow, by clicking the event displayed under Associated Events on the Manage Process Flow screen.

	If a process flow is opened in **Read-Only** mode, you can view and modify it, but you cannot save the changes, as all Save options are disabled. However, if you open a process flow that has write permissions, from the Process Designer applet, then Save options become activated.

Changing Transformer Type

In the Adeptia Server two types of transformers are used:

- **Stream2XMLStream/XMLStream2stream**: This transformer converts the source data to XML (i.e. Stream2XMLStream) and then the XML to target data (i.e. XMLStream2Stream). This transformer is used when a mapping activity is used in the process flow. In this case, the Stream2XMLStream is used in the source schema and the XMLStream2stream is used in the target schema.

- **SchemaStream2Record/SchemaRecord2Stream**: This transformer converts the source data to intermediate format (i.e. SchemaStream2Record) and then the intermediate format to target data (i.e. SchemaRecord2Stream). This transformer is used when record to record process of the data is required.

Steps to change the transformer type

1. Right-click the schema activity in the Graph Canvas Area, and select **View Properties**. Properties of the selected schema activity are shown in the Properties Panel (see Figure 16.21).

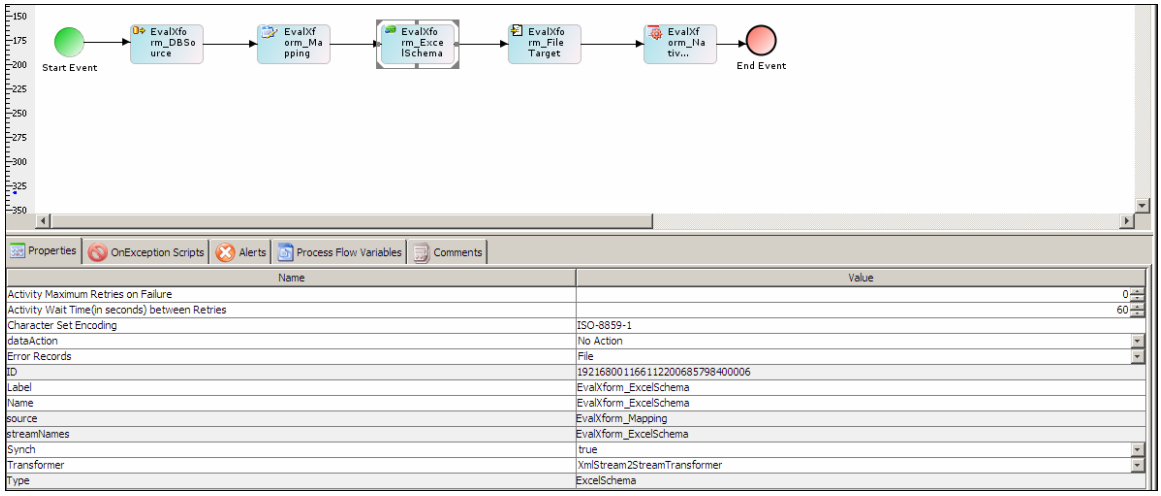


Figure 16.21: Changing Transformer Type

2. Select the required transformer type (e.g. *Stream2XMLStream* in case of source schema and *XMLStream2Stream* in case of target schema) from the *Transformer* drop-down list.

ACTIVATING/DE-ACTIVATING PROCESS FLOW

Process flows can be in activated or de-activated state. You cannot execute a process flow, which is in de-activated state. If you want to execute a de-activated process flow, you have to first activate it. When a process flow is created, it is in activated state, if number of activated process flows is less than maximum allowed by license. Otherwise process flow is created in de-activated state.

Steps to activate/de-activate a process flow

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. A The Manage Process Flow screen is displayed.
3. If the process flow is in activated state, a **De-activate** link is highlighted when select the radio button adjacent to it.
4. To de-activate the process flow, click the **De-activate** link.
5. Similarly to activate a de-activated process flow, select the radio button adjacent to it and click **Activate** link.

TESTING A PROCESS FLOW

Process flow can be tested, before executing it. By testing a process flow you will be able to know the behavior of the process flow, right before executing it in production environment. Testing is useful especially for those process flows, in which decision nodes are used. When you test a process flow, you can give values of different process flow variables used in the process flow and check its behavior. The values of process flow variables are given using a XML file.

Steps to test a process flow

1. In the Manage Process Flow screen, click the radio button against the process flow that you want to edit. This selects the process flow and activates the *Edit* link. Clicking the **Edit** link displays the Edit Process Flow screen (see Figure 16.22).

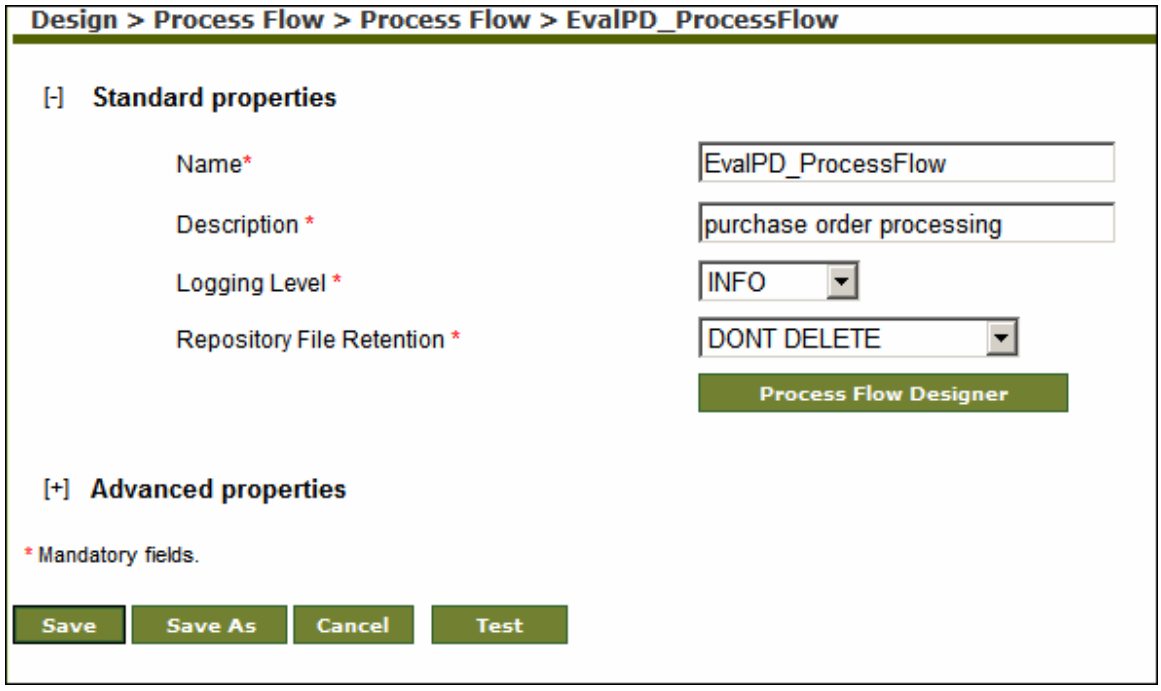


Figure 16.22: Edit Process Flow

2. Click the **Test** button. The Test Process Flow screen is displayed. (see Figure 16.23).

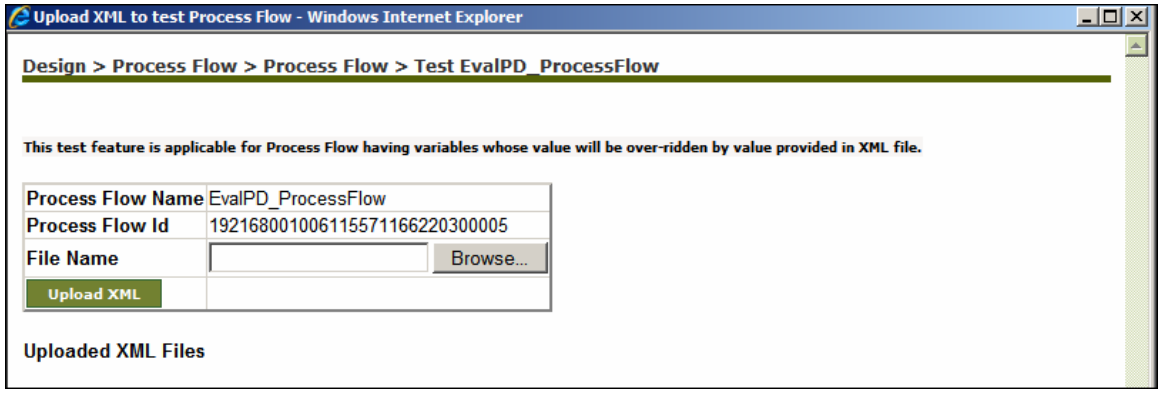


Figure 16.23: Test Process Flow

3. Click **Browse** and select the required XML files. Following is the sample of XML file used to test the process flow. (see Figure 16.24).

```
<?xml version="1.0"?>
<Variables>
  <Variable name="Data" value="55000"/>
</Variables>
```

Figure 16.24: Sample XML

4. Once the required file is selected, click **Upload XML** button. The uploaded file is show in the Test Process Flow screen. (see Figure 16.25).

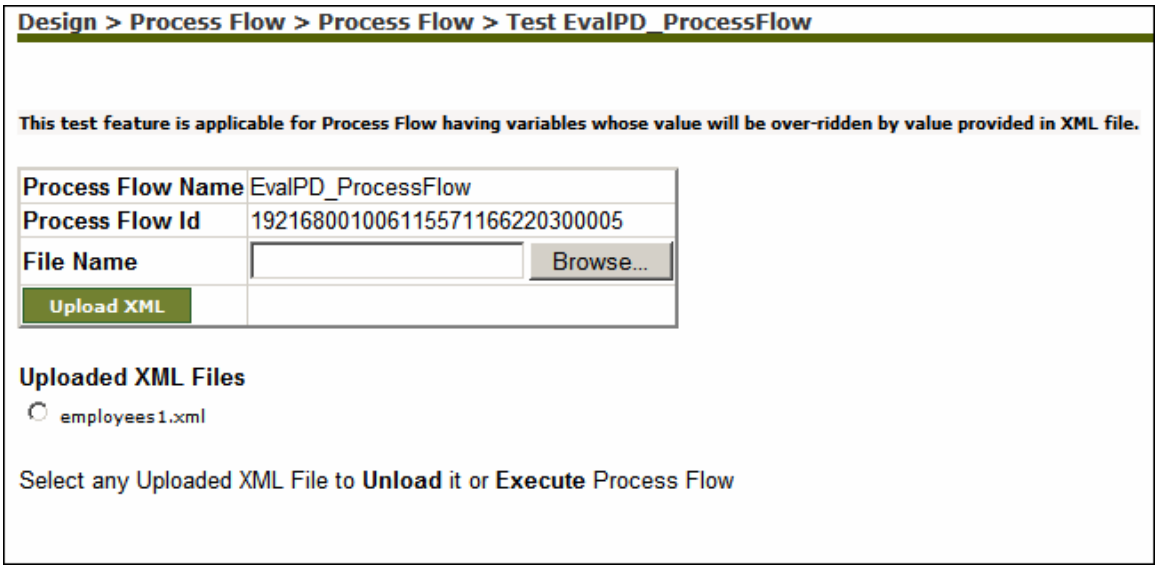


Figure 16.25: Select XML File

	You can upload more than one file.

5. Select the uploaded XML file and click the **Execute** button (the *Execute* link changes to a button when a file is selected). The following screen is displayed. (see Figure 16.26).

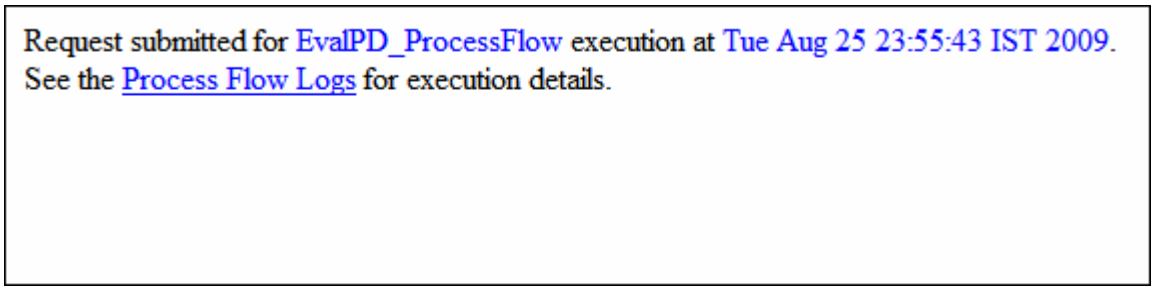


Figure 16.26: View Process Flow Log

6. Click **Process Flow Logs** link, to check the status of the process flow.

UNDERSTANDING VALIDATION

Validation is a mechanism, which ensures that the process flow created in Graph Canvas is correct as per the BPMN standard and Adeptia Server. If the process flow is not correct, a message is displayed in the Alerts Panel of Bottom Pane (see Figure 16.27).

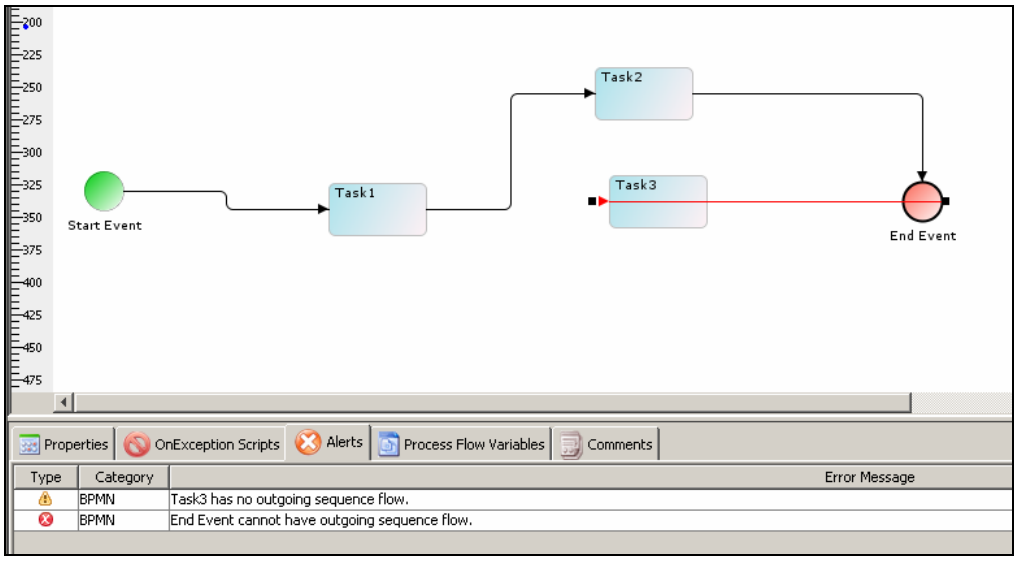


Figure 16.27: Validation

In Figure 16.27 shown above, you can see that *Task1* and *Task 2* are properly connected with the incoming and outgoing sequence flow but *Task3* has an incoming sequence but no outgoing sequence flow. The validation message is displayed in the Alerts Panel. There are two categories of validations:

- **BPMN Server specific Validation:** This includes Adeptia Server validation like file target has no input source stream, etc. This is displayed with "BPMServer" category in the Alerts Panel.
- **BPMN Specific Validation:** This includes BPMN specific validation like end event cannot have outgoing sequence flow, etc. This is displayed with "BPMN" category in the Alerts Panel.

Validation messages are further divided into two types:

- **Error:** These are displayed in the Alerts Panel with type ( ) along with error message and the corresponding user action is reverted if possible. If revert action takes place successfully then a warning message is displayed explaining the reason of revert operation. For example, start event cannot have incoming sequence flow, hence removed.
- **Warning:** These are displayed the Errors Panel with type ( ) and the corresponding warning message is displayed. For example, start event has no outgoing sequence flow.

By default, *Validation* is enabled and *Revert Action* is disabled. Revert action specifies reverting (if possible) wrong user action automatically by Process Designer.

Disabling Validation

Steps to disable Validation

1. Click **Preferences** tab in Process Designer screen. The Change Preferences screen is displayed (see Figure 16.28).

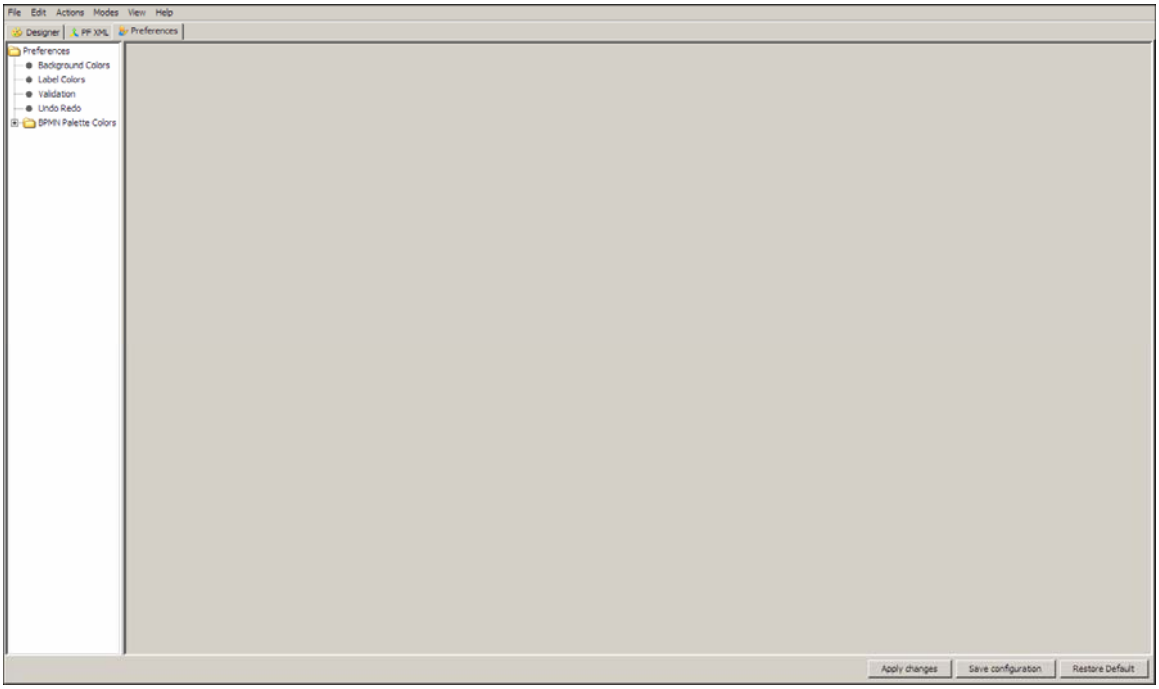


Figure 16.28: Changing Preferences

2. Click **Validation** under the **Preferences** Menu. The Validation Options screen is displayed (see Figure 16.29).

Figure 16.29: Disable Validation

3. Select **Disable** from *Enable or Disable Validation* drop-down list.
4. Click **Apply Changes** button and then click **Save Configuration** button.

	Similar steps need to be done to change the status of Revert Action.

USING UNDO REDO

This feature allow user to perform UNDO and REDO operations. This can be done either through the **Undo** and **Redo** submenu under **Edit** menu or by clicking the **Undo** ( ) and **Redo** ( ) buttons in the toolbar.

- **Undo:** This action will replace the user current action with the previous action. For example, moving the BPMN Element to its previous position.
- **Redo:** This action will replace the user recent action with his undone action. For example, moving the BPMN Element to its previous position where undo action took place.

USING MULTIPLE SELECTIONS

User can select multiple BPMN Elements from the Flow Canvas and move them to other location in the Flow Canvas. Multiple BPMN Elements will be selected with the combined event of left mouse click and **<CTRL>** key or drawing selection rectangle on flow canvas. A selection rectangle is a virtual rectangle drawn as the user press right mouse button and drag over the flow canvas. On release of mouse button the rectangle becomes invisible and the entities inside the drawn rectangle will be selected. An example of multiple selections is displayed in Figure 16.30.

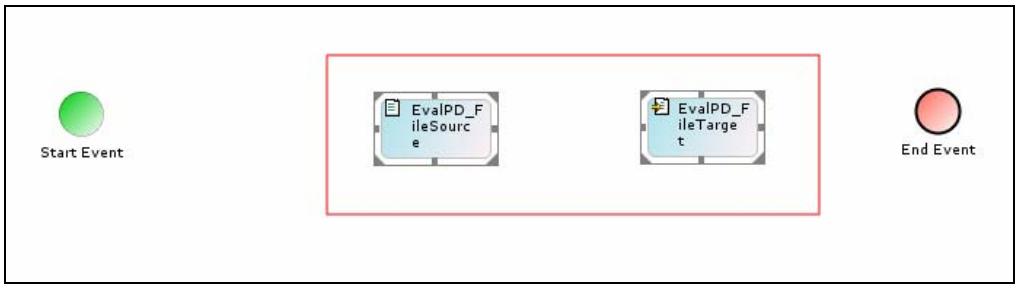


Figure 16.30: Multiple Selections

	Multiple activities can be deleted by selecting multiple activities on the canvas and selecting delete from the right clicked popup menu. Alternately, you can select the activities and press **<Delete>** on the keyboard.

USING ZOOM IN AND ZOOM OUT

This feature allow user to perform ZOOM IN and OUT operations on the canvas. This can be done either through the **Zoom In** and **Zoom out** submenu under Edit menu or by clicking the **Zoom In** ( ) and **Zoom out** ( ) buttons in the toolbar.

In Zoom In/Zoom Out mode, all activities in the Graph Canvas will be resized. Additionally, all connecting sequence flows will be also redrawn. A process flow can be zoomed in to any limit. However when zooming out, it has to conform to certain limits. It will stop in the listed cases:

- Any element location becomes (0,0) coordinates
- Any element dimension becomes less than its minimum dimension

You can drag and move existing activities or add new activities in zoom mode, but when saving the process flow to server, they are resized and relocated to the default proportions. If any error event is attached to an activity, it will also be zoomed in the same proportion. The zoomed size is never saved to the server. When the process flow is reloaded, it will display in normal mode.

	When a process flow is zoomed, the size of images inside a BPMN element will not be zoomed. Additionally, the size of an arrow head will also not change.

USING ACTIONS IN PROCESS FLOW

Actions enable necessary control over a process flow. Lists of actions that can be used to control a process flow are displayed in the table below.

Table 16.8: Actions for a Process Flow

Action	Description
Call:	This action is used to call another process flow (sub-flow) synchronously. Users need to specify the following properties while adding the Call action to a process flow:
	flowId: Name of the process flow (sub-flow). You can override its value dynamically during execution of the process flow. To override the flowId you can use custom plugin activity or put-context-var before call action. Following is the code which is used in custom plugin activity to override the flow id: context.setActivityParameter("Call1","flowId", "192168001158117188341381200001"); where *Call1* is the name of the call activity and *192168001158117188341381200001* is ID of the child process flow. *Label:* The label displayed for the call action. *Name:* Name for the call action *resultCtxVarName:* Name of the context variable that contains the status of the sub-flow. Its value is Boolean.
	Using *Call* action you can select Parent Process flow itself to be executed as child process flow. In this case the process flow will run in infinite loop. You should use some condition, which is use to stop the process flow after certain recursion. If the condition fails, you can kill the process flow from System Monitoring.
Checkpoint:	This action is used to resume the execution of a process flow from its current status if kernel stops unexpectedly. The Checkpoint action saves process flow state (context variable, checkpoint info, state name etc) in a file. When the Adeptia Server kernel is restarted it checks for the recoverable process flow by scanning all the existing process flow files and starts the particular process flow from its last checkpoint. Checkpoint should not be used within JTA block. It should be used before or

	after the JTA block.
	Label: The label displayed for the Checkpoint action. *Name:* Name for the Checkpoint action.
Delay:	This action is used to pause the propagation of process flow for a given time. The Delay action pauses the propagation of process flow but the asynchronous activities, which were started earlier, keep running in parallel. Users need to specify the following properties while adding the Delay action to a process flow:
	Label: The label displayed for the Delay action. *Name:* Name for the Delay action. *Time:* Specify the time in seconds till which the execution of process flow is paused.
JTA-Begin/ JTA-End:	These actions are used to create a JTA block. JTA block is used to create a set of activities within a process flow. When all the activities in a JTA block are completed successfully, the data is committed, and the process flow control can move beyond the JTA block to the next activity. If any of the activities in the JTA block fails to complete successfully, the rollback function is called and the whole process flow is stopped and error is logged. JTA is applicable only when source or target is a database.
	Label: The label displayed for the JTA-Begin/End action. *Name:* Name for the JTA-Begin/End action.
JTA-RollBack:	This action is used to call the rollback function at any point in a process flow. The JTA–RollBack function is always used within a JTA block.
	Label: The label displayed for the JTA-Rollback action. *Name:* Name for the JTA-Rollback action.
Put-Context-Var:	This action is used to declare one or more context variables with values assigned to it at any point in the process flow. A context variable is declared when this action is executed while running the process flow. Put-Context-Var is generally used to set the value of any field of an activity used in a process flow, during execution of the process flow. For example you can set/overwrite the subject of mail source activity during execution of process flow. Another example can be appending current date stamp at the end of the name of a file, created as target, during a process flow execution. User needs to specify the following properties while adding the Put-Context-Var action to a process flow:

		Edit Context: Displays a condition screen which allows you to add a new context variable or edit or delete existing context variables. *Label:* The label displayed for the Put-Context-Var action. *Name:* Name for the Put-Context-Var action. *Type:* Displays the type of action selected. This is a read-only field.
Set-Child-Context	This action is used to set the value of Process Flow Context Variable or Activity Context Variable from parent process flow to child process flow. Set-child-Context must be used before Call action. User needs to specify the following properties while adding the Set-Child-Context action to a process flow:	
		Activity: Name of activity in parent process flow, whose 'Activity Context Variable' value will be used to set to the child context variable specified by Child Key. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by Key will be used. *ChildActivityName:* Name of activity of child process flow whose value will be set. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by Child Key will be set. *Childkey:* Name of Context Variable of the child process flow whose value will be set. *ChildName:* Name of the Call or Spawn action, which is used to call or spawn child process flow. Set-child-context uses the ChildName to find out the call or spawn action and the corresponding process flow to set the context variable. *Key:* Name of the context variable of the parent process flow whose value will be used to set the child context variable specified by Child Key. *Label:* The label displayed for the Set-Child-Context action. *Name:* Name for the Set-Child-Context action.
Set-Parent-Context:	This action is used to set the value of 'Process Flow Context Variable' or 'Activity Context Variable' from child process flow to the 'Process Flow Context Variable' or 'Activity Context Variable' of the parent process flow, which initiated the child process flow by call/spawn action. User needs to specify the following properties while adding the Set-Parent-Context action to a process flow:	

		Activity: Name of activity of the child process flow, whose 'Activity Context Variable' value specified by Key will be used to set the parent process flow variable specified by Parent key. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by Key will be used. *Key:* Name of the context variable of the child process flow whose value will be used to set the parent context variable specified by Parent Key. *Label:* The label displayed for the Set-Parent-Context action. *Name:* Name of Set-Parent-Context action *ParentActivityName:* Name of activity of parent process flow whose 'Activity Context Variable' specified by Parent Key will be set. If name of the activity is not specified, then the 'Process Flow Context Variable' specified by Parent Key will be set. *ParentKey:* Name of Context Variable of parent process flow whose value will be set.
Spawn:	This action is used to call another process flow (sub-flow) asynchronously. Users need to specify the following properties while adding the Spawn action to a process flow:	
		flowId: Name of the process flow (sub-flow). You can override its value dynamically during execution of the process flow. To override the flowId you can use custom plugin activity or put-context-var before spawn action. Following is the code which is used in custom plugin activity to override the flow id: context.setActivityParameter("Spawn1","flowId", "192168001158117188341381200001"); where *Spawn1* is the name of the spawn activity and *192168001158117188341381200001* is ID of the child process flow. *Label:* The label displayed for the Spawn action. *Name:* Name for the Spawn action. *resultCtxVarName:* Name of the context variable that contains the status of the sub-flow. Its value is Boolean. Signal: Name for the signal that is generated after the asynchronous process flow (sub-flow) is completed. This property is applicable only when Wait for Child property is

		set to true. *Wait for Child:* Specify whether the parent process flow's end event will wait for the completion of child process flow or not. If this property is set to true, the parent process flow end event will wait till the child process flow is completed. During this period, parent process flow will be in running state. Once the child process flow is completed, it raises the signal specified in the Signal property and then the end event is executed. If the value of Wait for Child property is set to false, the parent process flow does not wait for the child process flow to be completed.
Synch:	This action is used to raise a signal to process engine when an asynchronous activity is completed. The following properties need to be specified while adding the Synch action in a process flow:	
		Label: The label displayed for the Synch action. *Name:* Name for the Synch action. *Signal:* Name of the signal that is specified in the signal properties of the Synch action.
Trace:	This action is used to log a message, which can be used later for information, debugging, or error log purposes. User can check the values of variable at run time in a process flow. This action uses Adeptia Server logging framework. Users need to specify the following properties while adding the Trace action to a process flow:	
		Label: The label displayed for the Trace action. *Log Level:* Log Level is the level at which the message is to be logged. It can be logged at DEBUG, INFO or ERROR levels. Logging level for trace action should not be higher than the logging level of the Process Flow. For example if you have selected 'INFO' logging level in the process flow, you should select 'INFO' or 'ERROR' in trace action. If you select 'DEBUG' logging level in trace action, the trace message will not be logged in process flow log. Similarly while viewing the process flow log, if you select logging level, lower than the logging level of trace action, you cannot see this trace message in process flow logs. For detailed information about Logging Levels, refer to the *Logging* section in the *Administratoristrator Guide*. *Message:* Message that is logged when the trace action is executed. To print the value of a variable in logs, enter $$variablename$$ in the message field. Name: Name for the Trace action.

Wait:	This action waits for certain variable value to be changed in a process flow 'context' to a predefined value for a given timeout. Basically process engine waits for some event to happen then it moves forward. This action is related to polling of a variable and setting some variable in context. Users need to specify the following properties while adding the Wait action to a process flow:	
		Label: The label displayed for the Wait action. *Name:* Name for the Wait action. pollinginterval: Time interval in seconds the wait action will poll for the above specified variable value. *resultCtxVarName*: Name of the context variable that contains the status of the Wait action. Its value is Boolean timeout: Timeout duration in seconds *value:* Value of the context variable. *var:* Name of the context variable for which the Wait action 'waits'.

Steps to add Actions to a Process Flow

1. Click tree structure in the **Repository View** panel. Expand the **Action** list of the Adeptia Server. The list of Adeptia Server actions is displayed (see Figure 16.31).

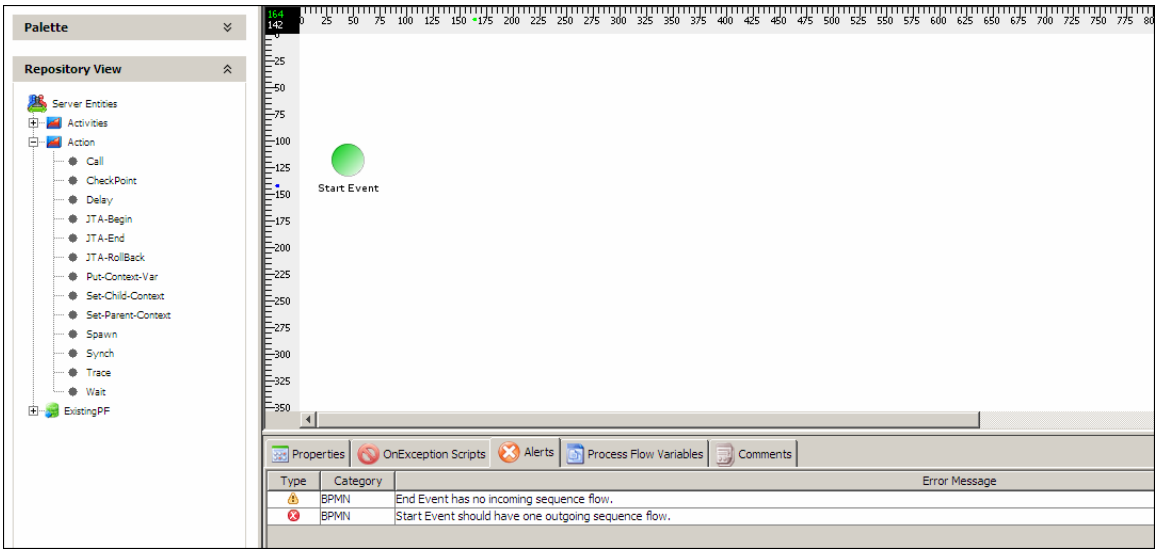


Figure 16.31: List of Actions in Adeptia Server

2. Select the required action and drag it to the Graph Canvas area. The dragged action is displayed in the Graph Canvas (see Figure 16.32).

Figure 16.32: Dragging Action to Graph Canvas

3. Right-click **Action** element and select **View Properties**. The properties of the action element are displayed in the Properties Panel in the Bottom Pane (see Figure 16.33).

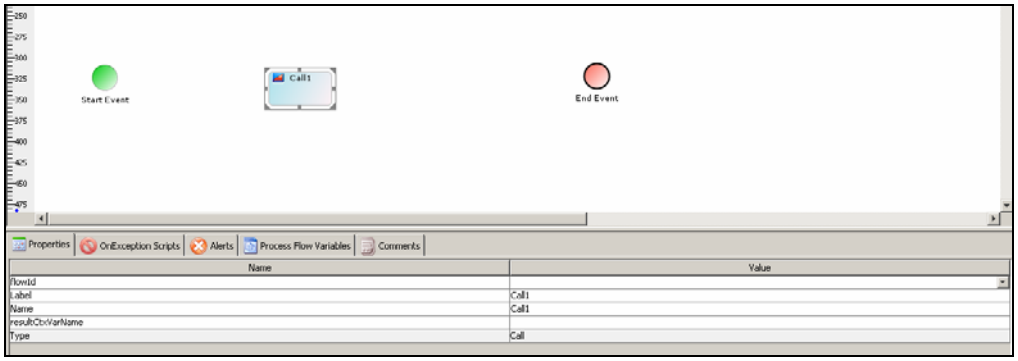


Figure 16.33: Action's Properties

4. Change the required properties and then click the action element in the Graph canvas area.

CREATING PROCESS FLOW VARIABLE

Process Flow Variable are created and used throughout the execution of a process flow.

Steps to create a Process Flow Variable

1. Click **Process Flow Variable** tab in the bottom pane. The Process Flow Variables Panel is displayed with the list of existing variables (see Figure 16.34).

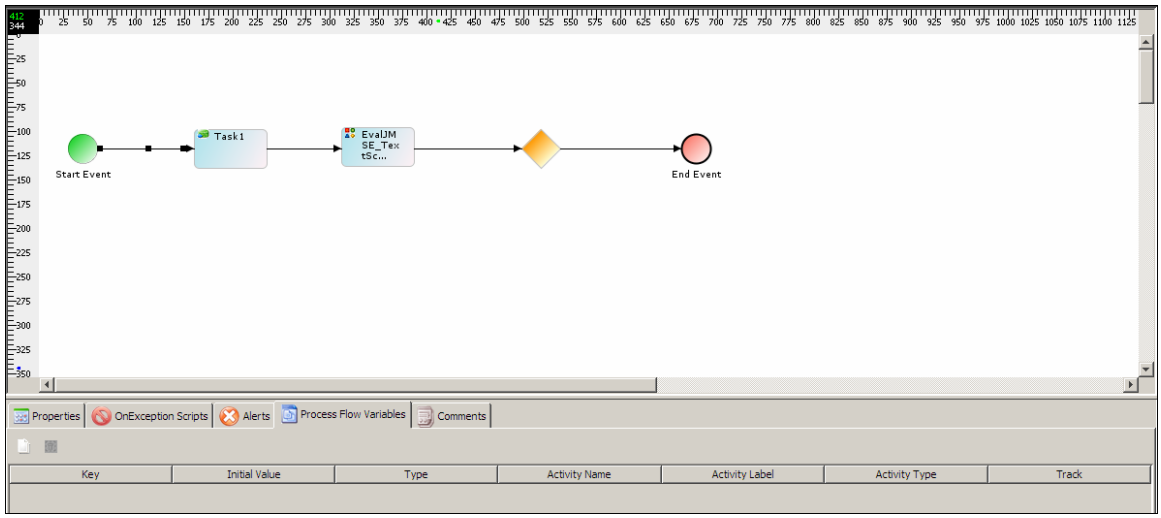


Figure 16.34: Process Flow Variable Panel

2. Click **New Process Flow Variable** ( ) button in the Process Flow Variable Panel. The Process Flow Variable Entry screen is displayed (see Figure 16.35).

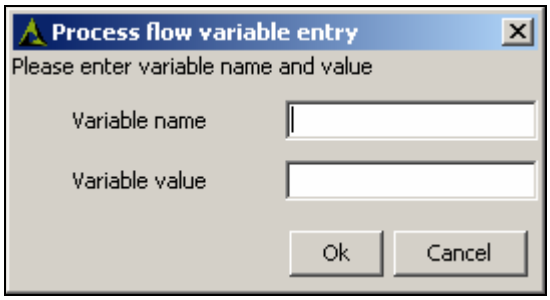


Figure 16.35: Create Process Flow Variable

3. Enter the name for the process flow variable in the *Variable Name* field.
4. Enter the value of process flow variable in the *Variable Value* field.
5. Click **OK** button to save the process flow variable and return to the Graph Canvas.
6. Once a new process flow variable is created, it is added to the list in the Process Flow Variables tab (see Figure 16.36).

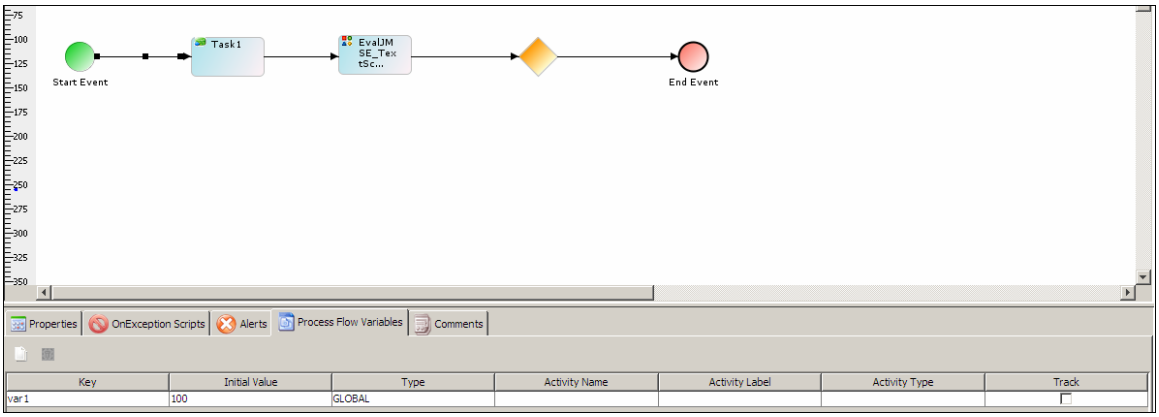


Figure 16.36: Process Flow Variable created

	To track all changes made on the process flow variable, check the *Track* checkbox against the process flow variable.

CREATING CONTEXT VARIABLE

You can create multiple context variables in a process flow using the *PutContextVar* action. Once they are created, you can use these context variables in the context of the process context as and when required.

Steps to create a Context Variable

1. Click tree structure in the **Repository View** panel. Expand the **Action** list and select the **Put-Context-Var** action. Drag it to the Graph Canvas area. The Put-Context-Var action is displayed in the Graph Canvas (see Figure 16.37).

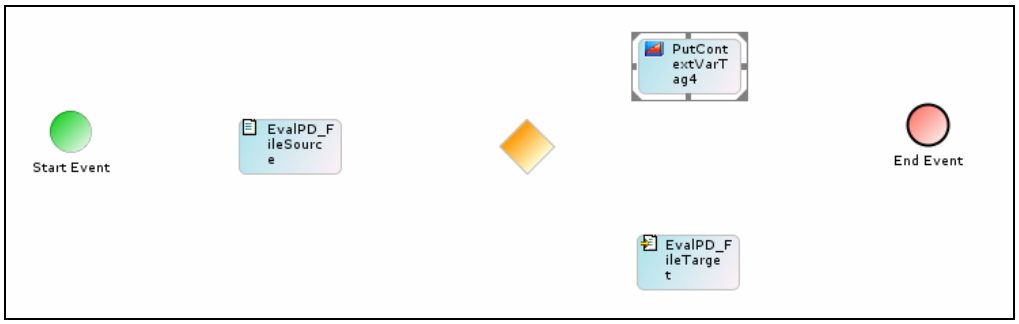


Figure 16.37: Drag Put-Context-Var action in Graph Canvas

2. Right-click **Put-Context-Var** action and select **View Properties**. The properties of the *Put-Context-Var* action are displayed in the Properties Panel in the Bottom Pane (see Figure 16.37).

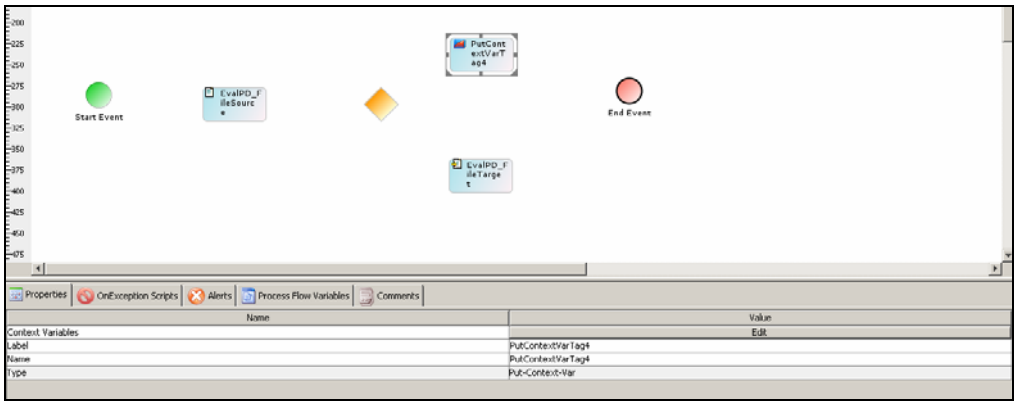


Figure 16.38: Properties of Put-Context-Var action

3. Click **Edit** button. The Edit Context Variables screen is displayed (see Figure 16.39). This screen displays a list of existing context variables and the Add Variable ( ), Edit Variable ( ) and Delete Variable ( ) buttons.

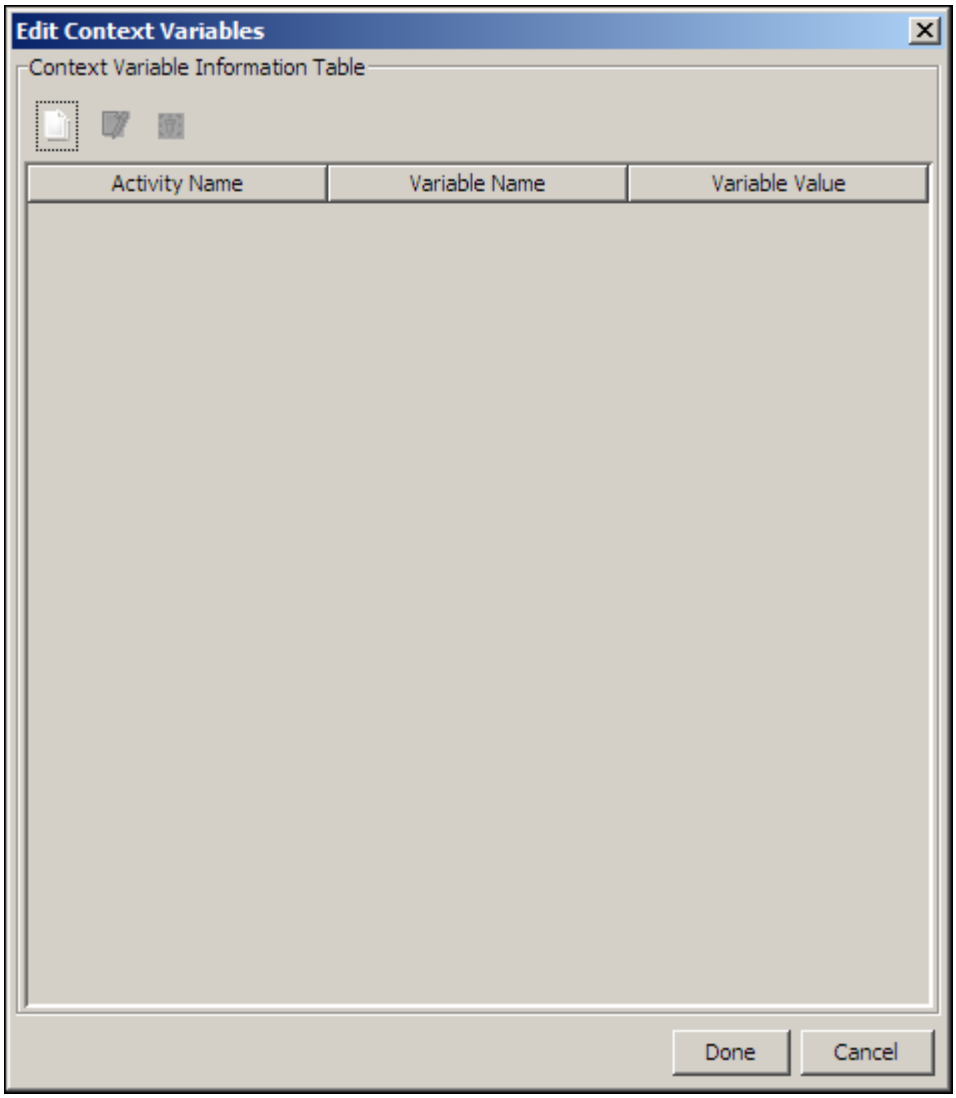


Figure 16.39: Edit Context Variables

	The **Edit Variable** and **Delete Variable** buttons appear as disabled if no context variables have been created or no existing context variables have been selected.

4. Click **Add Variable** ( ) button to create a new context variable. The Context Variable Information screen is displayed (see Figure 16.40).

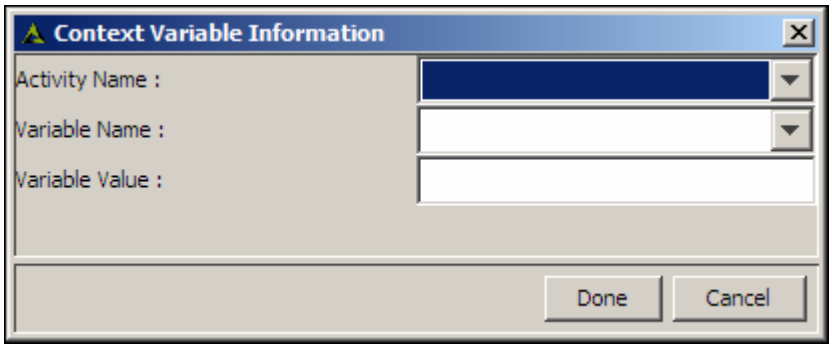


Figure 16.40: Context Variable Information

5. Select the name of the activity for which you want create the context variable, from the *Activity Name* drop-down list. This drop-down lists all the activities that are currently present on the Graph Canvas. Once you select the activity, all the attributes of the selected activity, are listed in the Variable Name drop-down list.

	If the *Activity Name* field is left blank, then a new context variable is declared.

6. Select the attribute for which you want to create the context variable, from the *Variable Name* drop-down list. For example, you can select the File Path field in case of a File Source or File Target activity.

	If the *Activity Name* field is blank, then the current context variables will be listed in the Variable Name drop-down list.

7. Enter the value that you want to set for the attribute/context variable, in the *Variable Value* field.

	You can even append the current date or time stamp with the name of the file in the *Variable Value* field. To do this, click in the *Variable Value* field, and press **<CTRL>** + **<Space Bar>.** A list of the date and time format is displayed. Selecting a format from this list, displays it in the *Variable Value* field. Alternately, you can enter the required file path with the file name as in the example: `C:\target\File_target%%yyyy-mm-dd%%.txt` The Variable Value field does not support '&' and '<' symbols.

8. Click **Done**. This adds the context variable and takes the control back to the Edit Context Variables screen, where the new context variable is added to the list of existing context variables (see Figure 16.41).

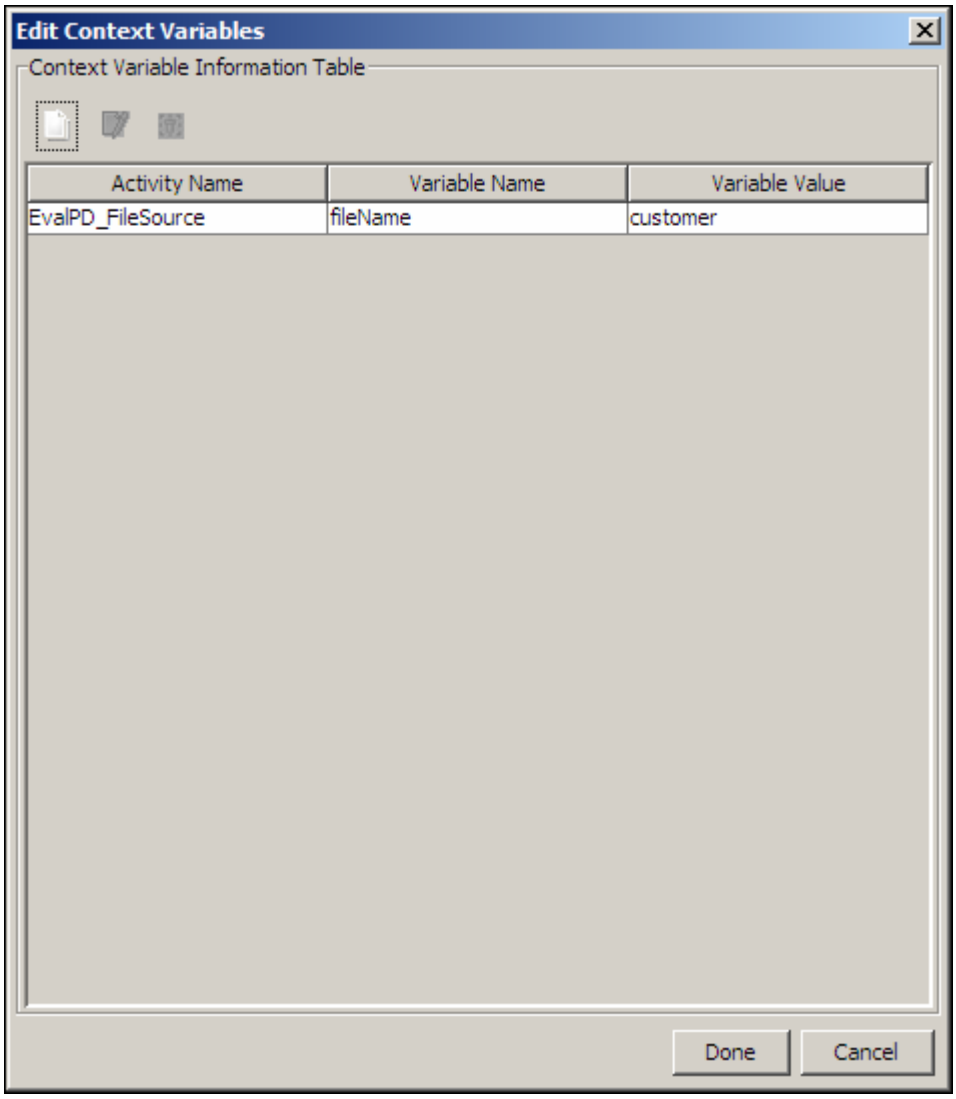


Figure 16.41: Context Variable Added

	You can add multiple context variables (up to a maximum of 100) for the process flow.

9. Click **Done** on the Edit Context Variables screen to add all the displayed context variables to the context of the process flow.

	Similarly, you can edit a context variable, by selecting it from the list of existing context variables on the Edit Context Variables screen. This selection will enable the **Edit Variable** button. Clicking this button will display the Context Variable Information screen with the selected context variable details in edit mode. You can make the necessary changes and click **Done** to save the modified context variable. You can delete a context variable, by selecting it from the list of existing

	context variables on the Edit Context Variables screen. You can select multiple context variables to delete, by pressing **<CTRL>** and the context variables. This selection(s) will enable the **Delete Variable** button. Clicking this button will delete the selected context variable(s).

USING CONTEXT SOURCE AND CONTEXT TARGET

Process Flow keeps it data in memory called Process Flow Context, which is used by the process flow during its execution. When process flow takes any data from the context, it uses the context source activity. Context Source is used to read a variable and stream it out to other activities of the process flow. In addition to that Context Source can also read file from repository (WebDAV or File) and pass it to other activities. Similarly, when process flow sends any data to the context, it uses the context target activity.

Steps to create a Context Source / Target activity

1. Click tree structure in the **Repository View** panel. Expand the **Activities** list and click **Source**. A list of source activities is displayed.
2. Select **Context Source**. Drag the Context Source to the Graph Canvas Area. A small image of the activity is displayed in the Graph Canvas Area (see Figure 16.42).

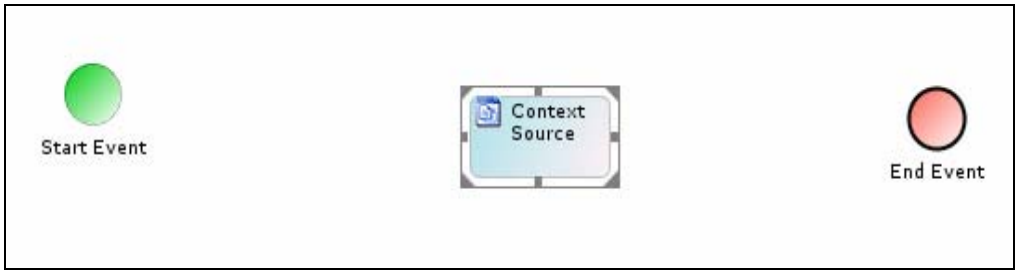


Figure 16.42: Drag Context Source

3. In the Graph Canvas area, right click the **Context Source** and select **View Properties**. This displays all properties for the context source in the Bottom Pane (see Figure 16.43).

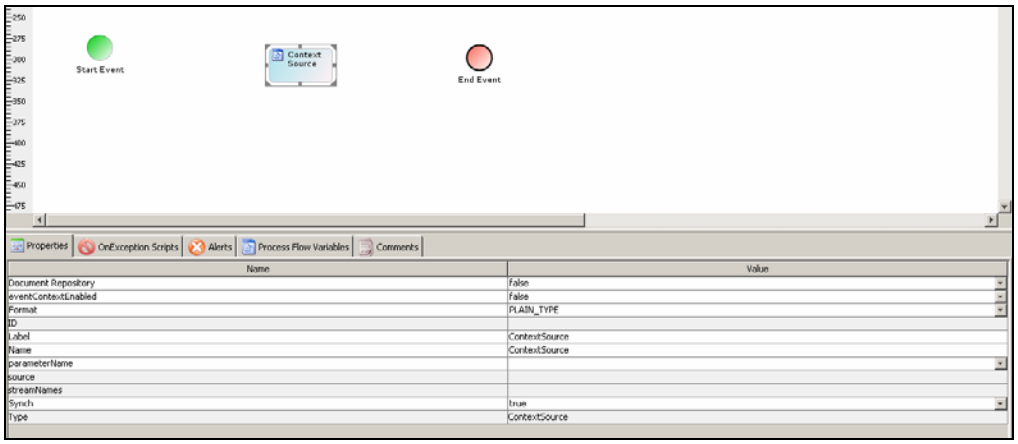


Figure 16.43: Properties of Context Source

4. A list of properties and their descriptions are displayed in the table below.

Table 16.9: Context Source/Target Activity Properties

Properties	Description
Document Repository	This property specifies whether the data to be taken from the value of process flow variable, or from the document repository. If it is set to false, the data is taken from the Process Flow variable, specified by the Parameter Name property. If it is set to true, the source data is taken from the repository file specified by the File Name (with full Path) Property. Specify the path of the document repository and the name of the repository file, from which the data is to be taken, in this field. Repository can be WebDAV repository or file repository. To know more about repository, refer to the *Administratoristrator Guide*.
eventContextEnabled	By default it is set to **NO**. If you want to pass any data from event, select **Yes**. For example, if you are triggering the process flow using Mail Event and you want to pass the content of the mail to the process flow, set this property to yes. To know about events, refer to the **Creating Trigger and Events** section.
Format	Data Format; whether Plain Type or Record Type. Select Plain Type, if data is in Byte Stream. Select Record type, if data is in record format.
Label	Label of the Context Source Activity displayed in the Graph Canvas area.
Name	Name of the Context Source activity. By default, it is same as the Process Variable name.
Repository Folder	Name and path of the WebDAV folder. This property is displayed only if the **Document Repository** property is set to *True*.
parameterName	Select the name of the Process Flow Variable, whose value is to be taken as context source. This property

	is applicable only when the **Document Repository** Property is set to *False*.
Source	Name of the stream being consumed by this activity. This property is applicable only for Context Target. It is non-editable.
streamNames	Name of the output stream name. This property is applicable only for Context Source. It is non-editable.
Synch	Specifies whether the activity will be executed in Synch mode or Asynch mode.
Type	Activity Type; whether Context Source or Context Target. By Default Context Source is selected. If you want to use it as Context Target, select Context Target from the drop-down list.
Version Control	It specifies whether versions are to be maintained for the repository file, which is created by context target. This property is displayed only if the **Document Repository** property is set to *True*. If this property is set to false, then the versions are not maintained. If it is set to True, then all versions are tracked.

OVERRIDING PROCESS FLOW DESCRIPTION AT RUNTIME

You can override the description of process flow during execution. To override the description put-context-var action is used. If the description of a process flow is overridden, in process flow log, new description is shown.

Steps to override Process Flow Description

1. Click tree structure in the **Repository View** panel. Expand the **Action** list and select the **put-context-var** action and drag it to the Graph Canvas area anywhere within the process flow.
2. Connect the put-context-var action as shown in Figure 16.44 .

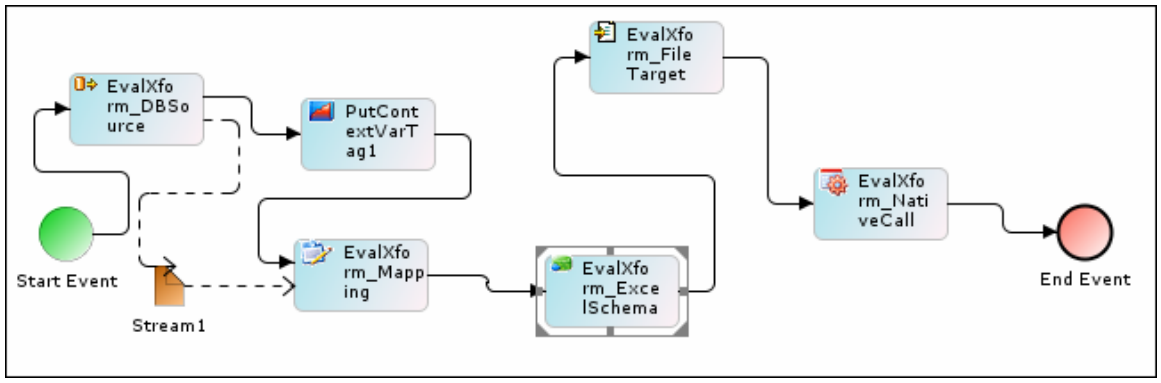


Figure 16.44: Connect Put-Context-Var to activity

3. Right-click **put-context-var** and select **View Properties**. Its properties are displayed in the Properties Panel in the Bottom Pane (see Figure 16.45).

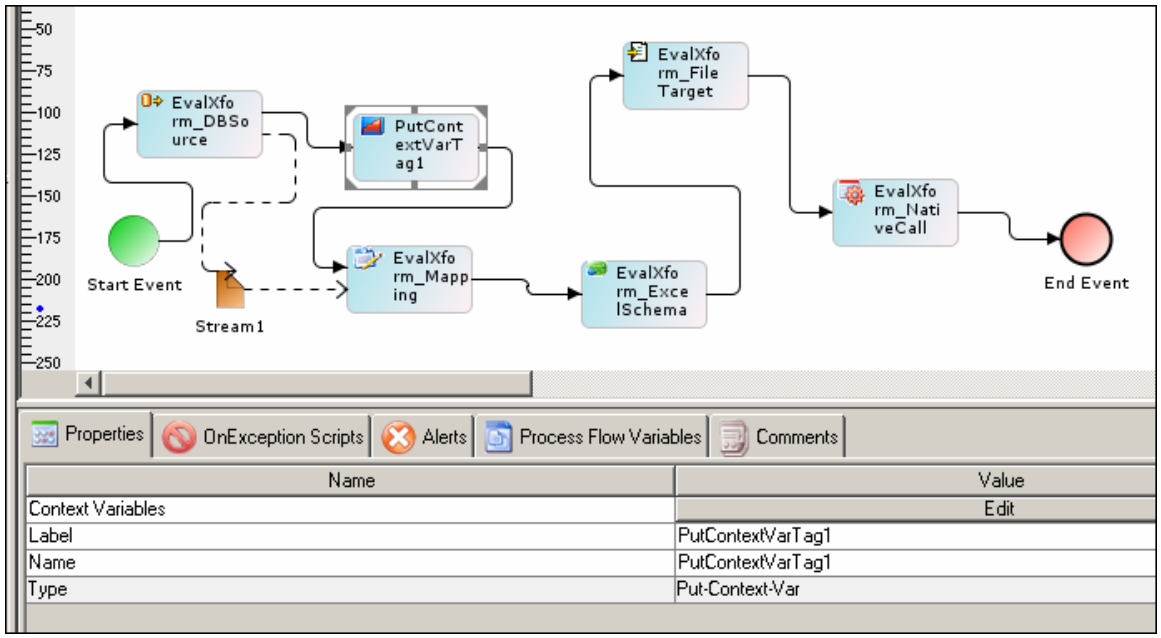


Figure 16.45: Properties of Put-Context-Var

4. Click **Edit** button to edit the value of context variable. The Edit Context Variables dialog box is displayed.

5. Click **Add Variable** ( ) to add new context variable. The Context Variable Information screen is displayed.

6. Leave the *Activity Name* field blank.

7. Select ProcessFlowDescription from *Variable Name* drop-down list.

8. Enter the new description of the process flow in the *Variable Value* field.

9. Click **Done** to close Context Variable Information screen. This takes the control back to the Edit Context Variables screen. The newly created variable is added to the list of existing context variables.

10. Click **Done** to close Edit Context Variables screen and return to Process Designer.
11. Save the process flow and exit from Process Designer.

OVERRIDING ACTIVITY OF A PROCESS FLOW AT RUNTIME

An activity of a process flow can be overridden by another activity during execution of the process flow. For example lets assume the following process flow:

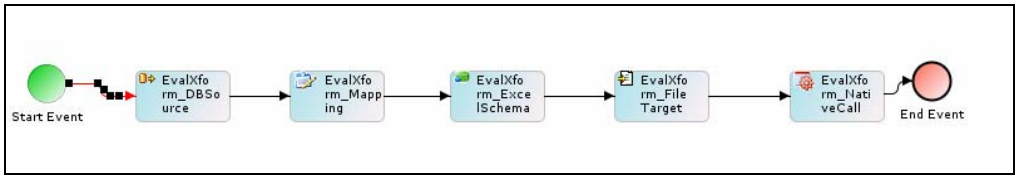


Figure 16.46: Usage Scenario

In Figure 16.46 the process flow uses the EvalXform_ExcelSchema. You can override this activity with any other schema activity (for example, EvalXform_Text Schema) during the execution of the process flow. In this case, EvalXform_TextSchema is executed during the execution of the process flow.

This functionality is used when the actual activity to be executed is decided at execution time, not at design time of the process flow. For example, in a process flow that handles data coming in various formats, you have to first add the schema activity for each data format and then route the data to appropriate format using decision node. This results in the process flow being bulky and unmanageable. At times, you may even need to design one process flow for each data format. Now, this problem can be overcome by using the same process flow with a different schema activity. You can simply override the EvalXform_ExcelSchema with any other schema of your choice. For details on the type of activities that can be overridden refer to the **Activities that can be overridden** section.

There are two ways of overriding an activity in a process flow. They are outlined as:

- **Overriding an activity using Custom Plugin**
- **Overriding an activity using put-context-var**

Overriding an activity using Custom Plugin

You can override an activity by using a custom plugin activity just before the activity, which needs to be overridden by another activity.

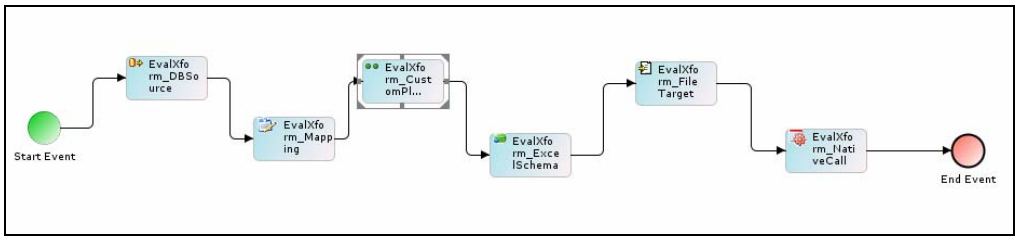


Figure 16.47: Overriding an Activity using Custom Plugin

As shown in Figure 16.47, the *EvalXform_CustomPlugin* activity is used just before the *EvalXform_ExcelSchema* activity.

The **setActivityParameter ( ) API** is used to override the activity.

The sample Java code, which is used to override a schema activity, is displayed in Figure 16.48.

```
context.setActivityParameter(ActivityName,"schemaTypedId",
activityType + ":" + dynamicActivityID);
```

Figure 16.48: Sample Java Code used to Override a Schema Activity

For all other activities the overriding is done using the Java code displayed in Figure 16.49.

```
context.setActivityParameter(activityName,"TypedId",activityType
":" + dynamicActivityID);
```

Figure 16.49: Sample Java Code used to Override Activities

Table 16.10: Arguments used in Java Code

#	Name	Description	Example
1	activityName	Name of the activity, which is to be overridden	EvalXfrom_ExcelSchema
2	activityType	Activity Type of the activity, which will override the existing activity. To know the Activity Type of the activity refer to the **Activities that can be overridden** section.	TextSchema
3	dynamicActivityID	Entity Id of the activity, which will override the existing activity. To know the Entity Id of an activity, in Manage activity screen, click on the activity.	192168001006115537684214000004

Overriding an activity using put-context-var

You can override an activity using put-context-var action just before the activity, which needs to be overridden by another activity.

Steps to override an activity using put-context-var

1. Click tree structure in the **Repository View** panel. Expand the **Action** list and select the **put-context-var** action and drag it to the Graph Canvas area just before the activity, which is to be overridden.
2. Connect the put-context-var action to the activity (see Figure 16.50).

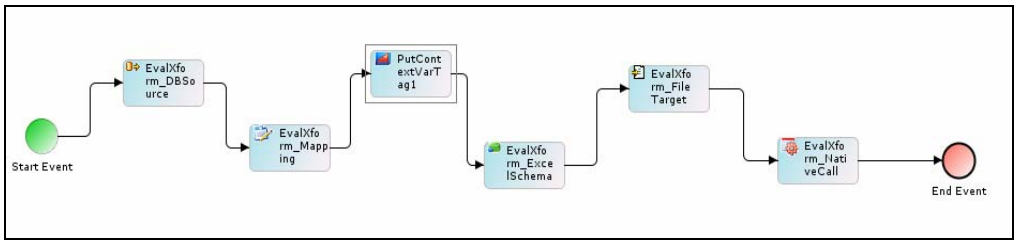


Figure 16.50: Connect Put-Context-Var to activity

3. Right-click **put-context-var** and select **View Properties.** Its properties are displayed in the Properties Panel in the Bottom Pane (see Figure 16.51).

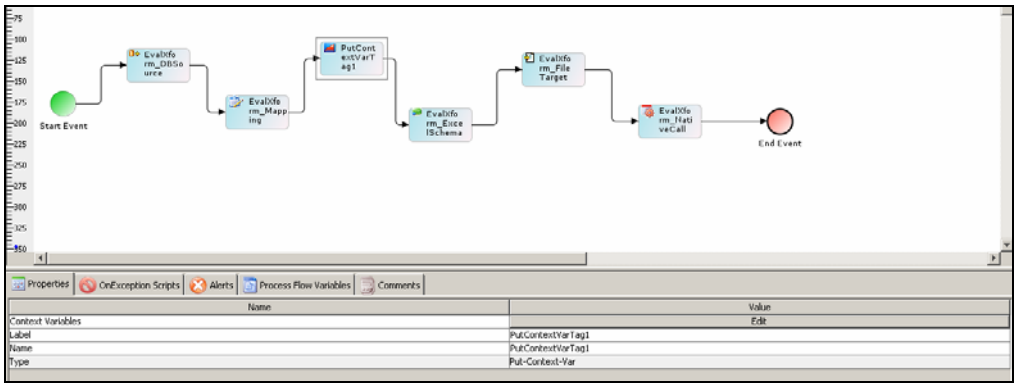


Figure 16.51: Properties of Put-Context-Var

4. Click **Edit** button to edit the value of context variable. The Edit Context Variables dialog box is displayed.
5. Click **Add Variable** ( ) to add new context variable. The Context Variable Information screen is displayed.
6. Select the activity, which is to be overridden (for example, *EvalXform_ExcelSchema*) from *Activity Name* drop-down list.
7. Select SchemaTypedId (for Schema activity) or *TypedId* (for all other activities) from Variable Name drop-down list.
8. Enter the *Activity Type* and the *Entity Id* of the activity, which will override the existing activity in following format, in the *Variable Value* field.

Format : Activity Type: EntitiyID
For Example : TextSchema: 19216800100611553768421400000 4

	To know the Entity Id of an activity, in Manage activity screen, click the activity.

The entered information is displayed as shown in Figure 16.52.

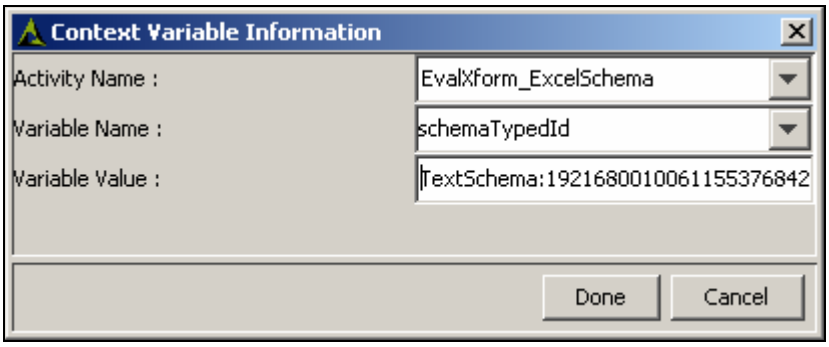


Figure 16.52: Context Variable Details for Overriding an Activity

9. Click **Done** to close Context Variable Information screen. This takes the control back to the Edit Context Variables screen. The newly created variable is added to the list of existing context variables.
10. Click **Done** to close Edit Context Variables screen and return to Process Designer.
11. Save the process flow and exit from Process Designer.

Activities that can be overridden

You can override many types of activities in a process flow. These are outlined as:

- **Source Activity**
- **Target Activity**
- **Schema Activity**
- **Polling Activity**
- **Other Activities**

Source Activity

Any type of source activity can be overridden by another type of source activity. For example, a file source activity can be overridden by an FTP source activity. The types of source activities, that can be overridden and their TypedId are listed in the table below.

Table 16.11: Source Types that can be Overridden

Source Type	Activity Type
Advanced Database Source	AdvancedDatabaseSource
Database Source	DatabaseSource
File Source	FileSource
FTP Source	FtpSource
HTTP Source	HttpSource
JMS Source	JmsSource
LAN File Source	LanFileSource
Mail Source	MailSource
WebDAV Source	WebdavSource

Target Activity

Any type of target activity can be overridden by another type of target activity. For example, a file target activity can be overridden by an FTP target activity. The types of target activities, that can be overridden and their TypeId are listed in the table below.

Table 16.12: Target Types that can be Overridden

Target Type	Activity Type
Advanced Database Target	AdvancedDatabaseTarget
Database Target	DatabaseTarget
File Target	FileTarget
FTP Target	FtpTarget
HTTP Target	HttpPost
JMS Target	JmsTarget
LAN File Target	LanFileTarget
Mail Target	MailTarget
WebDAV Target	WebdavTarget

Schema Activity

Any type of schema activity can be overridden by another type of schema activity. For example, a text schema activity can be overridden by an excel schema activity. The types of schema activities, that can be overridden and their schemaTypedId are listed in the table below.

Table 16.13: Schema Types that can be Overridden

Schema Type	Schema Activity Type
Excel Schema	ExcelSchema
Text Schema	TextSchema
XML Schema	XMLSchema
EDI Schema	EDISchema
Advanced Positional Schema	AdvancePositionalSchema
Positional Schema	PositionalSchema

Polling Activity

Any type of polling activity can be overridden by another type of polling activity. For example, a file polling activity can be overridden by a mail polling activity. The types of polling activities, that can be overridden and their TypeId are listed in the table below.

Table 16.14: Polling Types that can be Overridden

Possible Polling Type	Activity Type
Database Polling	DatabasePollingService
File Polling	FilePollingService
FTP Polling	FtpPollingService
Mail Polling	MailPolling

Other Activities

All other types of activity can be overridden by exactly the same type of activity. For example, a Custom Plugin activity can be overridden by another Custom Plugin activity only. Similarly, a Data Mapper activity can be overridden by another Data Mapper activity only. All other types of activities, that can be overridden and their TypeId are listed in the table below.

Table 16.15: Other Activity Types that can be Overridden

Activity	Activity Type
Data Mapping	DataMapping
Record to Record	ScriptedRecord2RecordTransformer
Custom Plugin	CustomPlugin
Human Workflow	HumanInteraction
Context Download	ContextDownload
Context Upload	ContextUpload
MIME Message: Decoder Encoder	 MessageExtractor MessageComposer
Custom Report	IndigoReport
Native Call	NativeCall
Mail Notification	MailNotification
Web Service: WsMessage Call WsRpc Call	 WsMessageCall WsRpcCall

Overriding subject and body of email sent for a Human Workflow Task

The subject of emails sent when a Human Workflow task is created/updated, is already pre-defined in the code. Now, you can override this subject by using *put-context-var* action.

Steps to override email subject using put-context-var

1. Click tree structure in the **Repository View** panel. Expand the **Action** list and select the **put-context-var** action and drag it to the Graph Canvas area before the human work flow task whose email subject is to be overridden.
2. Connect the *put-context-var* action to the task (see Figure 16.53).

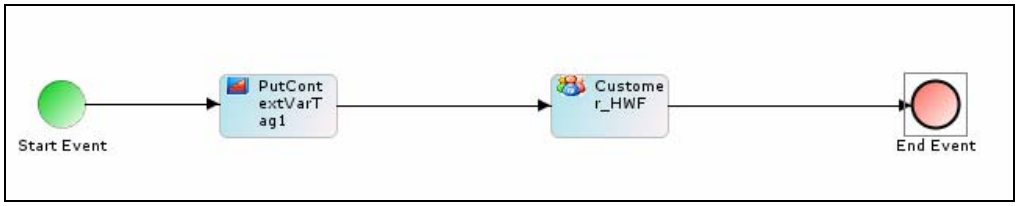


Figure 16.53: Connect put-context-var to Human Workflow task

3. Right-click **put-context-var** and select **View Properties**. Its properties are displayed in the Properties Panel in the Bottom Pane.
4. Click **Edit** button to edit the value of context variable. The Edit Context Variables dialog box is displayed.
5. Click **Add Variable** ( ) to add new context variable. The Context Variable Information screen is displayed.
6. Select the task, which is to be overridden (for example, Customer_HWF) from *Activity Name* drop-down list. All variables of this task are listed in the Variable Name drop-down list.
7. Select **emailSubject** from *Variable Name* drop-down list.
8. Enter the new email subject that you want to display, in the *Variable Value* field. You can also define the subject as extracted from a variable, by entering $$ variable name $$ in the *Variable Value* field.

   The entered information is displayed as shown in Figure 16.54.

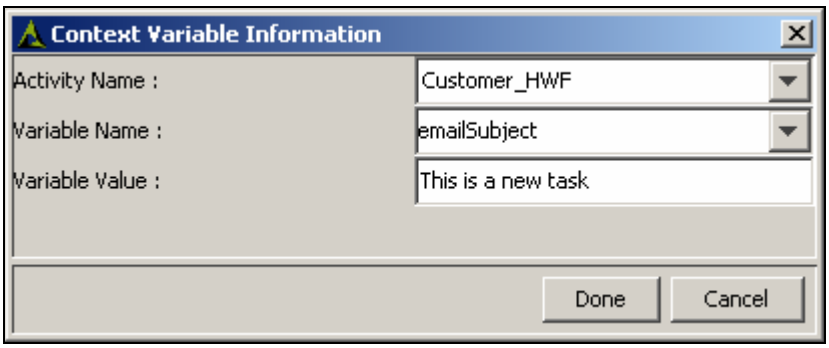


Figure 16.54: Context Variable Details for Overriding email subject

9. Click **Done** to close Context Variable Information screen. This takes the control back to the Edit Context Variables screen. The newly created variable is added to the list of existing context variables.
10. Click **Done** to close Edit Context Variables screen and return to Process Designer.
11. Save the process flow and exit from Process Designer. When you execute this process flow, and an email for a new task is sent, then 'New Task' is appended in the email subject. This is pre-defined in the code and is displayed when you override the email subject. It is subject to change, based on the action performed. If an existing task is deferred, then 'Deferred Task' is appended.

	Similarly, you can dynamically override the first line of the email subject.

Overriding Assignee User of a Human Workflow Task

You can dynamically override the assignee (user to whom task is assigned) of a Human Workflow activity, during the execution of a process flow using *put-context-var* action.

Steps to override assignee using put-context-var

1. Click tree structure in the **Repository View** panel. Expand the **Action** list and select the **put-context-var** action and drag it to the Graph Canvas area before the Human Work flow task assignee is to be overridden.
2. Connect the *put-context-var* action to the task (refer to Figure 16.53).
3. Right-click **put-context-var** and select **View Properties**. Its properties are displayed in the Properties Panel in the Bottom Pane.
4. Click **Edit** button to edit the value of context variable. The Edit Context Variables dialog box is displayed.
5. Click **Add Variable** ( ) to add new context variable. The Context Variable Information screen is displayed.
6. Select the task, which is to be overridden (for example, *Customer_HWF*) from *Activity Name* drop-down list. All variables of this task are listed in the *Variable Name* drop-down list.
7. Select *hiReceiverUsers* from *Variable Name* drop-down list.
8. Enter the User Id of the user to whom you want to assign the Human Workflow task, in the *Variable Value* field. The entered information is displayed as shown in Figure 16.55.

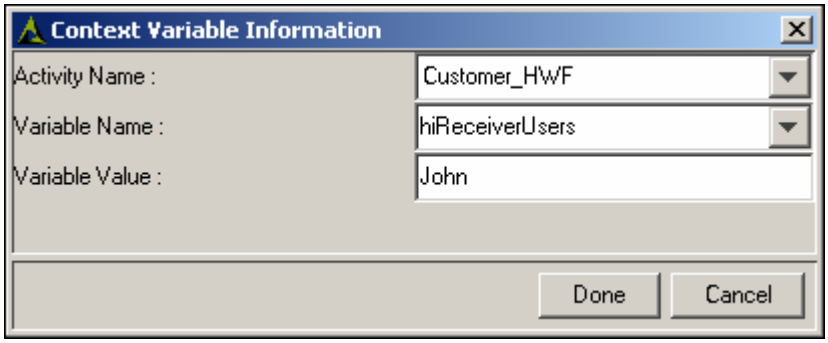


Figure 16.55: Context Variable Details for Overriding email subject

9. Click **Done** to close Context Variable Information screen. This takes the control back to the Edit Context Variables screen. The newly created variable is added to the list of existing context variables.
10. Click **Done** to close Edit Context Variables screen and return to Process Designer.
11. Save the process flow and exit from Process Designer. When you execute this process flow, the process will be assigned to user (for example *John* instead of *Administrator*) to whom it was originally assigned.

PROCESSING RECORD QUEUE

By default, during the execution of process flow, the entire input data is processed at a time. Adeptia allows you to process the input data record-by-record. To process the data record-by-record, Record Queue Processor is used. Using Record Queue Processor, you can:

- Process the input data record-by-record
- Specify the number of records to be processed

How Record Queue Processor works?

Record Queue processor consists of *Record Queue Producer* and *Record Queue Receiver*. *Record Queue Producer* is an asynchronous activity, which sets records one by one in a queue and waits for it to get consumed by *Record Queue Receiver*. *Record Queue Receiver* consumes the record from the queue and produces a stream. This stream can be further processed by other activities of the process flow. Once the record is consumed by *Record Queue Receiver*, *Record Queue Producer* sets the next record and waits for it to get consumed by *Record Queue Receiver*. This loop continues until all the records are queued and processed. Once all the records are processed, the loop is broken and the process flow stops.

The usage of the Record Queue Processor is explained in Figure 16.56.

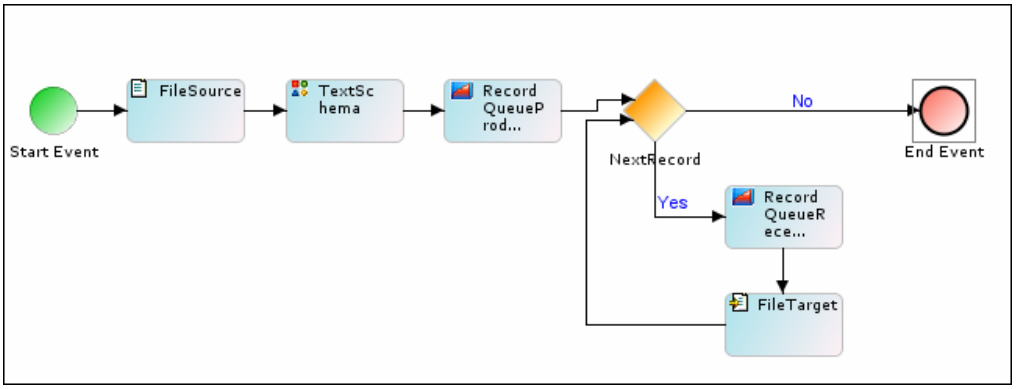


Figure 16.56: Using Record Queue Processor

As shown in the figure above, data from *File_Source* is consumed by *Text_Schema*, which further passes it to Record Producer. Record Producer takes the first record and sets it into Queue as specified by Record Queue Producer properties. Gateway is used to check availability of records. Following is the code, which is used at Gateway to check the availability of the record in queue.

```
String       queue       =       context.get("Service.queueName.nextRecord")
if(queue.equals("true")){
 return true;
}
return false;
```

Where queueName is the name of the queue specified in *Record Queue Producer*. "*nextRecord*" is a variable which is used to decide whether the record is available in the queue or not. Value of the variable *NextRecord* can be true or false. When *Record Queue Producer* sets a record in the queue, the value of *nextRecord* variable become true. If the value is true, it means the next record is available in queue for processing. In this case *Record Queue Receiver* takes the record from queue, changes the value of *nextRecord* variable to false and passes the record to file

target. In the meantime *Record Queue Producer* again sets the record in the queue and changes the value of *nextRecord* to true and waits for it to get consumed by *Record Queue Receiver*. If the value of *NextRecord* variable is not changed to true, it means there is no record available for processing and the record queue processor is stopped.

Steps to process records using Record Queue Producer and Record Queue Receiver

1. Click tree structure in the **Repository View** panel. Expand the **Activities** list of the Adeptia Server and select and drag the *File source* and *Text schema* activities to the Graph Canvas area.
2. Select the **RecordQueueProducer** activity and drag it to the Graph Canvas area.
3. Drag a gateway element to the Graph Canvas area.
4. Drag a *File target* to the Graph Canvas area.
5. Select the **RecordQueueReceiver** activity and drag it to the Graph Canvas area.
6. Connect all the activities as shown in Figure 16.56.
7. Right-click **RecordQueueProducer** and select **View Properties**. Its properties are displayed in the Properties Panel in the Bottom Pane (see Figure 16.57).

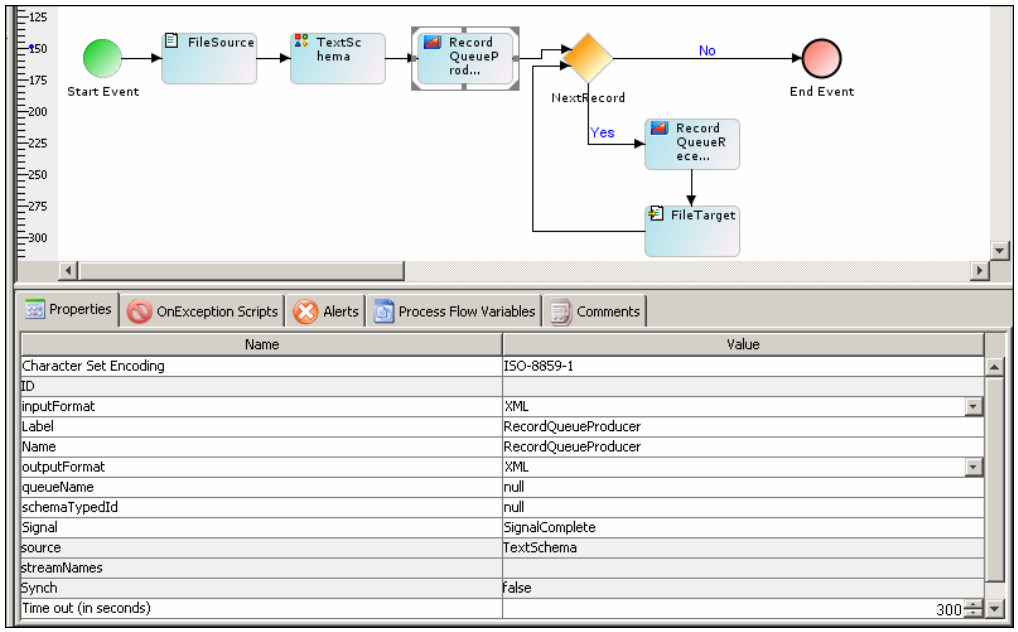


Figure 16.57: Properties of RecordQueueProducer

8. Set the appropriate properties for the RecordQueueProducer. For details on the properties and their description, refer to the table below.

Table 16.16: Record Queue Producer Properties

Properties	Description
Character Set Encoding	Character set encoding that is used for parsing, incase input data is XML. By default it is ISO-8851-1

InputFormat	Format of the input data. It can be XML or Native.
Label	Label of the Record Queue Producer activity displayed in the Graph Canvas area.
Name	Name of the Record Queue Producer activity. By default, it is Record Queue Producer.
OutputFormat	Format of the output record. It can be XML or Native.
schemaTypedId	TypedId and the 30 digit activity ID of the source schema separated by colon (:). For example TextSchema:192168001158117196729809300003 To know the TypedID of Schema refer to Table 16.13. To know the 30 digit activity, click the activity name from the manage page. The 30 digit activity Id along with other properties are shown.
Source	Name of the Activity, which is passing the record to Record Queue Producer. By default this field is populated. You cannot edit this field.
streamNames	Record Queue Producer doesn't produce any stream. This field remains blank.
queueName	Enter any queue name. This will be the queue name in which records are set. QueueName must be same as sourceQueueName of Record Queue Receiver activity used in the process flow.
Synch	Specifies whether the activity will be executed in Synch mode or Asynch mode. Record Queue Producer is always executed in asynch mode. This field is non editable. To know more about Synch and Asynch mode of execution, refer to the section **Working with Process Flow**.
Type	Type of the activity. By default this field is populated. This field is non editable.

9. Right-click **RecordQueueReceiver** and select **View Properties**. Its properties are displayed in the Properties Panel in the Bottom Pane (see Figure 16.58).

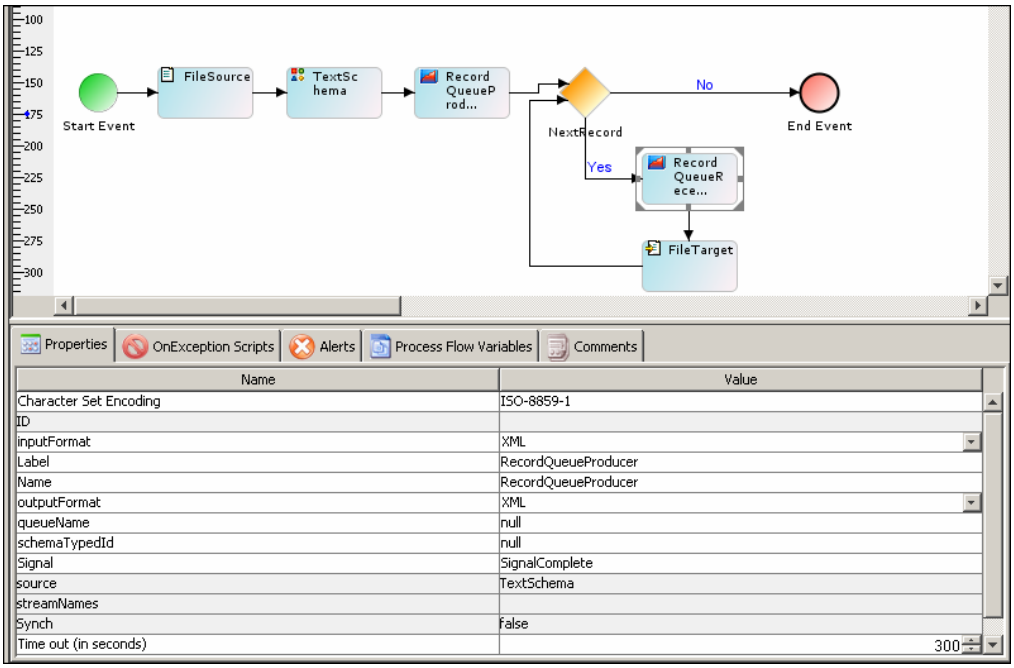


Figure 16.58: Properties of RecordQueueReceiver

10. Set the appropriate properties for the *RecordQueueReceiver*. For details on the properties and their description, refer to the table below.

Table 16.17: Record Queue Receiver Properties

Properties	Description
Label	Label of the Record Queue Producer activity displayed in the Graph Canvas area.
Name	Name of the Record Queue Producer activity. By default, it is Record Queue Producer.
Source	Record Queue Receiver does not consume any stream. This field remains blank.
streamNames	Name of the stream produced by Record Queue Receiver.
SourceQueueName	Enter the name of the queue from which Record Queue Receiver will fetch the record. sourceQueueName must be same as QueueName of Record Queue Producer activity used in the process flow.
Synch	Specifies whether the activity will be executed in Synch mode or Asynch mode. Record Queue receiver can be executed in Asynch or Synch mode. To know more about Synch and Asynch mode of execution, refer to the section **Working with Process Flow**.
Type	Type of the activity. By default this field is populated. You cannot edit this field.

Based on the selected properties for the *RecordQueueProducer* and *RecordQueueReceiver*, and the code specified for the Gateway element, the records are processed.

	Record processing can be stopped based on specified conditions. If the condition is met, then the signal is set by the *RecordQueueReceiver* to stop further processing of records.

ADDING CONDITIONS IN PROCESS FLOW

Conditions determine whether a certain transition is executed in a process flow. Conditions are used to change the direction of the process flow based on a decision. There are three types of conditions:

- Process Flow Variable Condition
- Java Condition
- Expression Condition Builder

Process Flow Variable Condition

A transition can have condition based on the value of the context variable present in the current process flow. Once the condition is met, transition takes place. The Process Flow Variable Condition can be of two types:

- **Activity Attributes Condition**: Activity Attribute Condition is used to define condition based on the value activity specific context variable in a process flow.

- **Other Condition**: Other Condition is used to define condition using pre-created context variables.

	Process Designer allows Conditions to be added only for uncontrolled or default flow having gateway as its source. User cannot enter Condition for Association and Direction association flow under any circumstances.

Adding Condition Using Process Flow Activity Attributes

Steps to add a Condition in a Process Flow using Activity Attributes

1. Select the **Gateway** () element in the Palette and drag it to required place in the Graph canvas area. The Gateway element is displayed in the Graph canvas area (see Figure 16.59).

Figure 16.59: Drag Gateway Element to Graph Canvas Area

2. Connect the activities with the Gateway element using uncontrolled or default control flow (see Figure 16.60).

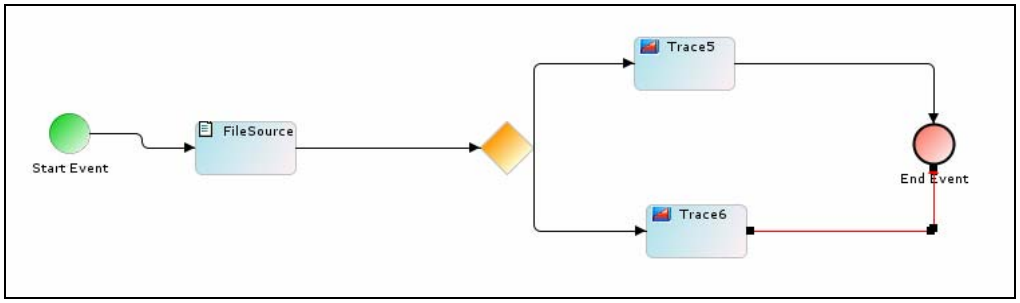


Figure 16.60: Connecting Elements

To learn how to connect activities, refer to the **Creating Process Flow** section.

3. To add Condition, right-click control flow and select **View Properties**. The properties of the control flow are displayed in the Properties Panel in the Bottom Pane (see Figure 16.61).

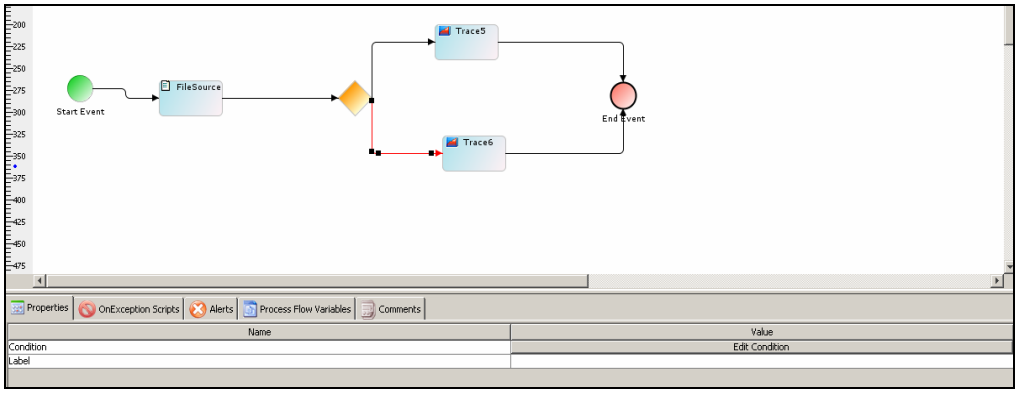


Figure 16.61: Change Gateway Element Properties

4. Click **Edit Condition** button. The Condition Wizard screen is displayed (see Figure 16.62).

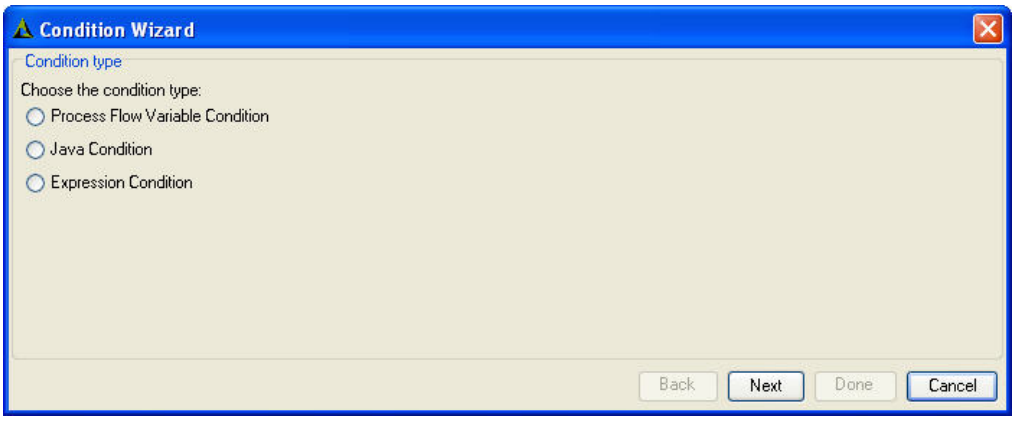


Figure 16.62: Condition Wizard

5. Select the **Process Flow Variable Condition** and click **Next** button. The Choose Process Flow Variable Condition Type screen is displayed (see Figure 16.63).

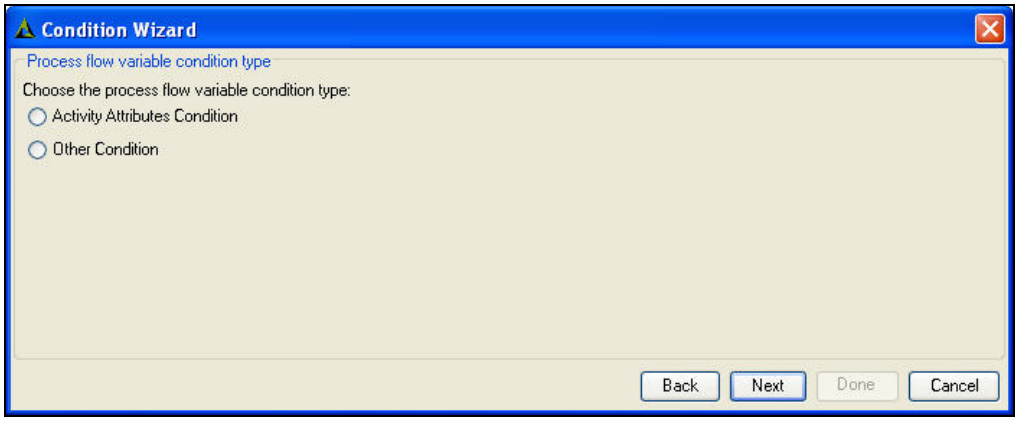


Figure 16.63: Select Process Flow Condition Type

6. Select **Activity Attribute Condition** and then click **Next** button. The Activity Attributes Condition screen is displayed (see Figure 16.64).

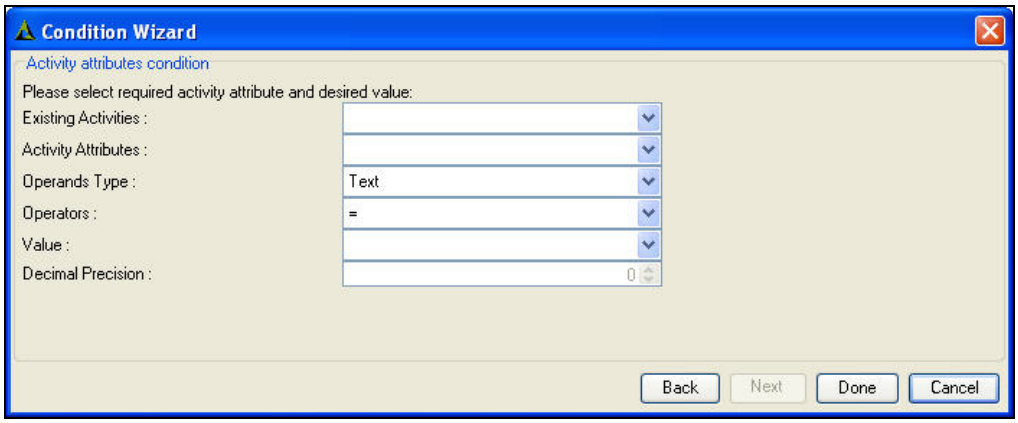


Figure 16.64: Activity Attributes Condition

7. Select the activities of the process flow from the *Existing Activities* drop-down list.

8. Select the attribute of the selected activity from the *Activity Attributes* drop-down list.
9. Select the data types of the value contained by the above specified attribute from *Operand Type* drop-down list. The data types supported for the value are listed in the table below.

Table 16.18: Data Types Supported for Operand Type Value

Data Type	Description
Number	This data type is selected if the specified activity attribute contains numeric value. For example, 1, 12.
Text	This data type is selected if the specified activity attribute contains text value. For example, abc, xyz.
Decimal	This data type is selected if the specified activity attribute contains decimal value. For example, 10.211, 100.50. The decimal precision can be defined under the decimal precision text field.

10. Select the operator's type from the *Operators* drop-down list. The operators supported are "=", "! =", ">", "<", ">=", "<=".
11. Enter or select the value of the above specified attribute from *Value* drop-down list.
12. Specify the decimal precision (only if operand type is "Decimal") under the *Decimal Precision* text field.
13. Click **Done** button to close the Condition Wizard and return to the Graph Canvas.
14. Repeat steps 4 to 14 to add condition on another control flow.

Adding Condition Using Process Flow Variable Other Condition

Using Process Flow Variable we specify that if the value of a specified variable is equal to the defined value, the transition will execute.

Pre-Requisite:

- Process Flow variable must be created. To learn how to create process flow variable, refer to the **Creating Process Flow Variable** section.

Steps to add a Condition in a process flow using Process Flow Variable

1. Select the **Gateway** ()element in the Palette and drag it to required place in the Graph canvas area. The Gateway element is displayed in the Graph Canvas area (refer to Figure 16.59).
2. Connect the activities with the Gateway element using uncontrolled or default control flow (refer to Figure 16.60).

	To learn how to connect activities, refer to **Creating Process Flow** section.

3. To add Condition, right-click control flow and select **View Properties**. The properties of the control flow are displayed in the Properties Panel in the Bottom Pane (refer to Figure 16.61).
4. Click **Edit Condition** button. The Condition Wizard screen is displayed (refer to Figure 16.62).
5. To use Condition using Process Flow Variable, select the **Process Flow Variable Condition** and click **Next** button. The Select Process Flow Condition Type screen is displayed.
6. Select Other Condition and then click Next button. The Other Condition Process Details screen is displayed (see Figure 16.65).

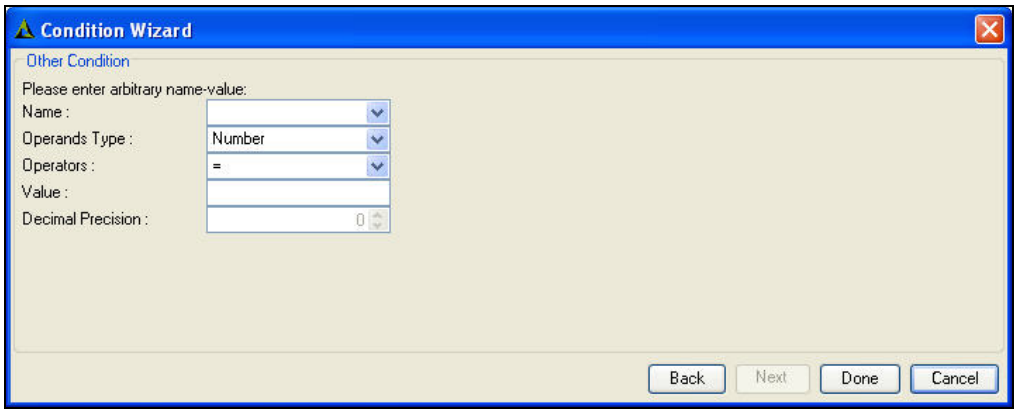


Figure 16.65: Other Condition Process Details

7. Select the Process Flow Variable from the *Name* drop-down list and enter the Value in the *Value* field.
8. Select the data types of the value contained by the above specified context variable from *Operand Type* drop-down list. For information regarding data types supported for the value refer to Table 16.18.
9. Select the operator's type from the *Operators* drop-down list. The operators supported are "=", "! =", ">", "<", ">=", "<=".
10. Enter the value of the above specified context variable in the *Value* field.
11. Specify the decimal precision (only if operand type is "Decimal") under the *Decimal Precision* field.
12. Click **Done** button to apply the condition and return to the Graph Canvas.

Java Condition

A transition can have condition, which can be in form of script. Once the condition is met, transition takes place. Java Condition is a part of transition.

The sample conditional transition is given as below:

```
<transition from="state.2" to="state.4">
          <guard>
             <indigo:scripted-guard>
                   <![CDATA[
                                          ..script
                         ]]>
             </indigo:scripted-guard>
          </guard>
</transition>
```

A State in Process XML either has normal transition(s) and/or conditional transition(s). If there are more than one transition from a State, only one transition takes place at a time and it depends upon the order of their occurrence and condition satisfaction (in case conditional transition). Conditional transition takes place based on appropriate condition defined in form of java code. When the condition is met, transition takes place. A Condition is satisfied when java code script returns true. The Java code script should return true or false otherwise exception is raised.

Adding Condition Using Java Condition

Steps to add a Condition in a process flow using Java Condition

1. Select the **Gateway** ()element in the Palette and drag it to required place in the Graph canvas area. The Gateway element is displayed in the Graph canvas area (refer to Figure 16.59).
2. Connect the activities with the Gateway element using uncontrolled or default control flow (refer to Figure 16.60).

To learn how to connect activities, refer to **Creating Process Flow** section.

3. To add Condition, right-click control flow and select **View Properties**. The properties of the control flow are displayed in the Properties Panel in the Bottom Pane (refer to Figure 16.61).
4. Click **Edit Condition** under the Properties tab. The Condition Wizard screen is displayed.
5. To define Condition using Java Condition, select the **Java Condition** and click **Next** button. The Java Condition screen is displayed (see Figure 16.66).

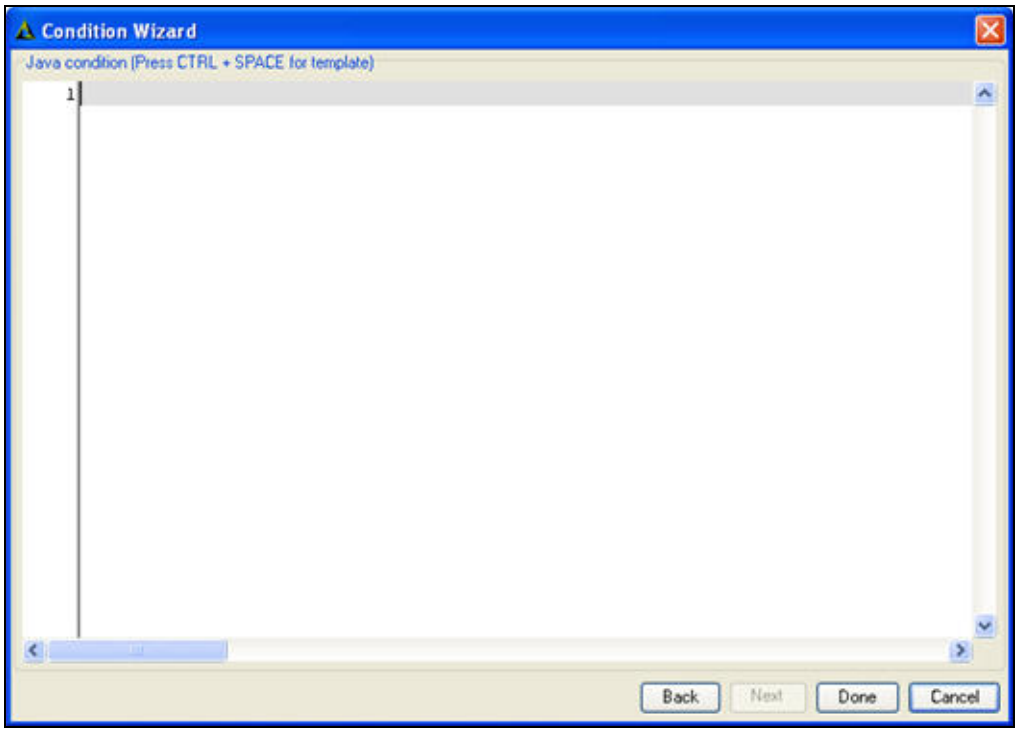


Figure 16.66: Enter Java Condition

6. Enter the Java Code in the *Java Condition* field and click **Done** button to return to the Graph Canvas area.

	You can use **<CTRL>+<Space Bar>** to view pre-defined template of Java Code, which can be used in creating Java Condition. You can select any of them and edit it according to your requirement.

7. Similarly, repeat steps 4 to 7 to add Condition to another control flow.

Expression Condition Builder

A transition can have condition based on an expression. This expression is built using the Activity Attributes or the Process Flow variable with the 'AND' & 'OR' condition.

Adding Condition Using Expression Builder

Using Activity Attributes and Process Flow variables, and the 'AND' or 'OR' conditions, you can generate an expression.

Steps to add a Condition in a process flow using Expression Builder

1. Select the **Gateway** ( ) element in the Palette and drag it to required place in the Graph canvas area. The Gateway element is displayed in the Graph Canvas area (refer to Figure 16.59).
2. Connect the activities with the Gateway element using uncontrolled or default control flow (refer to Figure 16.60).

	To learn how to connect activities, refer to Creating Process Flow section.

3. To add Condition, right-click control flow and select **View Properties**. The properties of the control flow are displayed in the Properties Panel in the Bottom Pane (refer to Figure 16.61).
4. Click **Edit Condition** button. The Condition Wizard screen is displayed.
5. To use Condition using Expression Builder, select the **Expression Condition** and click **Next** button. The Condition Wizard screen is displayed (see Figure 16.67).

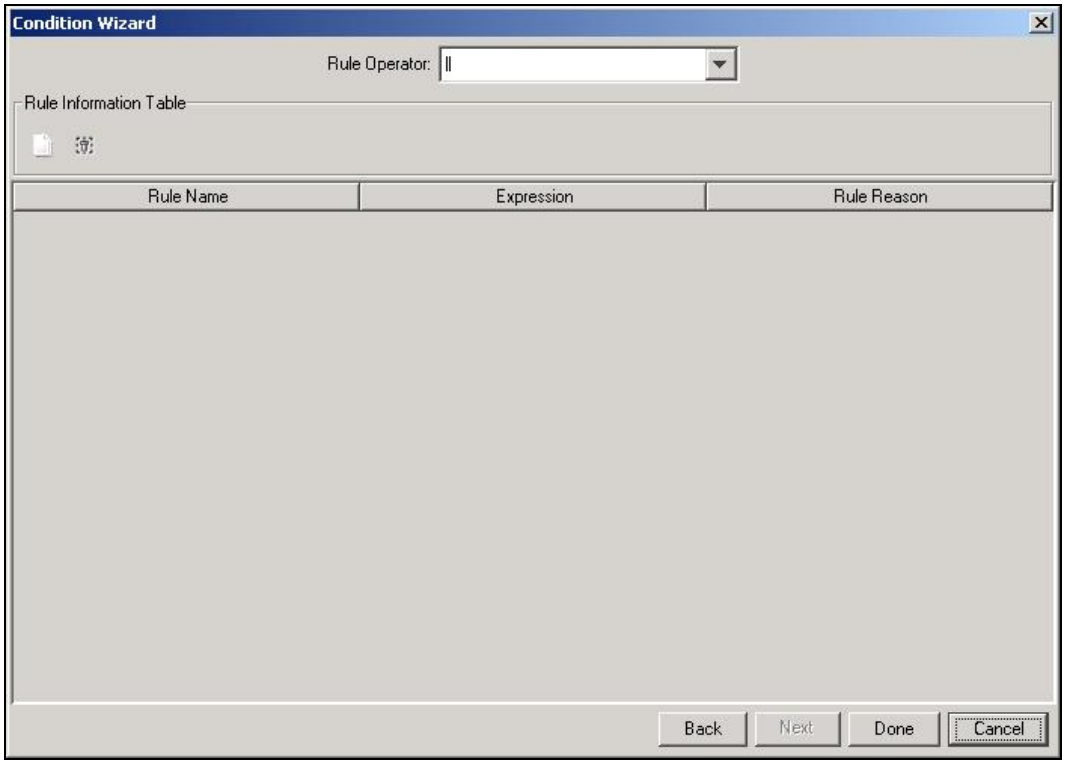


Figure 16.67: Condition Wizard

For defining expressions you need to define rules. A rule supports multiple expressions. You can link the multiple expressions using the rule operator.

6. Click **Add Rule** ( ) to add a new rule. This displays the Rule Information screen (see Figure 16.68).

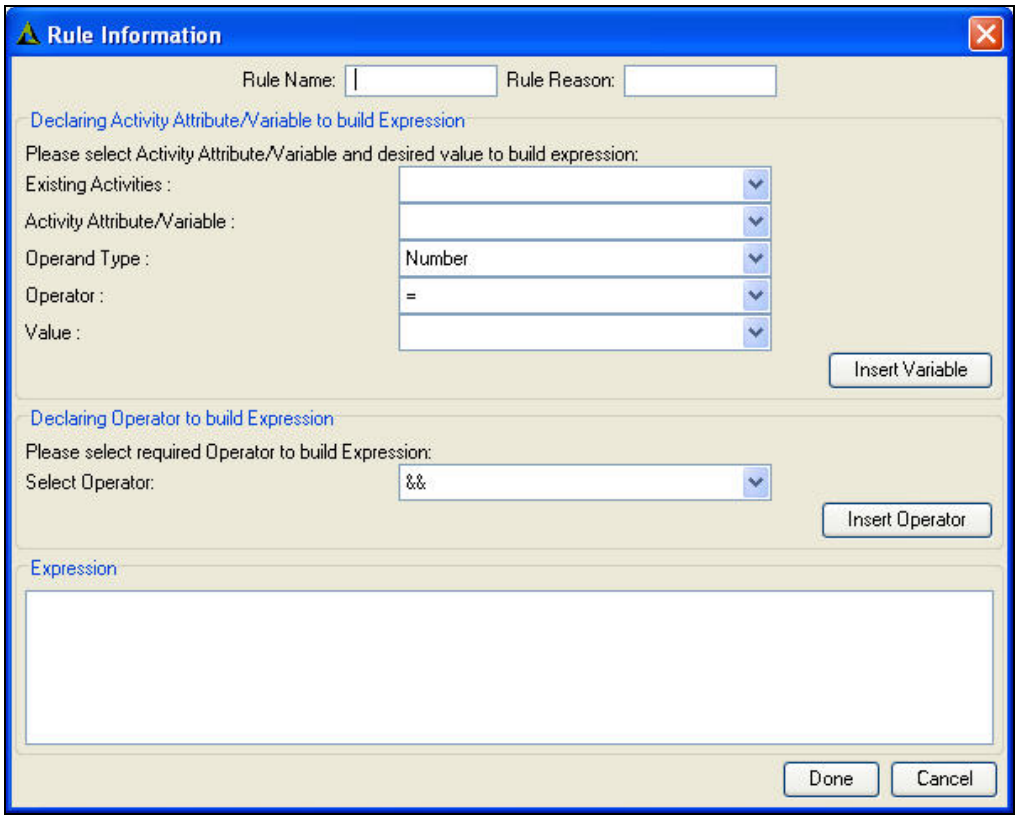


Figure 16.68: Rule Wizard

7. Enter the name of the Rule (for example, Rule 1) in the *Rule Name* field.
8. Enter the reason on the basis of which the rule will be evaluated as true or false (for example, Rule 1 is valid), in the *Rule Reason* field. This is set in the context.
9. To build the expression, you first need to select the Activity Attribute or Process Flow variable to be used in the expression. Select the activity to be used, from the *Existing Activities* drop-down menu. This drop-down is populated with the current activities. Alternately, if you select a blank value, then a process flow variable is selected.
10. Select the activity attribute or the process flow variable, from the *Activity Attribute/Variable* drop-down menu. This drop-down is populated with values based on the selection in the *Existing Activities* field. If an activity is selected, then this drop-down lists all attributes of the selected activity. If a process flow variable is selected, then this drop-down lists the currently available process flow variables.
11. Select the data type of the selected activity attribute or process flow variable, from the *Operand Type* drop-down menu. This drop-down is populated with values of *Number, Text (String)* and *Decimal*. These are description are outlined in the table below.

Table 16.19: Data Types Supported for Operand Type Value

Data Type	Description
Number	This data type is selected if the specified activity attribute contains numeric value. For example, 1, 12.

Text	This data type is selected if the specified activity attribute contains text value. For example, abc, xyz.
Decimal	This data type is selected if the specified activity attribute contains decimal value. For example, 10.211, 100.50. The decimal precision can be defined under the decimal precision text field.

12. Select the operator to be applied on the activity attribute or process flow variable, from the *Operator* drop-down menu. This drop-down is populated with values based on the selection in the *Operand Type* field. The possible Operator values for various Operands are listed in the table below.

Table 16.20: Possible Operators for Operands

Operand	Possible Operators
Number	=, !=, <, >, <=, >=
Decimal	=, !=, <, >, <=, >=
Text	Equal, Not Equal, Equal Ignore Case and Not Equal Ignore Case

13. Select the value for the activity attribute or process flow variable that needs to be verified, from the *Value* drop-down menu. This drop-down is editable. The *Number* Operand type should be of data type *Long*. The *Decimal* Operand type should be of *Double* data type.
14. Click **Insert Variable** to insert the defined condition (comprising of activity attribute or process flow variable) into the *Expression* text area, displayed at the bottom of the screen.
15. Once the activity attributes or process flow variable is inserted, you can create another condition for building the expression. Select the condition to be used for building the expression, from the *Select Operator* drop-down menu. This drop-down is populated with values of *&& (AND)* and || *(OR)*.
16. Click **Insert Operator** to insert the selected operator into the *Expression* text area.
17. Both the conditions are displayed in the *Expression* text area. You can edit this expression for evaluation based on certain rules. These rules are outlined in the table below.

Table 16.21: Rules for Evaluation

Object	Rule
Activity Attribute/Process Flow Variable	It is to be displayed between $$. For example, Activity Attribute will be displayed as $$ Service. Activity. *Activity Attribute* $$ Process Flow variable will be displayed as $$ *var1* $$
Operand Text	▪ All values are enclosed within double quotes. For example, (“text”).

	▪ Values having '\' are replaced by '\\'. ▪ Values having double quotes (") are replaced by single quotes (').
Operator Equal	The condition is replaced as ($$var$$.equals ("text")).
Operator Not Equal	The condition is replaced as! ($$var$$.equals ("text")).
Operator Equal Ignore Case	The condition is replaced as ($$var$$.equalsIgnoreCase ("text")).
Operator Not Equal Ignore Case	The condition is replaced as! ($$var$$.equalsIgnoreCase ("text")).

If an invalid expression is entered in the Expression text area, the transaction will abort.

The information is displayed in the Rule Information screen as in Figure 16.69.

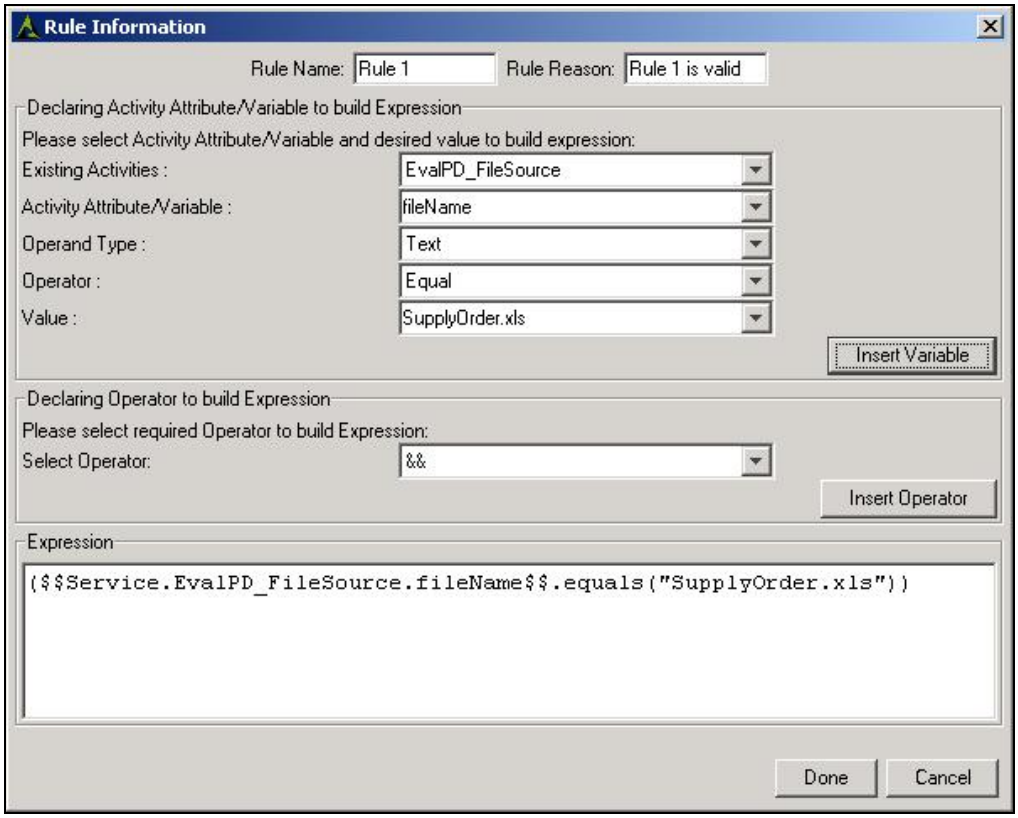


Figure 16.69: New Rule Information

18. Click **Done** button. This returns the control to the Condition Wizard screen with the newly added rule. Similarly, you can add more rules. They are displayed in the Condition Wizard screen (see Figure 16.70).

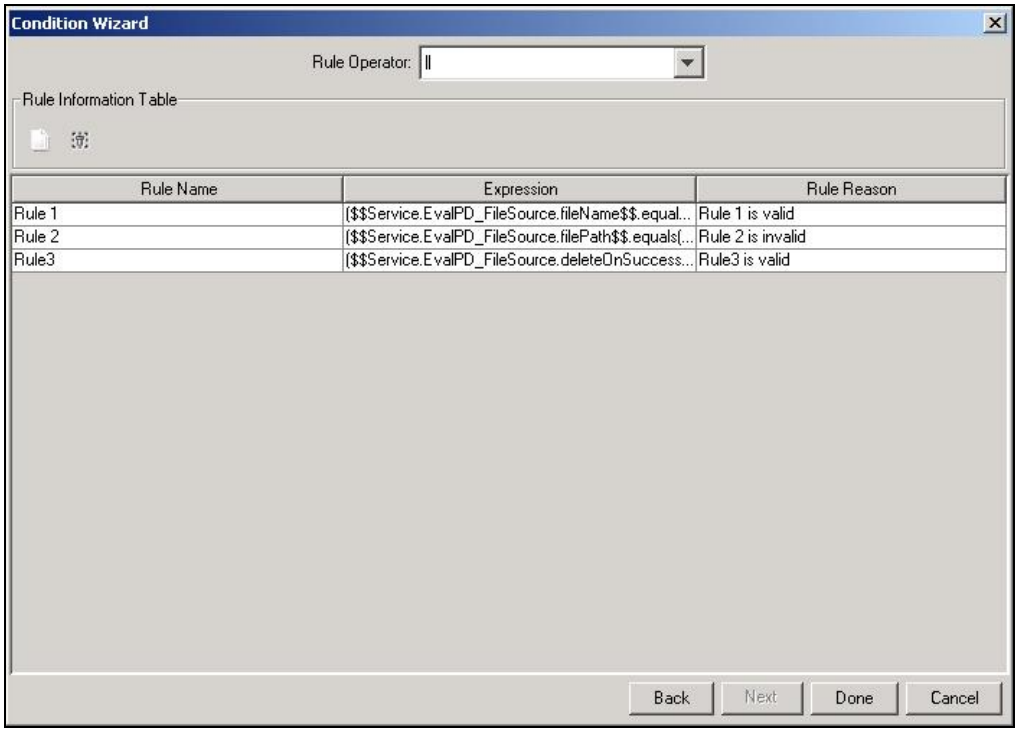


Figure 16.70: Added Rules

	You can edit the *Rule Name* and *Rule Reason* from this screen.

19. Select the operator on the basis of which you want to evaluate these rules, from the *Rule Operator* drop-down list. The various rule operators are outlined in the table below.

Table 16.22: Rule Operators

Rule Operator	Description
||	This evaluates the rules based on the OR operator. If any of the listed rules is true, then the decision value in the process flow will be executed as true.
&&	This evaluates the rules based on the AND operator. If all the listed rules are true, then only the decision value in the process flow will be executed as true. If any of the listed rules is false, then the decision value will be executed as false.

20. You can evaluate the rules using one operator at a time. For example, if you select ||, then all rules will be evaluated on the basis of the OR operator. You cannot evaluate

two rules (for example, Rule 1 and Rule 2) based on OR operator and two rules (for example Rule 2 and Rule3) based on the AND operator.

21. Click **Done**. This closes the Condition Wizard screen and returns to the Graph Canvas.

Once you execute the process flow, all the listed rules will be evaluated in top to bottom sequential order.

If they are evaluated using the *OR (||)* rule operator, then if any of the listed rules is true, then the decision value will be true and the process flow will be executed in the True path. The reason for all rules that are evaluated as True, will be added as comma separated values and set as a single value in the context variable *ruleReason* in the format *ruleName (ruleReason)*.

If the rules are evaluated using the *AND (&&)* rule operator, then only if all the listed rules are true, then the decision value will be true and the process flow will be executed in the True path. Even if one rule is evaluated to false, then the decision value will be false and process flow will execute in the False path. The reason for all rules that are evaluated as True, will be added as comma separated values and set as a single value in the context variable *ruleReason* in the format *ruleName (ruleReason)*. The rules that are evaluated as False, will appear as an empty value in the *ruleReason* context variable.

DEFINING SEQUENCE FLOW ORDERING

When more than one sequence flow is attached with an activity, the transition that was dragged first will be executed before the transition dragged later. You can specify the ordering of the execution of the transitions.

Steps to define sequence flow ordering

1. Right-click the activity in the Graph Canvas to which more that one sequence flow is attached and select Sequence Flow Ordering (see Figure 16.71).

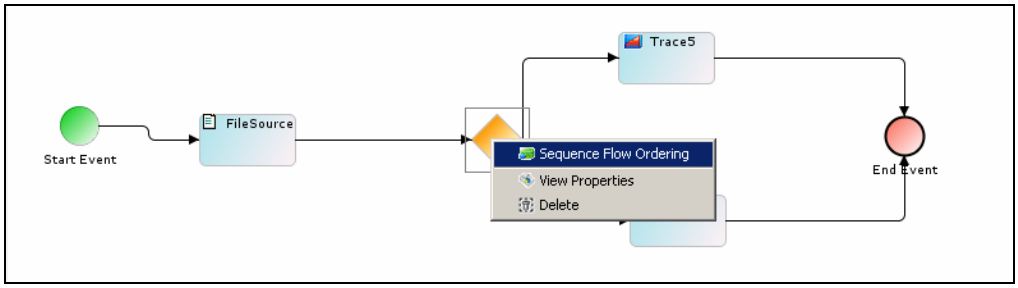


Figure 16.71: Define Sequence Flow Ordering

2. The Sequence Flow Ordering screen is displayed (see Figure 16.72).

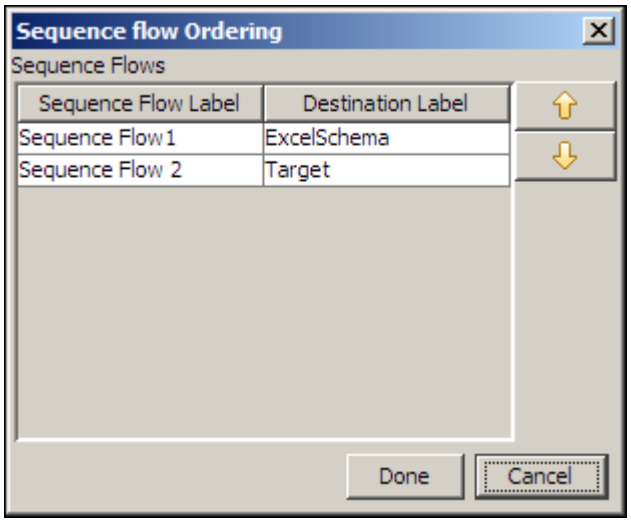


Figure 16.72: Sequence Flow Ordering

3. Select the required activity and move it using **Up** ( ) and **Down** ( ) arrow buttons to define the sequence.
4. Click **Done** button to return to the Graph Canvas.

CREATING MULTIPLE STREAMS

Stream represents the flow of data in a process flow. Multiple streams are used when the output of an activity needs to be sent to two or more activities in a process flow.

Steps to create multiple streams for an activity

1. Right-click the activity in the Graph Canvas from which multiple streams are to be generated and then select **Multiple Streams**. The Multiple Stream Dialog screen is displayed (see Figure 16.73).

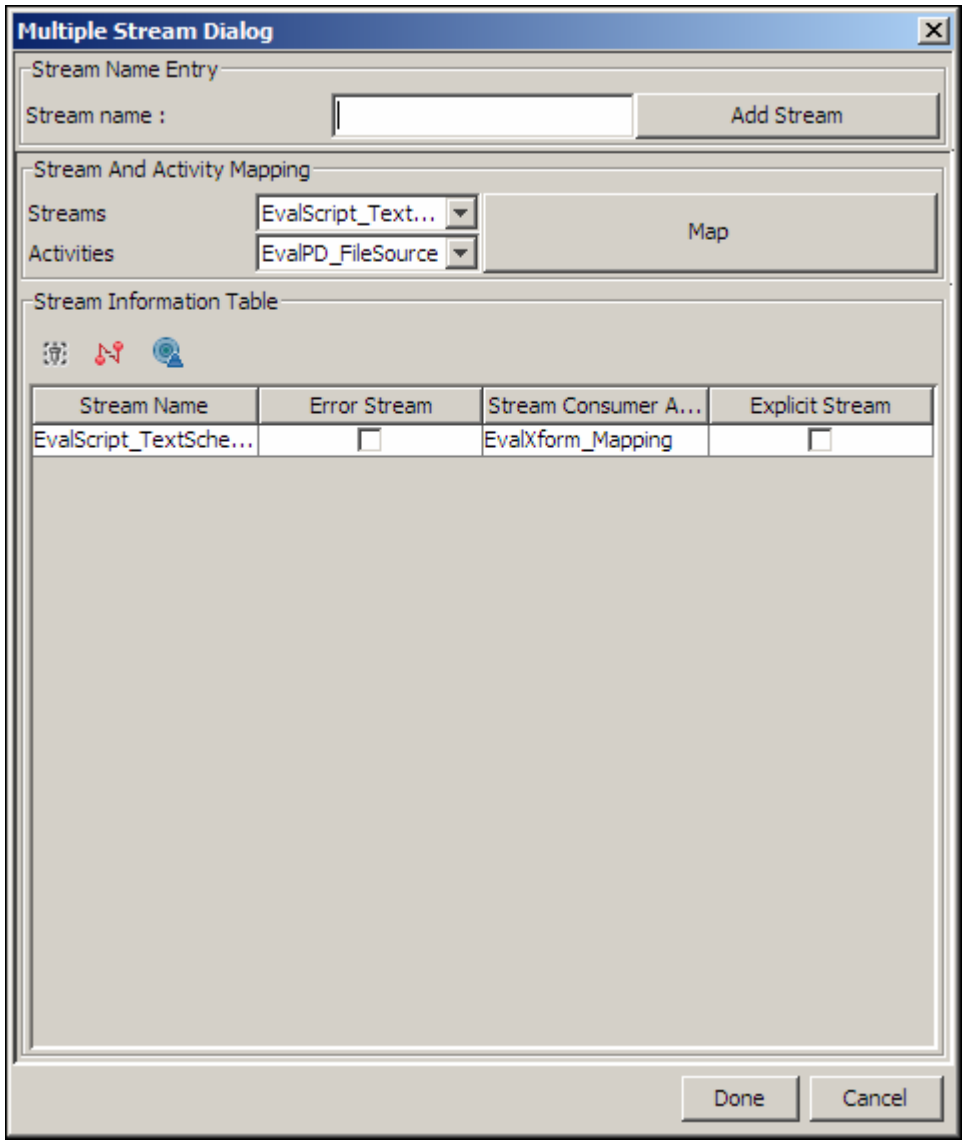


Figure 16.73: Multiple Stream Dialog Box

2. In the Stream Information Table, the existing stream is displayed. If you want to delete the existing stream, click on the stream to select it and then click **Delete Stream** ( ) button. The selected stream is deleted.

3. To create a new stream, enter the name for the Stream in the *Stream Name* field and then click **Add Stream** button. The name of the added Stream is displayed in the *Streams* drop-down list (see Figure 16.74).

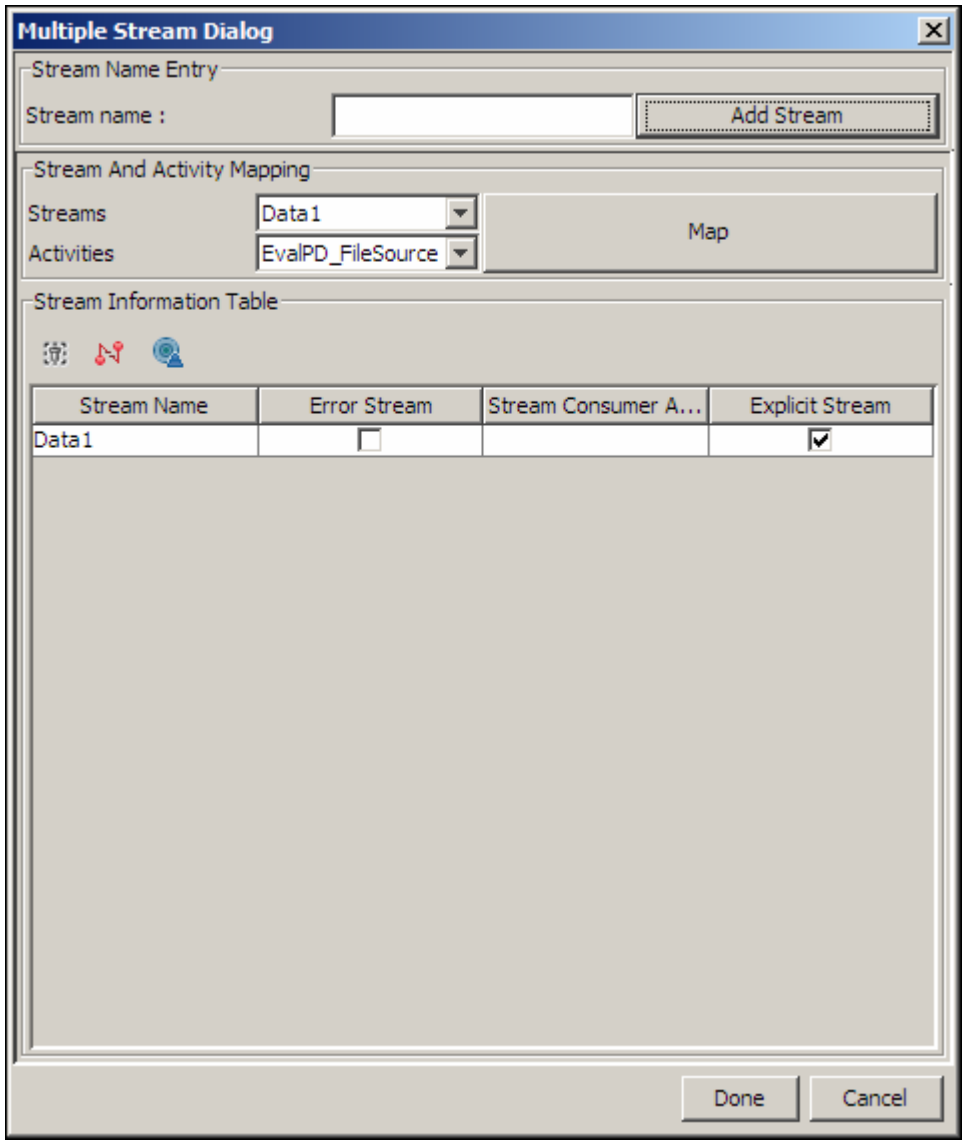


Figure 16.74: New Stream Added

4. The Activity which will consume the stream currently selected in the *Streams* drop-down list is already displayed, however to change it, select the activity from the *Activities* drop-down list, and then click **Map** button. The mapped stream and the activity are displayed in the *Stream Information Table* (see Figure 16.75).

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

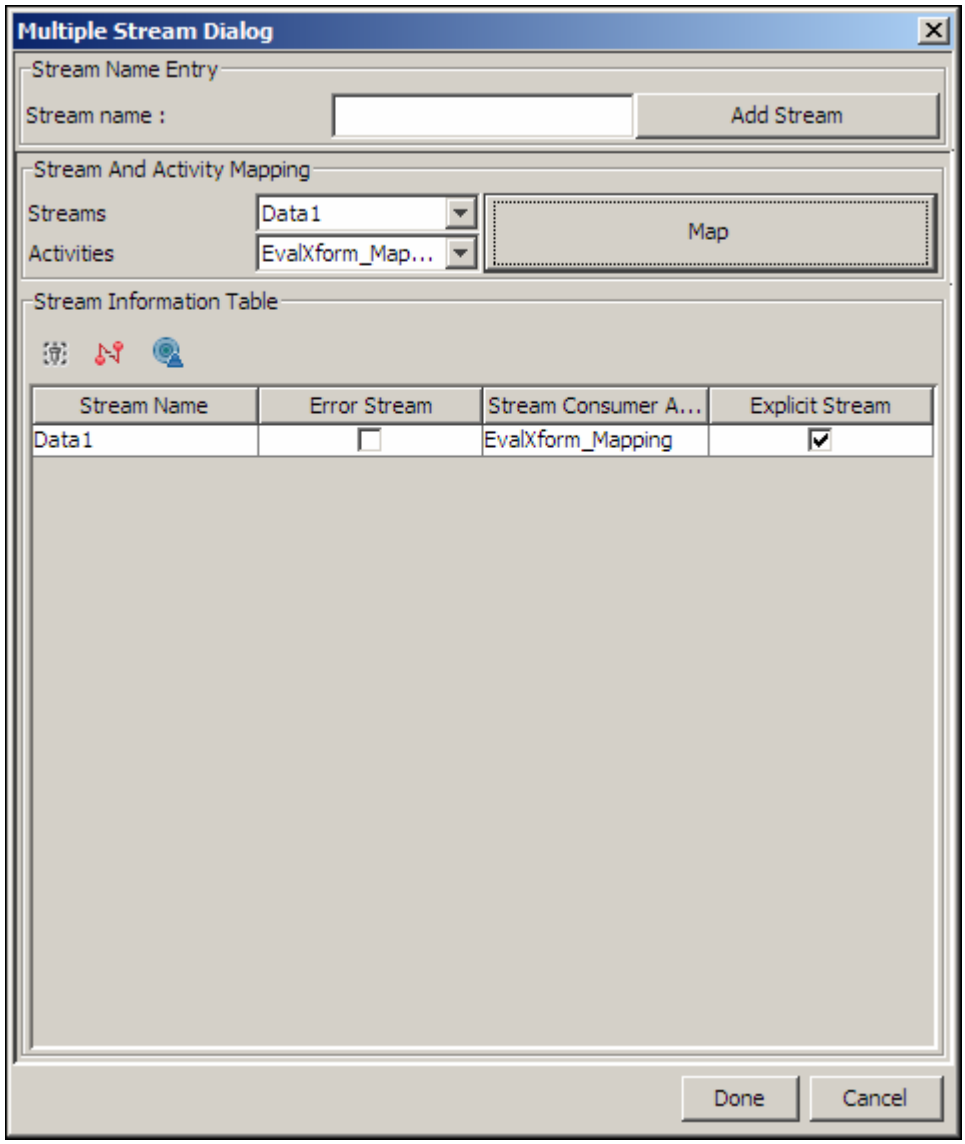


Figure 16.75: Stream Created

	If you are creating an error stream check the *Error Stream* checkbox.

6. Repeat steps 4 to 6 to create another stream.
7. Click **Done** button to return to the Graph Canvas. The created streams are displayed in the Graph canvas by data objects Artifact (see Figure 16.76).

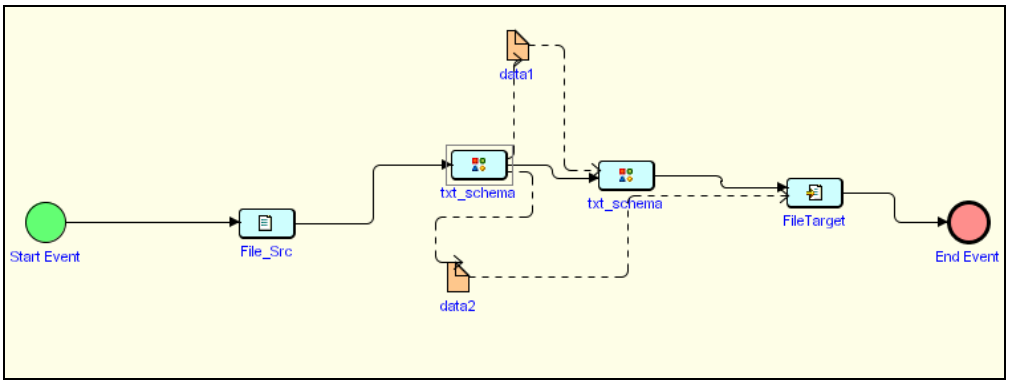


Figure 16.76: Showing Multiple Streams in Process Flow

	The Artifacts are only to show the flow of streams therefore it is necessary to add the activities with appropriate flows. If user deletes a stream from the Multiple Stream Dialog box, then the corresponding Artifacts are also deleted. If an Artifact representing a stream is deleted then the stream is also deleted. Process Designer asks user if he/she wants to delete the underlying stream. To hide the Artifacts, click **View** in the menu bar and uncheck the *Show Artifacts and Associations* option.

The Multiple Stream feature can be used in three scenarios:

- Creating more than one Stream

When data from one stream is sent to more than one activity, you can create more than one stream. However, more than one stream can be created only with the selected activities. These activities are Mapping Transformation, Scripted Service, Repeater Service, XML Validator and Decoder.

	A Mapping activity does not always generate multiple streams. It is based on the schemas used in the mapping activity.

- Creating Error Stream

If a source or target schema is not compliant with the corresponding source or target data, then the process flow execution will generate error records. Sometimes these error records are useful for users and user may want to store them for debugging. In such a case, error stream is created along with data stream. This error stream can be mapped to some other activity for further processing. For example, user may want to store the error records in a file target. To specify a stream as an error stream, check the *Error Stream* checkbox in the Multiple Stream Dialog Box. Error stream can be generated for Schema, Database source, Database target, Advanced Database target and XML Validator only.

- Creating Default Stream

Sometimes an action, for example the Delay action, is used between two activities. In such type of situation data from the first activity does not pass to another activity because the action Delay does not consume data from the first activity and hence cannot pass it to another. To avoid this condition default stream is used.

Steps to create a default stream

1. Right click the first activity and select **Multiple Stream**. The Multiple Stream dialog box appears (refer to Figure 16.73).
2. Delete the existing stream.
3. Select another activity from the *Activities* drop-down list.
4. Click **Default Stream** () button and then click **Map** button (see Figure 16.73).
5. Ensure that the *Explicit Stream* checkbox is checked.

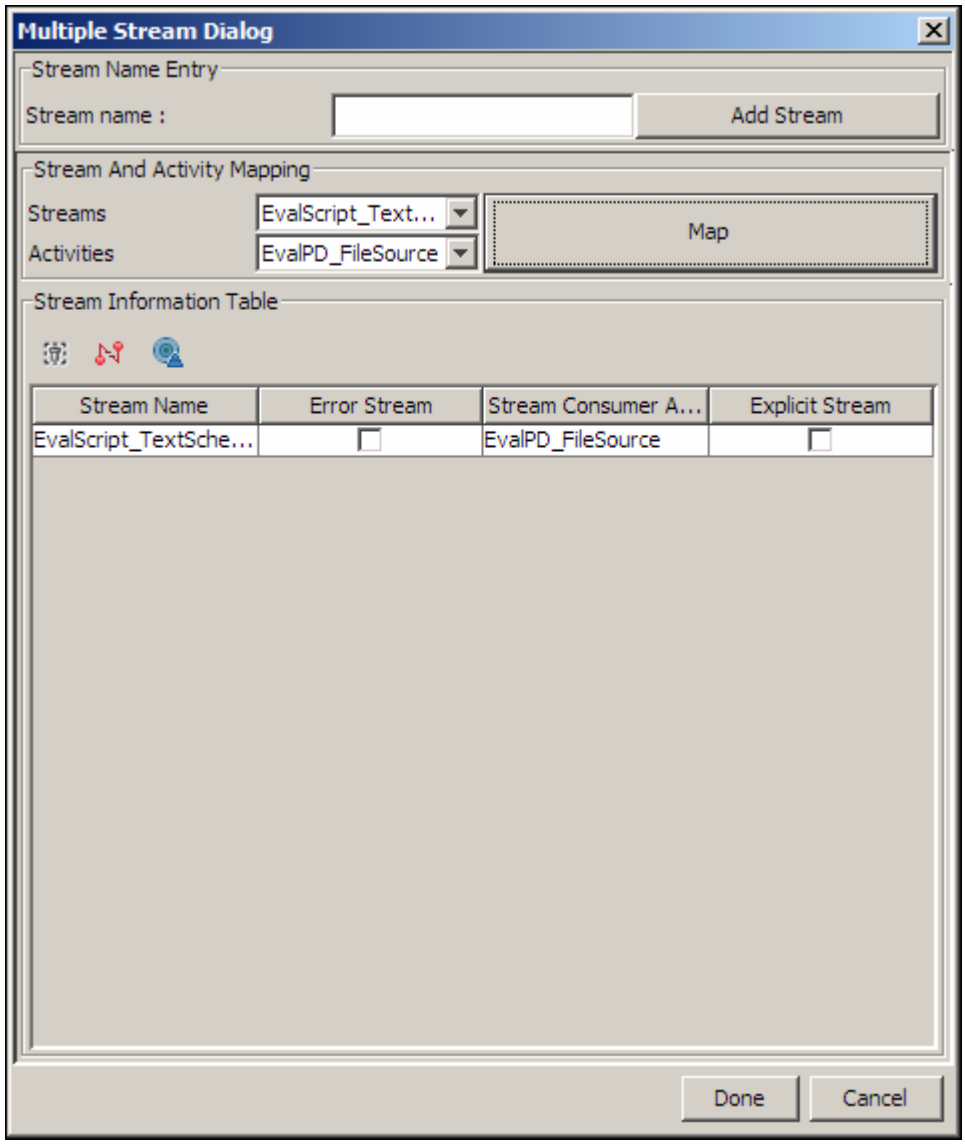


Figure 16.77: Creating Default Stream

6. Click **Done** button to save the stream and return to Process Designer screen.

	If you map a stream from a source activity to multiple activities (for example, Schema or Target activities) using the Multiple Stream option, only one target activity can get the stream from the source activity. The other target activities do not get the stream from the source activity and thus get aborted. This in turn aborts the process flow. It will only work if a gateway element is used in a process flow. This element will include a condition, on the basis of which one target activity can be selected at runtime. Another way is to use a Repeater Node to pass the stream to more than one activity. For details on using a Repeater Node, refer to the **Using Repeater Node** section.

USING STREAM SELECTOR

When more than one input stream is connected to an activity, Stream Selector is used to specify the input stream to be consumed by the activity. To understand the use of Stream Selector consider the following scenario (see Figure 16.78).

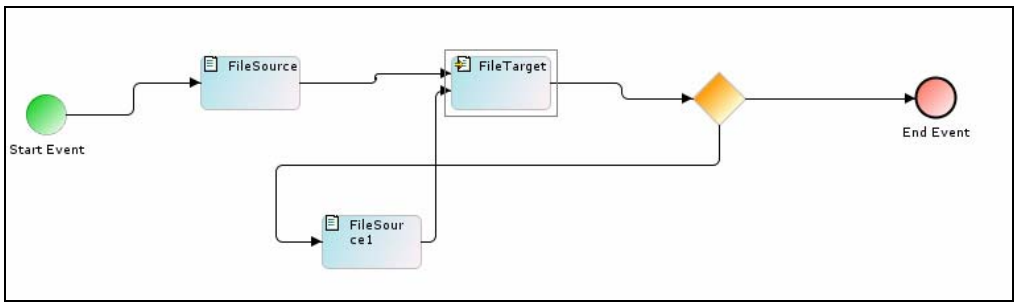


Figure 16.78: Scenario

In this process flow, you can see that two input streams are passed to the File Target. The desirable execution sequence is that first *File Source* will be executed then *File Target* and there after the *Gateway*. Now, if the condition specified at the *Gateway* is satisfied, the process flow will be finished. If the condition specified at *Gateway* is not satisfied, the control will be passed to the *File Source1* and further to *File Target*. Practically, when this process flow will be executed, the *File Target* activity will get confused whether to take input from *File Source* or from *File Source1*. Thus, this process flow will fail.

To avoid such a situation, **Stream Selector** is used. Streams from *File Source* and *File Source1* are passed to Stream Selector and then from Stream Selector to the *File Target*. The use of Stream Selector is displayed in Figure 16.79.

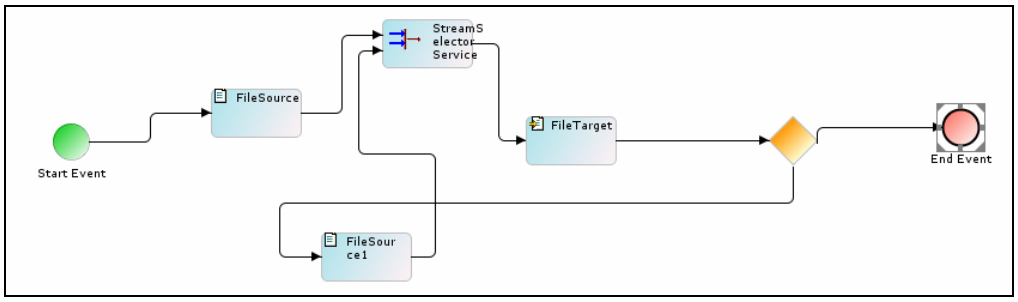


Figure 16.79: Stream Selector

Stream Selector takes input from the activity, which has just executed. If *File Source* has executed before the execution of stream selector, it will take input from *File Source*. If *File Source1* has just executed, it will take input from *File Source1*.

Steps to use a Stream Selector

1. Click tree structure in the **Repository View** panel. Expand the **Activities** list of the Adeptia Server and select **Selector**. A list of selector activities is displayed.

2. Click **Stream Selector Service** and drag it to the Graph Canvas. The Stream Selector Service node is displayed in the Graph Canvas area (see Figure 16.80).

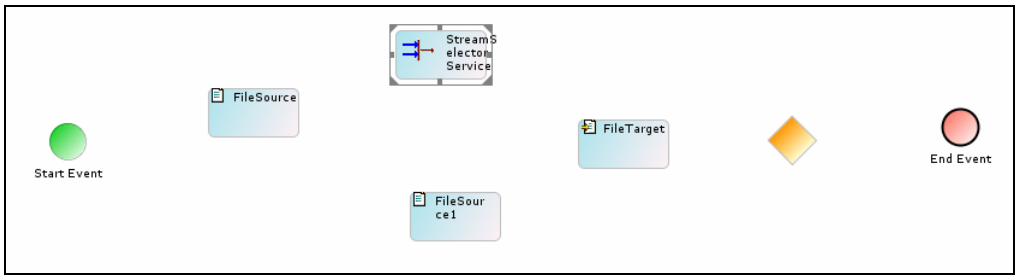


Figure 16.80: Drag Stream Selector to Graph Canvas

3. Connect the required activities with the Stream Selector Service node (see Figure 16.81).

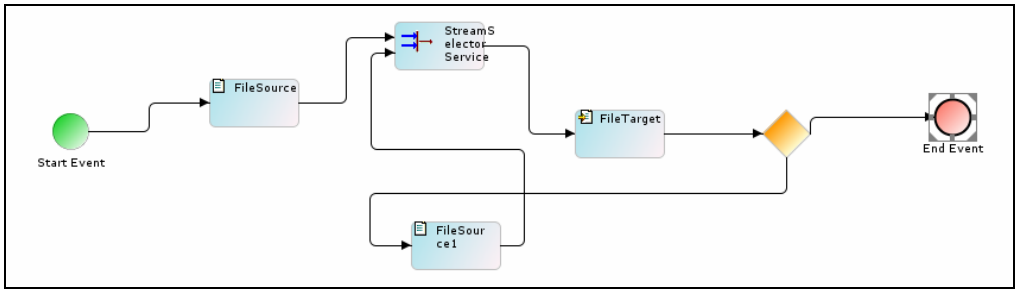


Figure 16.81: Connect Activities with Stream Selector Node

USING REPEATER NODE

Repeater node is used when output of one activity is sent to more than one activity. For example, data from a file source is sent to two file targets.

Steps to use a Repeater Node

1. Click tree structure in the **Repository View** panel. Expand the **Activities** list of the Adeptia Server and select the **Repeater Node**. A list of repeator node activities is displayed.
2. Select the **Repeater Service** under Repeater Node and drag it to the Graph Canvas. A Repeater Service node is displayed in the graph canvas (see Figure 16.82).

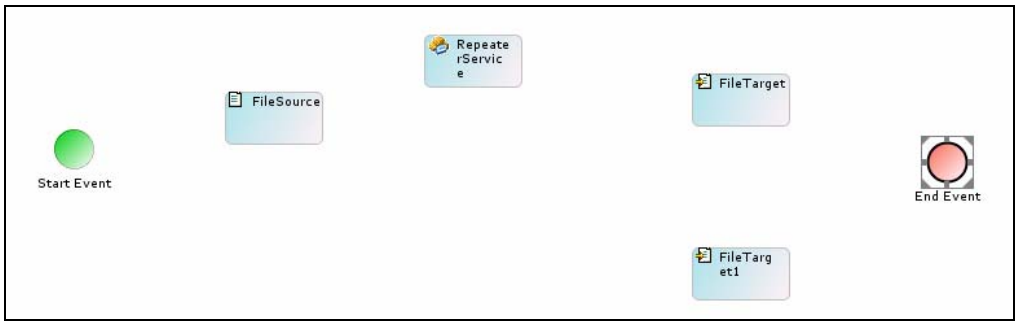


Figure 16.82: Repeater Service Node

3. Connect all activities with appropriate control flows in the Graph Canvas (see Figure 16.83).

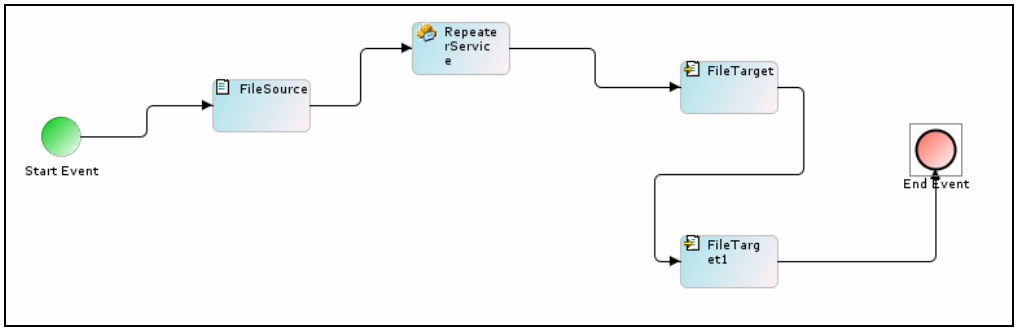


Figure 16.83: Connect Activities

	The Figure 16.83 displayed above does not illustrate the data flow. It is showing the sequence in which activities will be executed. This should be noted that File Target and File Target1 activities must not be connected with Repeater Service in parallel to each other. Now to specify data flow, multiple streams must be created with Repeater Service.

4. To create multiple streams, right click **Repeater Service** and select **Multiple Stream**. The Multiple Stream Dialog Box is displayed.
5. Enter number of streams in the *Enter Stream Count* field and click **Add Stream** button. Since we have two outputs from repeater node, enter 2 in the *Enter Stream Count* field.
6. Select 1 from the *Stream* drop-down list and select the first activity, which will consume the stream from the *Repeater Node*, from the *Activities* drop-down list.
7. Click **Map** button. The stream created is displayed in the *Stream Information Table*.
8. To create second stream, select 2 from the *Stream* drop-down list and select another activity, which will consume another stream from *Repeater Node* from *Activities* drop-down list.
9. Click **Map** button. The second stream is displayed in the *Stream Information Table* (see Figure 16.84).

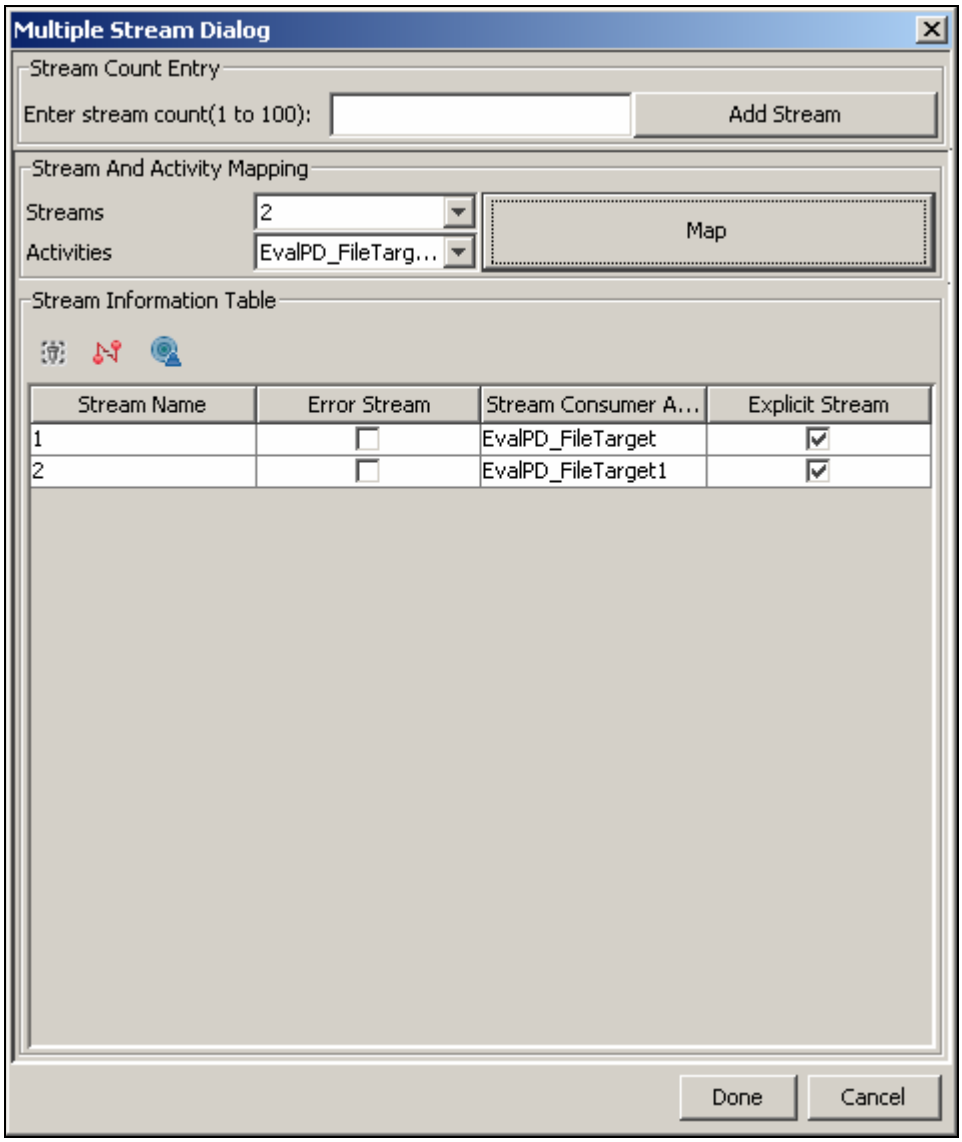


Figure 16.84: Create Second Stream

10. Click **Done** button to save the streams created and return to the Graph Canvas. The created streams are displayed in the Graph Canvas by Artifact (see Figure 16.85).

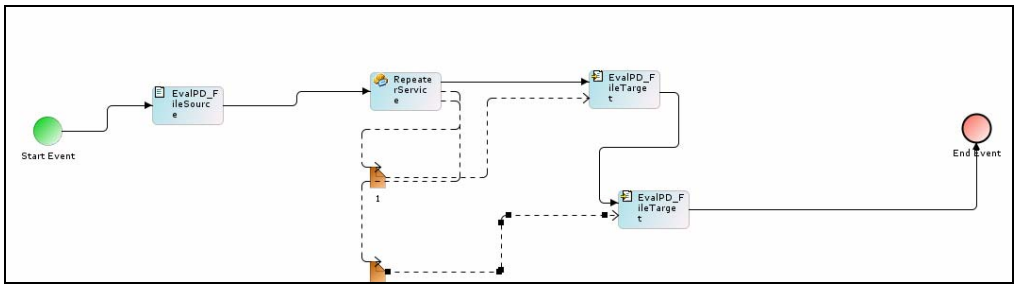


Figure 16.85: Repeater Node with Multiple Stream

USING ERROR INTERMEDIATE EVENT

Error Intermediate Event is used to redirect Process Flow execution to an alternate path in case of failure of any activity during process flow execution. To understand the use of Error Intermediate Event consider the following scenario:

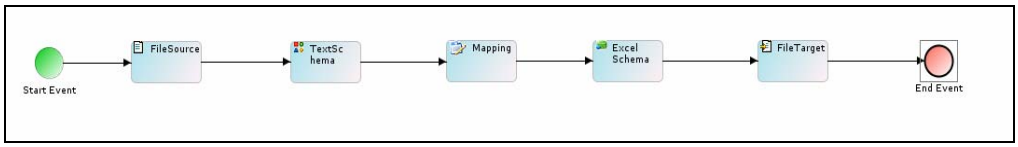


Figure 16.86: Scenario

In this process flow, data from a text file is converted to an excel file using schema and mapping activities. Now if the Mapping activity is critical for your business, you may want be notified, if mapping activity fails during process flow execution.

To handle this situation you can attach **Error Intermediate Event** with mapping activity so that, in case mapping activity fails, a notification activity is executed and sends an email to the specified email address. Use of Error Intermediate Event is displayed in the Figure 16.87

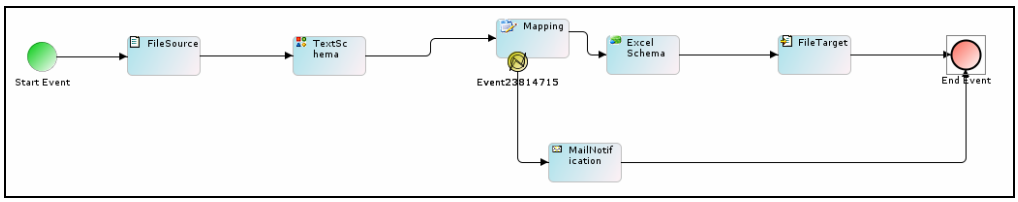


Figure 16.87: Use Error Intermediate Event

Steps to use Error Intermediate Event

1. Right-click the activity with which you want to attach the intermediate event and select **Add Intermediate Event** option (see Figure 16.88).

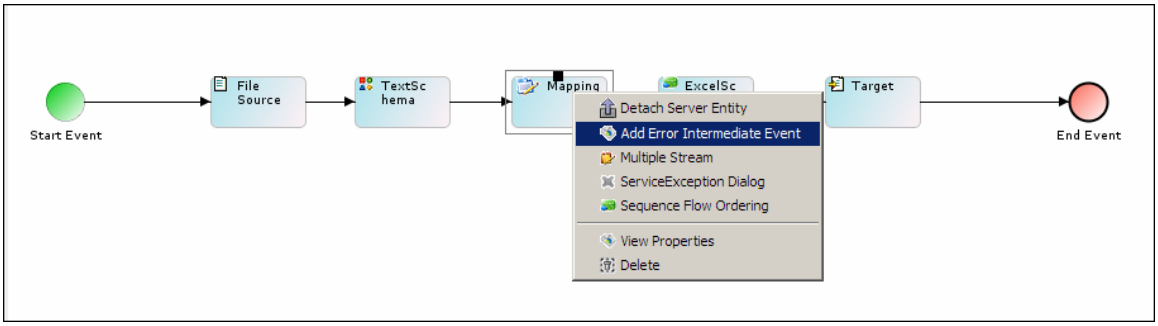


Figure 16.88: Select Add Intermediate Event

2. This attaches the Intermediate Event to the selected activity (see Figure 16.89).

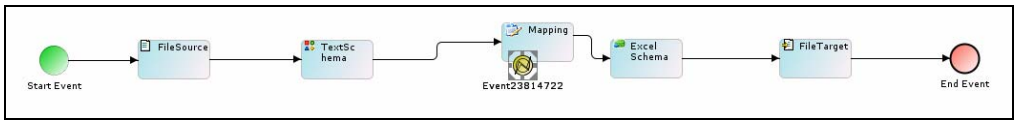


Figure 16.89: Error Immediate Event Attached

3. Drag another activity, which needs to be executed in case of failure of Mapping activity, to the Graph Canvas Area (see Figure 16.90).

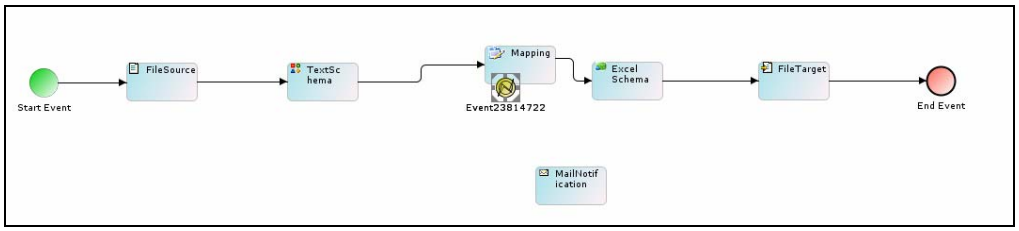


Figure 16.90: Drag Another Activity

4. Connect the Error Intermediate Event to Mail Notification activity and then Mail Notification activity to End Event (see Figure 16.91).

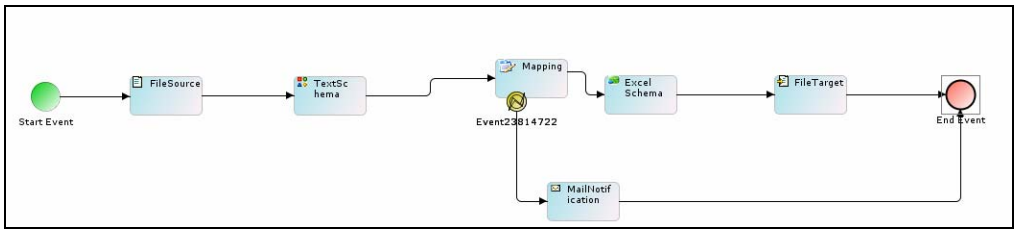


Figure 16.91: Connect Activities

	While connecting Error Intermediate Event with Mail Notification activity, please ensure that you are connecting Error Event with Mail Notification not the Mapping activity with Mail Notification.

CREATING EXCEPTION HANDLER SCRIPT

Exception Handler scripts are basically Java code, which is invoked, if any exception or error occurred during execution of a process flow. There are three types of Exception Handler Scripts:

- Service Exception
- Process Flow Exception
- Invalid Data Exception

These Exception Handler Scripts can be created at a global level (for all the activities in the Process Flow) or at the activity level (for a specific activity). If an exception or errors during the execution of a process flow, the activity level exception handler script is invoked first. If the activity level exception handler script is not created for that activity, then only the global level exception handler script is invoked.

These Exception Handlers are invoked at different stages of the process flow. These are depicted in the table below.

Table 16.23: Exception Handlers in a Process Flow

Service Exception	This exception handler script is invoked when any error related service (activity) occurs. For example: File not found, Stream closed etc.
Process Flow Exception	This exception handler script is invoked when any error occurs at Process Flow level. The Exception can only be declared at global level. For example: JMX not found, Repository not found etc.
Invalid Data Exception	This exception handler script is invoked when any error related to processing of data occurs. For example: Incorrect record format etc.

Creating Global Exception Handler Script

Steps to create a Process Flow (Global) Exception Handler Script

1. Click **On Exception Scripts** tab ( ) in the Bottom Pane. The Create Exception Handler screen is displayed (see Figure 16.92).

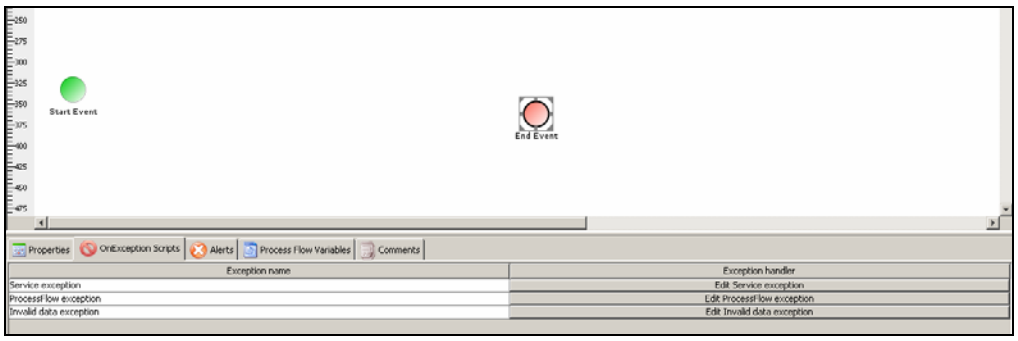


Figure 16.92: Creating Exception Handler Script

2. Click **Edit Service Exception**. The Service Exception dialog box is displayed (see Figure 16.93).

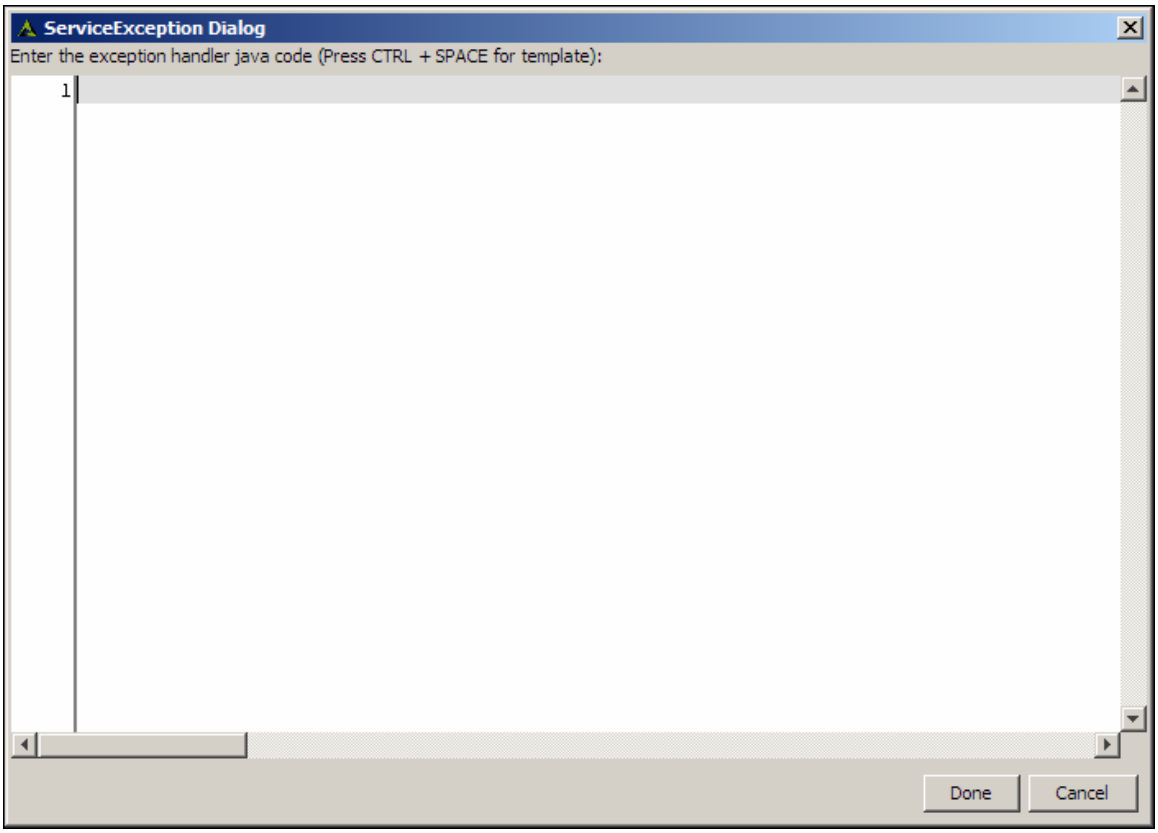


Figure 16.93: Service Exception Dialog Box

	You can use **<CTRL>+<Space Bar>** to view pre-defined template of Java Code, which can be used in creating Java Condition. You can select any of them and edit it according to your requirement. If you want to create the process Flow Exception Handler or Invalid Data Handler script, click **Edit Process Flow Exception** or **Edit Invalid Data Exception** buttons respectively.

3. Enter the Java code in the *Service Exception* dialog box and click **Done** button.

Creating Activity Exception Handler Script

Steps to create an Activity Exception Handler Script

1. Right-click the activity in the Graph Canvas and select **Service Exception Dialog**. The Service Exception Dialog Box is displayed (see Figure 16.93).
2. Enter the Java code in the *Service Exception* field and click **Done** button.

	You can use **<CTRL>+<Space Bar>** to view examples of Java Condition. **<CTRL>+<Space Bar>** show lists of example. You can select any of them and edit it according to your requirement. If you want to create Invalid Data Handler script, right-click the activity and select **Invalid Data Exception Dialog**.

USING COMPRESSION / DECOMPRESSION

Compression is used to compress any file into ZIP, RAR or JAR file. Similarly, Decompression is used to extract file from a ZIP, RAR or ZAR file. This feature supports Compression or Decompression of only one file.

Steps to use Compression / Decompression

1. Drag the required file source activity in the Graph Canvas area.
2. Click tree structure in the **Repository View** panel. Expand the **Activities** list of the Adeptia Server and select **Compression Service**. A list of compression service activities is displayed.
3. Select **Compression** and drag it to the Graph Canvas area (see Figure 16.94).

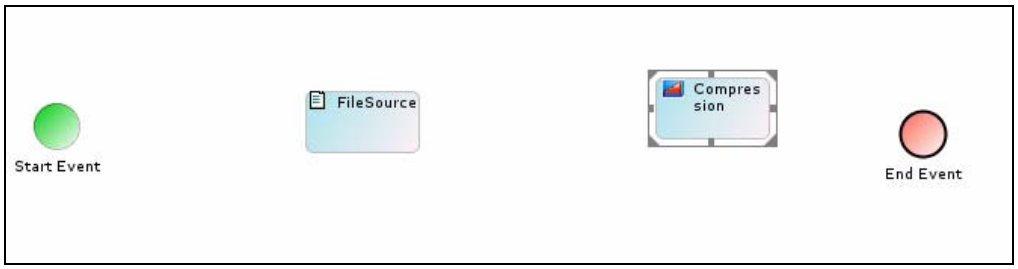


Figure 16.94: Drag Compression Activity

4. Drag the required target activity to the Graph Canvas area.
5. Connect all activities in Graph Canvas with **Unconditional Control flow** as displayed in Figure 16.95.

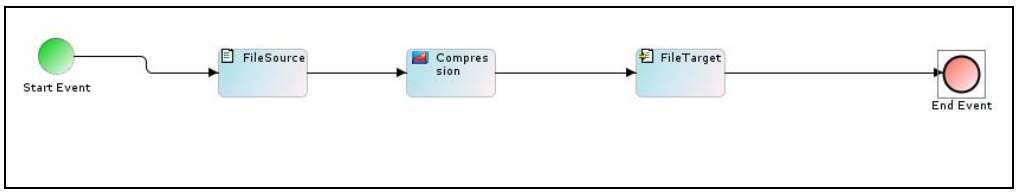


Figure 16.95: Connect Activities

6. Right-click the **Compression** activity and select **View Properties**. Properties of the Compression activity are displayed in the Properties Panel (see Figure 16.96).

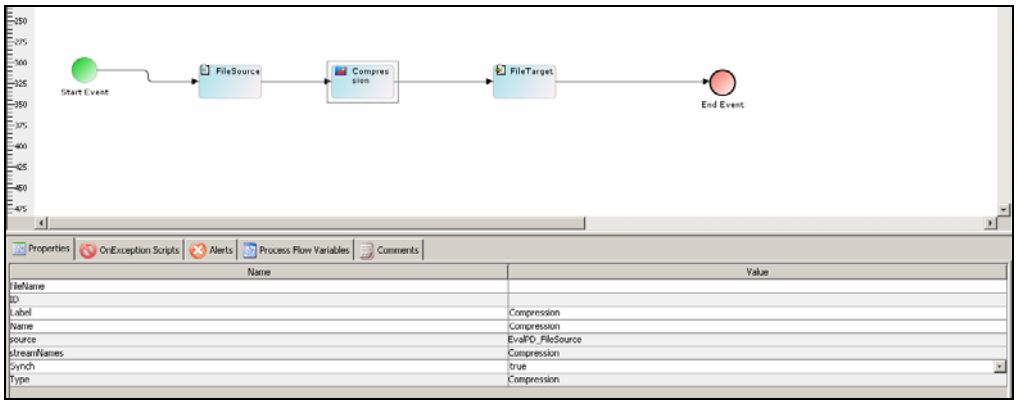


Figure 16.96: Compression Activity Properties

7. Enter the name of the file, which will be within the compressed file, in the *Filename* field.

	Name specified in the File Target activity will be the name of the compressed target file.

Similarly you can use Decompression feature.

USING CUSTOM REPORT IN PROCESS FLOW

A custom report is used to send generated report (in PDF or HTML formats) to a file based target (in PDF or HTML formats).

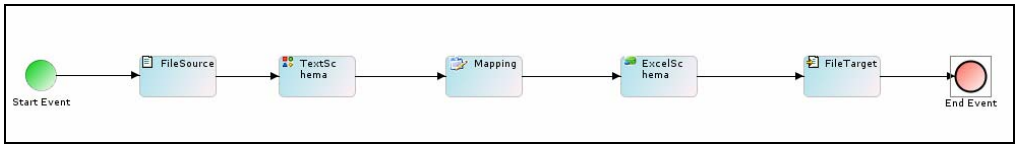


Figure 16.97: Scenario

In this process flow, data from a text file is converted to an excel file using the schema and mapping activities. The data generated in the excel file is converted to PDF/HTML format using a custom report and sent to a file target.

Steps to use a Custom Report

1. Click tree structure in the **Repository View** panel. Expand the **Activities** list of the Adeptia Server and select **Reports**. A list of reports is displayed.
2. Select the **Custom Report** and expand it to view the custom reports. Select the desired custom report and drag it to the Graph Canvas. The Custom Report node is displayed in the graph canvas (see Figure 16.98).

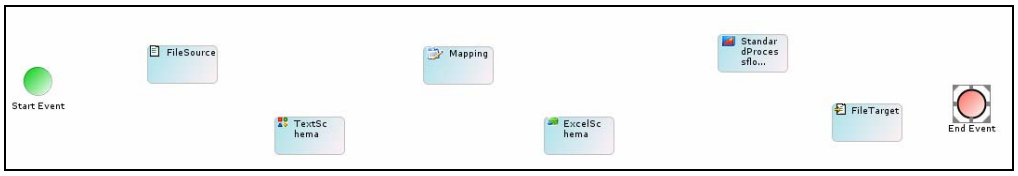


Figure 16.98: Custom Report Node

3. Connect all activities with appropriate control flows in the Graph Canvas as displayed below (see Figure 16.99).

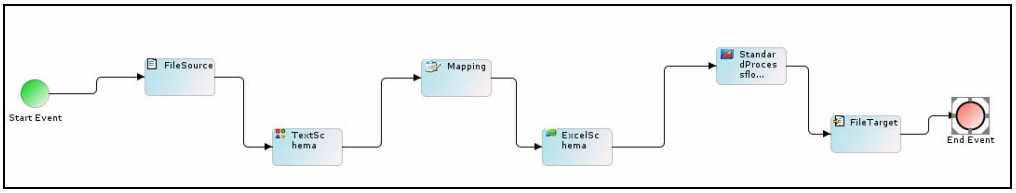


Figure 16.99: Connect Activities

4. Executing this process flow will send generated report to the file target.

ATTACHING END PROCESS (MAIL NOTIFICATION) TO PROCESS FLOW

Attaching End Process adds a Mail Notification activity at the end of the process flow. Mail Notification is used to send e-mail to appropriate users at the certain point in process flow.

Steps to Attach a Mail Notification at the end of Process Flow

1. In the Graph Canvas, right-click at the **End BPMN** element and then select the **View Properties**. The properties of the End BPMN element are displayed in the Properties Panel (see Figure 16.100).

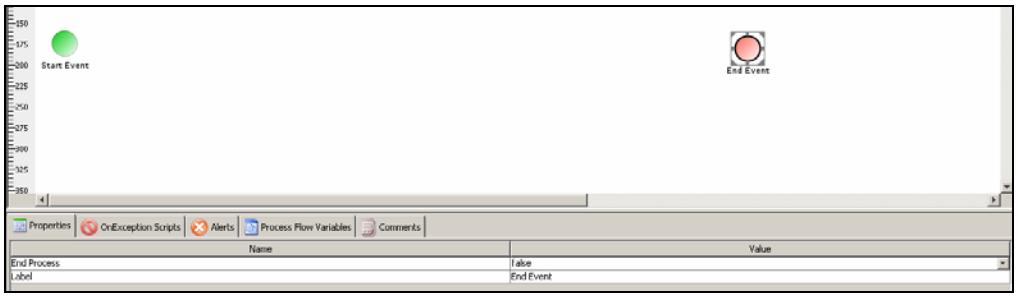


Figure 16.100: Attaching End Process

2. Click **End Process Value** column and select *True* from the drop-down list. The Select Notification screen is displayed with list of mail notification activities (see Figure 16.101).

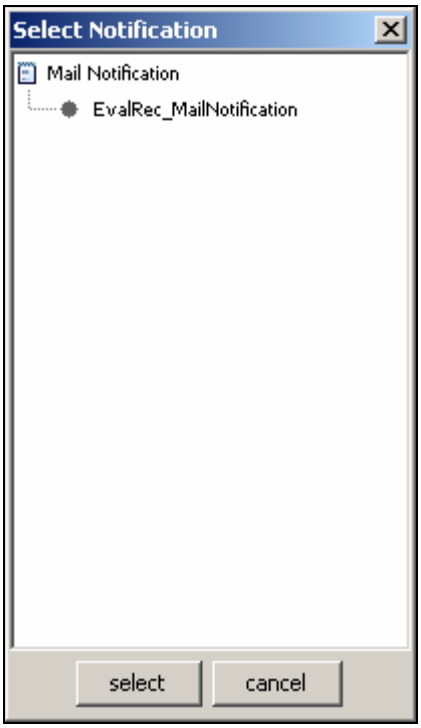


Figure 16.101: Selecting Mail Notification

3. Select the required mail notification activity and click **Select** button. The selected mail notification activity is attached at the end of the process flow.

	To know, how to create Mail Notification activity, refer to the **Creating Mail Notification activity** section.

GENERATING PDF FILE OF PROCESS FLOW

The process flow information can be saved into a PDF file for documentation purposes. There are three types of PDF files that can be generated from Process Designer.

- **Graph PDF:** Graph PDF only contains the screen shot of the Process Flow created in Graph Canvas area.
- **Summary PDF:** Summary PDF contains activity details of the Process Flow. It is implemented only for mapping and schema.
- **Entire Flow PDF:** Entire Flow PDF contains information regarding each and every activities of the Process Flow. List of information contained in the Entire Flow PDF is as below.

Process Flow Details

- Process Flow Name
- Process Flow ID
- Process Flow Description
- Creation Date
- Modification Date
- Debug Level
- Group Owner

Process Variable Details

- Key or Name of the Variable
- Initial Value
- Type (Global or Activity)
- Activity Name (If the Type is activity)
- Activity Label used in Process Designer
- Activity Type

Graph Canvas Entity

- Name
- Description
- User Owner
- Group Owner
- Creation Date
- Modification Date
- Permission
- Comment
- Label
- Inbound Connecting Objects
- Outbound Connecting Objects
- Activity Specific Information

For example, path of the source file in case of File Source activity, information of fields in case of Schema Activity or mapxmlfile in case of Mapping Activity.

Process Flow Graph

Steps to generate a PDF file of a Process Flow

1. In Process Designer, click **Generate PDF** () button. The PDF Generation dialog box is displayed (see Figure 16.102).

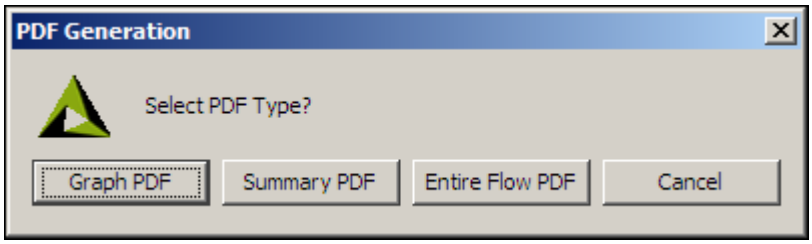


Figure 16.102: Select PDF Type

2. Select the type of PDF file to be generated. A *Graph PDF* includes all the rules applied on all activities in the process flow. A *Summary PDF* includes only the activity details, and is implemented only on mapping and schema. The *Entire Flow PDF* includes all details of the process flow. The Save box appears to select the path, where PDF file will be saved (see Figure 16.103).

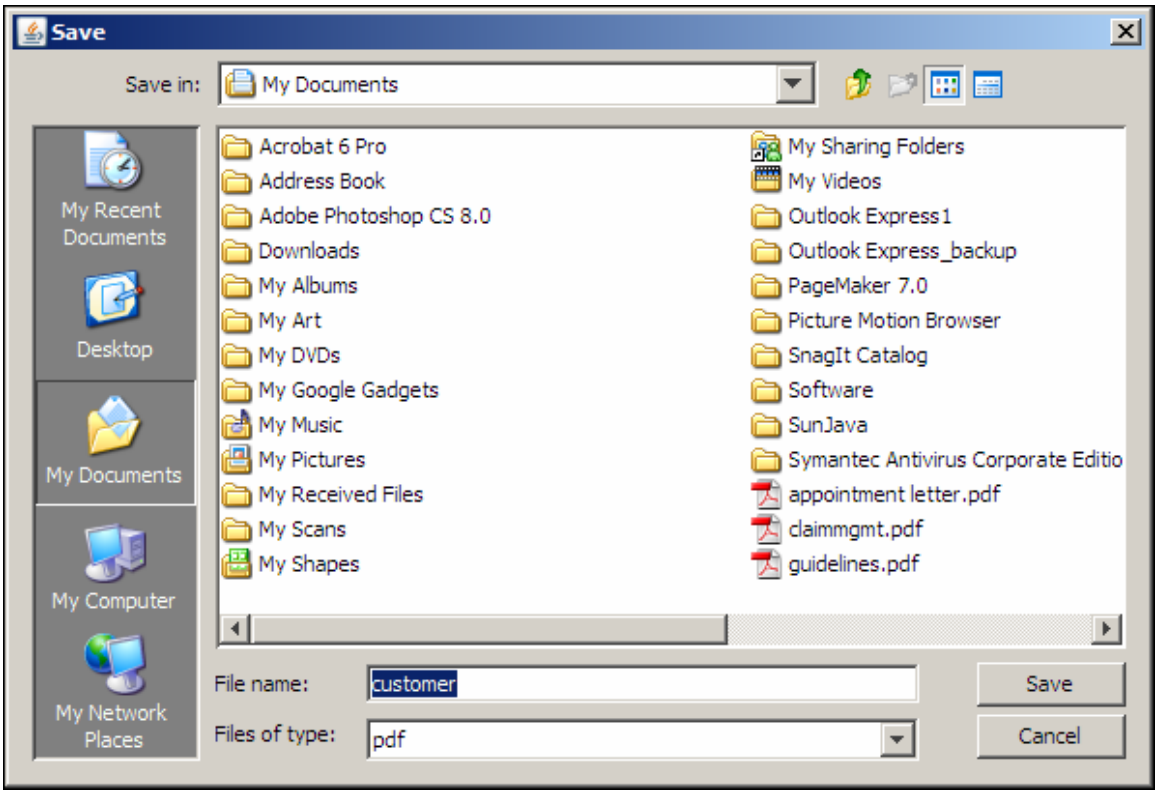


Figure 16.103: Save Box

3. Select the path and enter the name of the file in the *File Name* field.
4. Click **Save** button. The generated PDF file is saved in the specified folder.

WORKING WITH PROCESS DESIGNER IN OFFLINE AND ONLINE MODES

The Adeptia Server Process Designer allows user to work in offline mode. A user can design a process flow even if he/she is not connected with the Adeptia Server. At any moment user can

switch from Offline mode to Online and vice versa. The Online mode or Offline mode status is displayed in the bottom left side of the Process Designer window.

Steps to switch to the Offline mode

1. Click **Modes** menu and then click **Online** option. The checkmark displayed next to the Online mode is removed and you will be shifted to the Offline mode (see Figure 16.104).

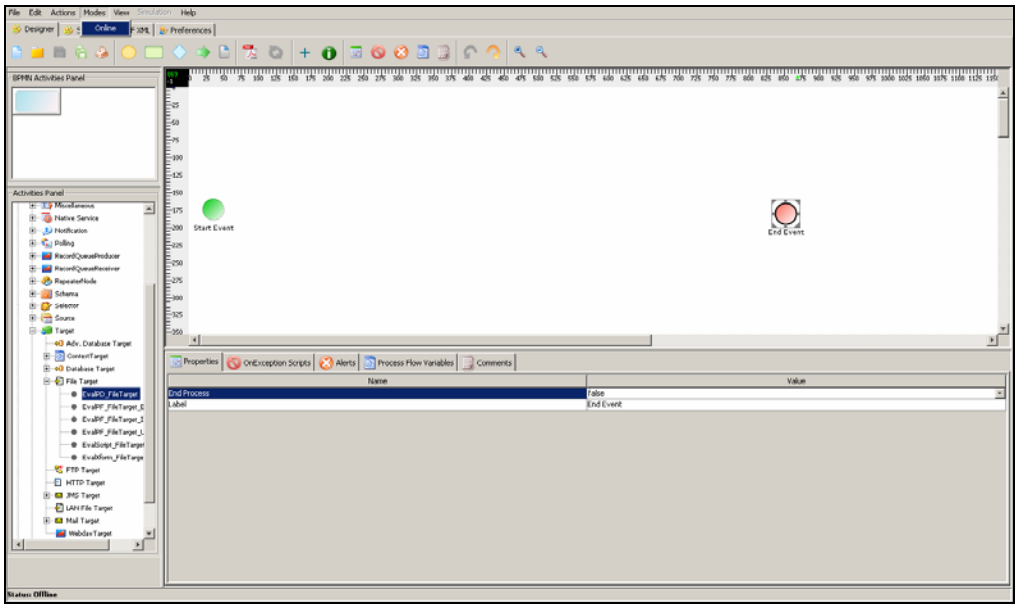


Figure 16.104: Changing Mode

	While working in the Offline mode, you cannot save the Process Flow on the Adeptia Server. You can save the Process Flow on your local hard disk. Users working in the Offline mode cannot see objects e.g. activities and process flows etc that other users create on Adeptia Server. To view those objects, shift to the Online mode. Click the Actions menu and then select Synchronize PD with Adeptia Server.

When switching to Online mode, you need to login again.

Steps to switch to the Online mode

1. Click **Modes** menu and then click **Online** option. The Process Designer Login screen is displayed (see Figure 16.105).

Figure 16.105: Process Designer Login

2. Enter the User ID and the password in their respective fields and click **Submit**. This activates the online mode for the Process Designer screen. If the user belongs to more than one group, then the Select Group screen is displayed (see Figure 16.106).

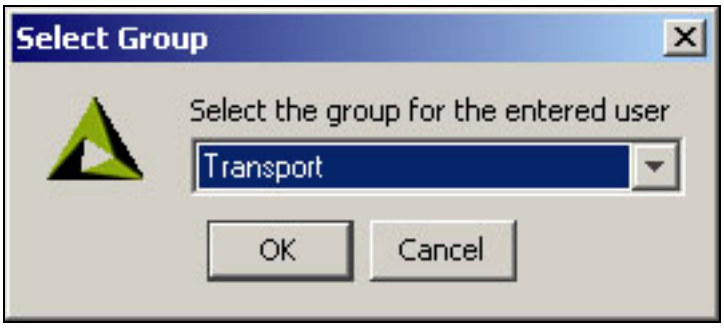


Figure 16.106: Select User Group

	The *Group* drop-down list is populated with only those groups of which the logged in user is a member.

3. Select the group from the *Group* drop-down list and click **OK**. This activates the online mode for the selected user.

EXECUTING A PROCESS FLOW

Once you have created a process flow, you can execute it, in order to process the data. You can execute a pocess flow in two ways:

- **Manually**
- **Automatically**

Executing a Process Flow Manually

Steps to execute a Process Flow manually

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. A The Manage Process Flow screen is displayed (refer to Figure 16.4).
3. Select the radio button adjacent to the required process flow and click **Execute** link. The process flow execution screen is displayed (see Figure 16.107).

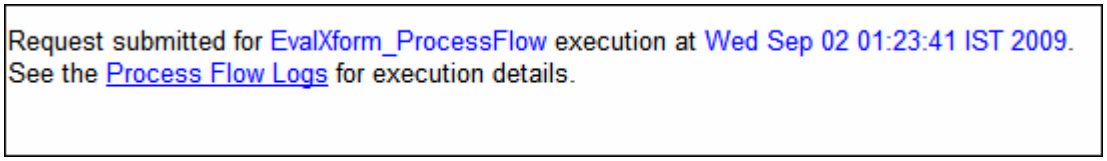


Figure 16.107: Process Flow Executed

4. To view the status of execution of the process flow, click **Process Flow Logs** link. The Process Flow Log screen is displayed (see Figure 16.108).

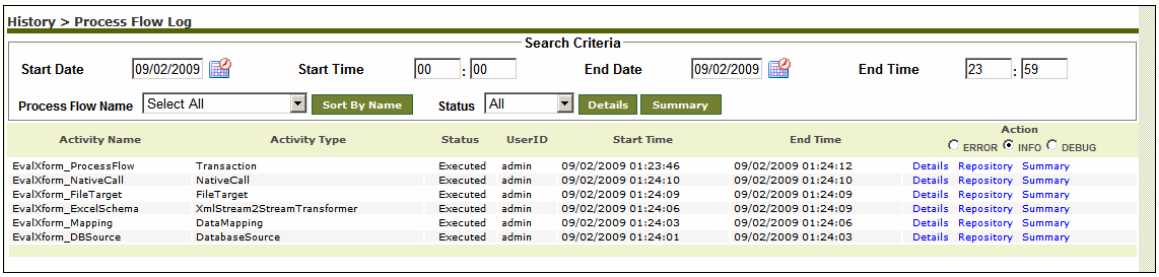


Figure 16.108: View Process Flow Log

To learn more about Process Flow Log refer to the *View Process Flow Logs* section in the *Business User Guide*.

Executing a Process Flow Automatically

You can execute a process flow automatically based on some events. For example, when you receive an email in your mailbox, the process flow can be executed. For this, you need to bind an event (for example, a mail event) with the process flow. You can either bind an existing event or create a new event and then bind it with a process flow.

Steps to execute a process flow automatically

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

2. Click **[+] Process Flow** to expand the tree and then click **Process Flow**. A The Manage Process Flow screen is displayed (refer to Figure 16.4).
3. Select the radio button adjacent to the required process flow and click **BindEvent** link. The Create Event Registry screen is displayed (see Figure 16.109).

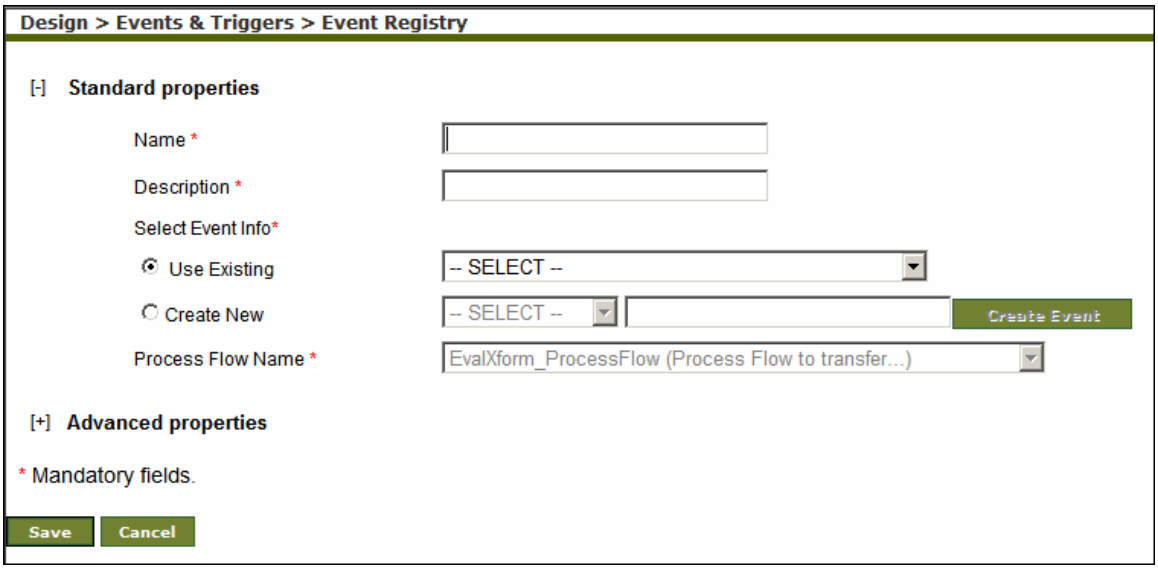


Figure 16.109: Create Event Registry

4. Enter the name and description of the event registry in their respective fields.
5. To bind an existing event to the process flow, select the *Use Existing* radio button, and select the event from the drop-down list. Else, to bind a new event to the process flow, select the *Create New* radio button. Select the event type from the drop-down list, and then click **Create Event** button. This displays the Create Event screen for the selected event type. For example, if you select Mail Event from the drop-down list, then clicking Create Event button, displays the Create Mail Event screen.
6. Enter the required parameters and click **Save** in the Create Event screen to save the event activity and bind it to the process flow and return the control to Manage Process Flowscreen.

	For details, on creating an event, refer to **Events and Triggers** section.

USAGE RECOMMENDATION

Following points should be kept in mind while working in Process Designer:

- User can View or change Process Flow properties from Process Flow Attribute in **Actions** menu.
- While working in Process Designer, user can login with different username or can switch to other Adeptia Server. To login with other username or switch to other Adeptia Server,

select Enter Login Information from Actions menu. If you switch to another Adeptia Server, you must have an account to another Adeptia Server and that account must have permission to use Process Designer.

For details, on different user types, refer to the *User* section in the *Administrator Guide*.

- Process Designer allows multiple object deletion. More than one activity can be selected by dragging mouse pointer and deleted collectively.

17 CREATING TRIGGER AND EVENTS

Trigger Events are used to schedule and trigger a process flow. Trigger Events enable you to specify when and how frequently the process flow should be executed on a recurring basis. The types of trigger events are outlined as:

- **Calendar Event**
- **Database Event**
- **Event Registry**
- **File Event**
- **FTP Event**
- **HTTP Trigger**
- **JMS Event**
- **Mail Event**
- **Timer Event**
- **Web Service Trigger**

By default, a trigger event remains deactivated after its creation. To trigger a process flow using the trigger events, you must bind it with appropriate trigger events and then activate those events.

CREATING CALENDAR EVENT

This service enables you to specify the recurring execution of process flow between the specified dates in conjunction with a specified calendar.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a Calendar Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Calendar**. The Manage Calendar Event screen is displayed (see Figure 17.1).

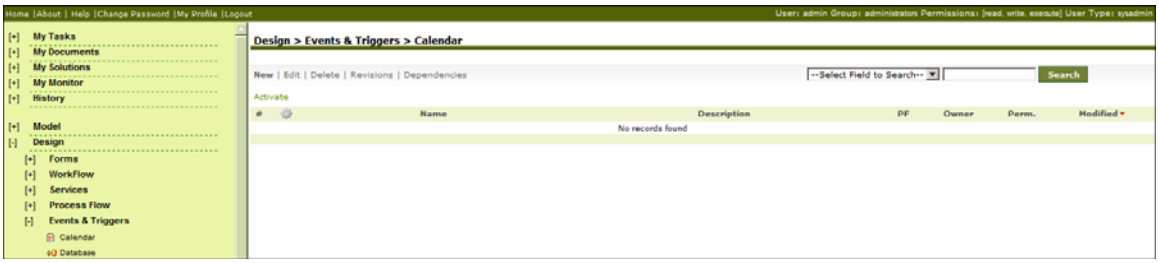


Figure 17.1: Manage Calendar Event

3. Click the **New** link. The Create Calendar Event screen is displayed (see Figure 17.2).

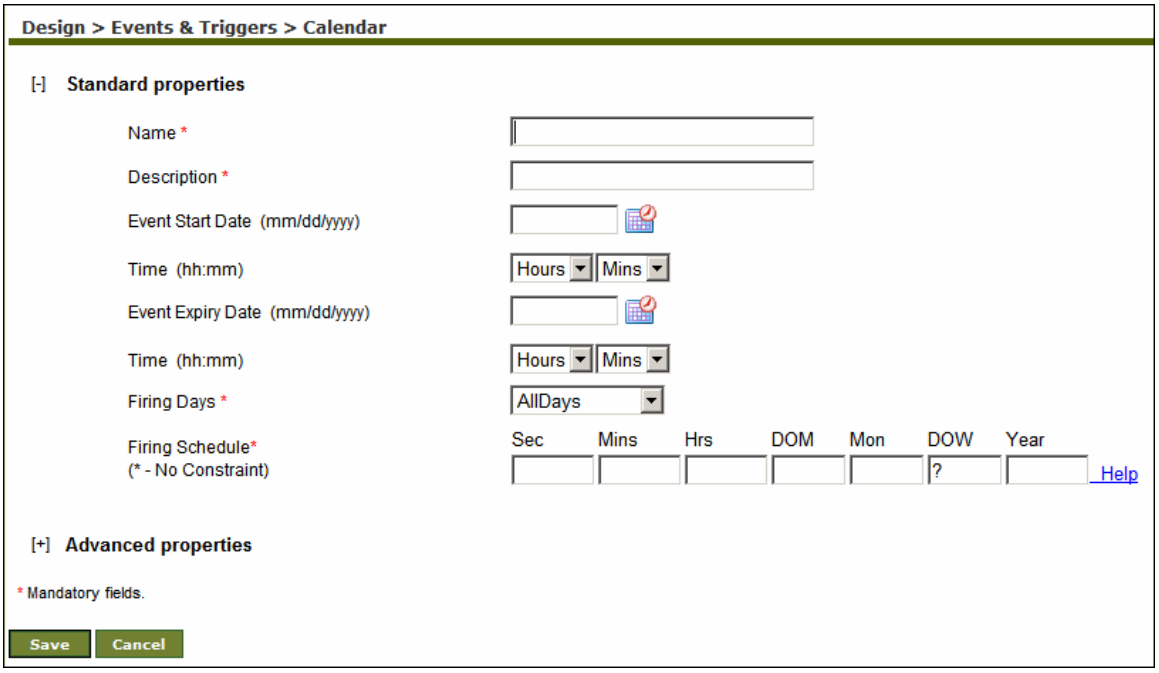


Figure 17.2: Create Calendar Event

4. Enter the name and description of the new Calendar Event activity in the *Name* and *Description* fields respectively.
5. Enter the date from which Calendar event will start triggering; in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon to select the required date from the calendar.
6. Enter the start time from the *Time* drop-down list.
7. Enter the date on which Calendar event will stop triggering, in the *Event Expiry Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon and select the required date from the calendar.
8. Enter the expiry time from the *Time* drop-down list.
9. Select the days of week on which the event should fire from the *Firing Days* drop-down list.

Types of firing days are described in the table below.

Table 17.1: Types of Firing Days

Days	Description
All Days	The event will fire on all days (Mon to Sun) of the week.
Business Days	The event will fire from Monday to Friday excluding holidays. To know how to specify holidays, refer to the *Business Calendar* section of *Appendix A* in *Administrator Guide*.
Week Days	The event will fire from Monday to Friday even if there are any holidays.

10. Define the frequency of execution in the *Firing Schedule* fields.

For more details about Firing Schedule, click **Help** or refer to *Appendix B: Cron Expression* in *Administratoristrator Guide.*

To learn about Advanced Properties refer to **Changing Advanced Properties** section.

11. Click **Save** button. This displays a screen confirming that the Calendar event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the calendar event (refer to Figure 3.6).
12. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

13. Click **OK** to save the comments. This displays a screen confirming that the calendar event has been created successfully.

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

You can view details of a process flow associated with a calendar event, by clicking the process flow displayed under *Associated Process Flows* on the Manage Calendar Event screen.

CREATING DATABASE EVENT

The Database Event enables you to schedule a process flow to be triggered when a record is inserted, updated or deleted in a database table.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Prerequisites:

- *Database Info* activity must be created before creating *Database Event* activity.

Steps to create a Database Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Database**. The Manage Database Event screen is displayed (see Figure 17.3).

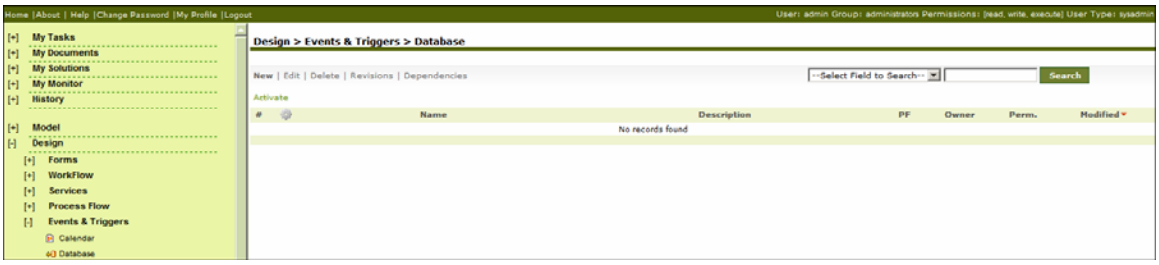


Figure 17.3: Manage Database Event

3. Click the **New** link. The Create Database Event screen is displayed (see Figure 17.4).

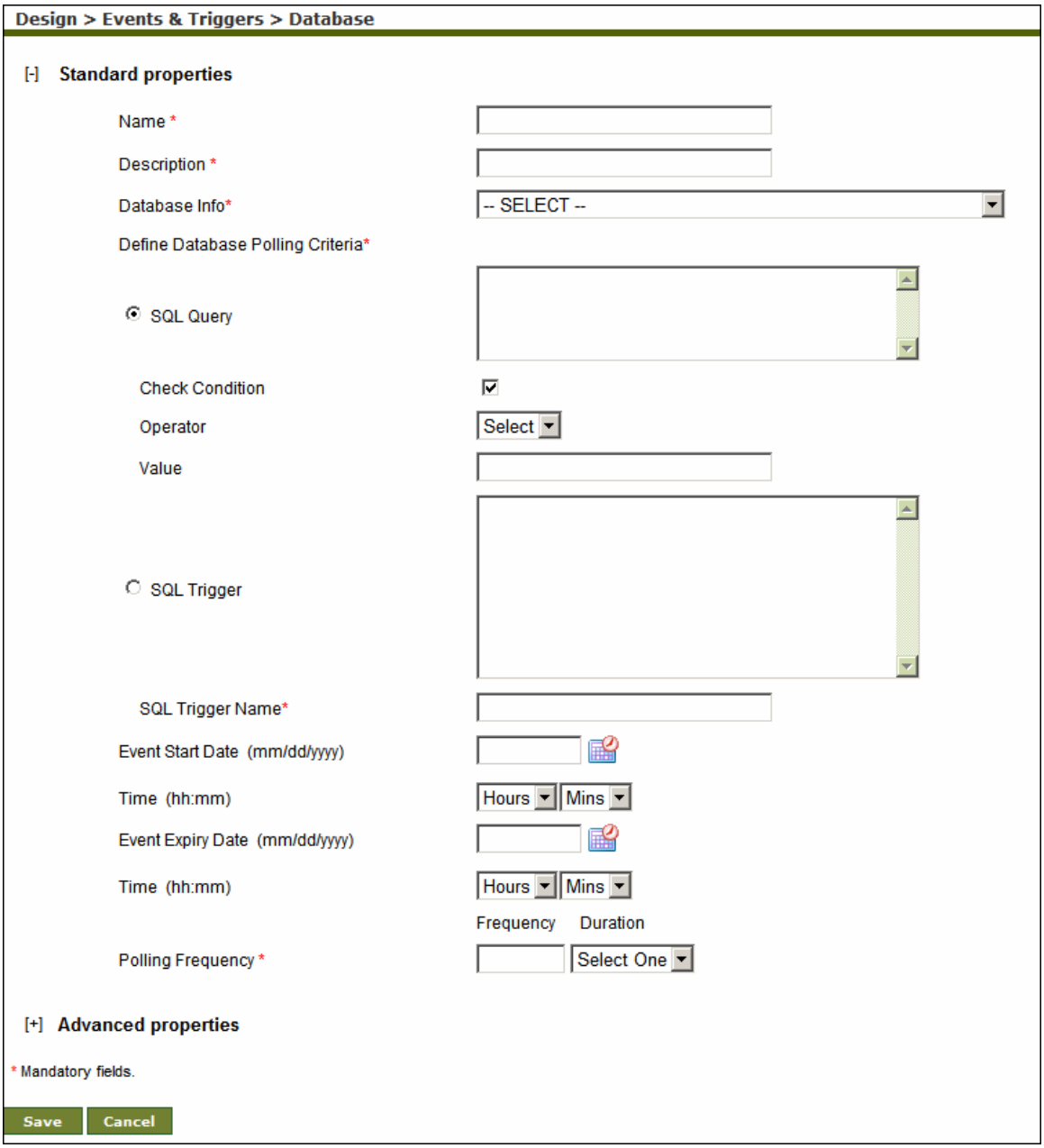


Figure 17.4: Create Database Event

4. Enter the name and description of the new Database Event activity in the *Name* and *Description* fields respectively.

5. Select the Database Info Id activity from the *Database Info Id* drop-down list.

	To learn how to create Database Info activity, refer to the section *Creating Database Info* in *Administrator Guide*.

6. You can create the database event definition by entering an SQL Query or a database trigger command. By default, *SQL Query* option is selected. Enter the query in the *SQL Query* field. Mark the *Check Condition* checkbox as checked, if you want to trigger the process flow based on a condition. The result of the query will be compared with a conditional value, and the process flow will be triggered, if the values match. Select the operator for the query from the *Operator* drop-down list. Enter the value to be compared in the query in the *Conditional Value* field. The query should return only one record. If the query returns multiple records, then only the first record is accepted. If the query returns one record, then it will compare the value of the first field with the value specified in the *Conditional Value* field. If the value matches, then the process flow is triggered. If the value does not match, then the system simply logs an error. It does not trigger the process flow.

If the *Check Condition* checkbox is marked as unchecked, then the process flow is triggered each time a row is returned.

7. Alternately, enter the database trigger command in the *SQL Trigger* field.

Following is the format of trigger used to trigger the process flow:

```
<Trigger Text>
  INSERT INTO dbeventtriggertable VALUES ('Query =<WHERE CLAUSE>');
  END <trigger name> ;
```

Edit the parts, which are within < >. You can define a 'Where' clause that indicates the row that is updated. When the command is parsed, it will return the updated row from the database source.

Do not delete the Insert query.

<trigger name> after the END tag should be used for Oracle only. In case of SQL server, <trigger name> is not needed.

Following is the example of the trigger used for SQL Server :

```
create trigger Trigger_test on emp for
insert,update
as
declare @empname varchar(20)
begin
set @empname=(select empname from inserted);
INSERT INTO dbeventtriggertable VALUES ('Query =WHERE empname='''
+@empname+''');

 END  ;
```

Following is the example of the trigger used for Oracle:

```
CREATE OR REPLACE TRIGGER Trigger_test
AFTER INSERT OR UPDATE ON Emp FOR EACH ROW                    BEGIN
INSERT INTO dbeventtriggertable VALUES ('Query = where rowid= ' || :ne
w.rowid );  END Trigger_test;
```

	Here : *Trigger_test* is name of the trigger. *Emp* is the name of the user table on which insert or update operation has to be done. *dbeventtriggertable* is the name of the temporary table used. Do not change it.

8. Enter the name of Trigger in the *SQL Trigger Name* field.
9. Enter the date from which Database event will start triggering; in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click calendar icon and select the required date from the calendar.
10. Enter the start time from the *Time* drop-down list.
11. Enter the date on which Database event will stop triggering, in the *Event Expiry Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon and select the required date from the calendar.
12. Enter the expiry time from the *Time* drop-down list.
13. Enter the time interval, the database event checks the database Server in the *Polling Frequency* field. Enter the digit in the Frequency field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

	Recommended minimum Polling Frequency is 30 seconds. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

14. Click **Save** button. This displays a screen confirming that the database event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the database event (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the database event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a database event, by clicking the process flow displayed under *Associated Process Flows* on the Manage Database Event screen.

CREATING EVENT REGISTRY

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to Register a Process Flow with a Trigger Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Event Registry**. The Manage Event Registry screen is displayed (see Figure 17.5).

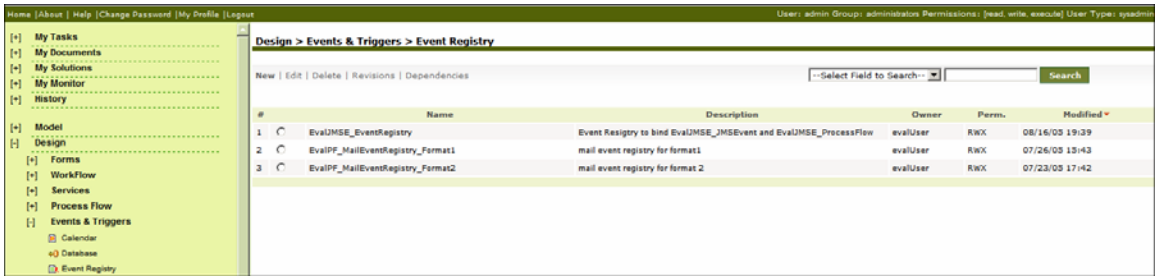


Figure 17.5: Manage Event Registry

3. Click the **New** link. The Create Event Registry screen is displayed (see Figure 17.6).

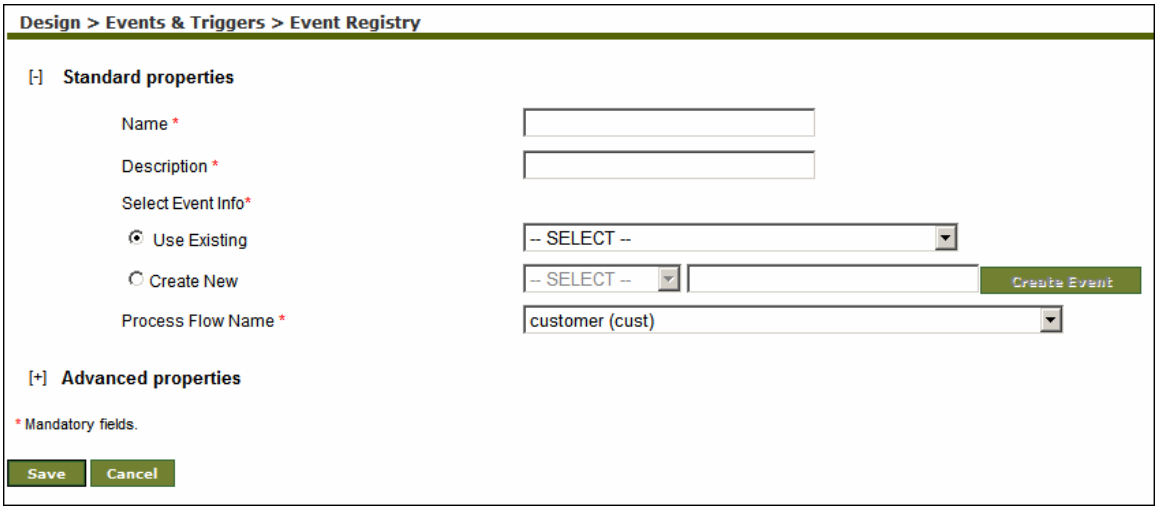


Figure 17.6: Create Event Registry

4. Enter the name and description of the new Event Registry activity in the *Name* and *Description* fields respectively.
5. Select the required Events activity.

	To learn how to create event activity, refer to **Creating Trigger and Events** section.

6. To select an existing event activity, select the *Use Existing* radio button and select the event activity from the drop-down list.
7. To create a new event activity, select the *Create New* radio button, select the event type from the drop down list and then click **Create Event** button. This displays the selected *Create Event* screen.
8. Enter the required parameters and click **Save** in the Create Event screen to save the event activity and return to Create Event Registry screen.
9. Select the required Trigger Event from the *Event Name* drop-down list.
10. Select the required process flow from the *Process Flow Name* drop-down list.

	To learn about Advanced Properties refer to section **Changing Advanced Properties** section.

11. Click **Save** button. This displays a screen confirming that the event registry has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the event registry (refer to Figure 3.6).
12. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING FILE EVENT

The File Event enables you to specify when and how frequently a process flow should be executed based on either creation of a new file, or existence of a file(s) in a pre-defined location or upon its modification.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a File Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **File**. The Manage File Event screen is displayed (see Figure 17.7).

Figure 17.7: Manage File Event

3. Click the **New** link. The Create File Event screen is displayed (see Figure 17.8).

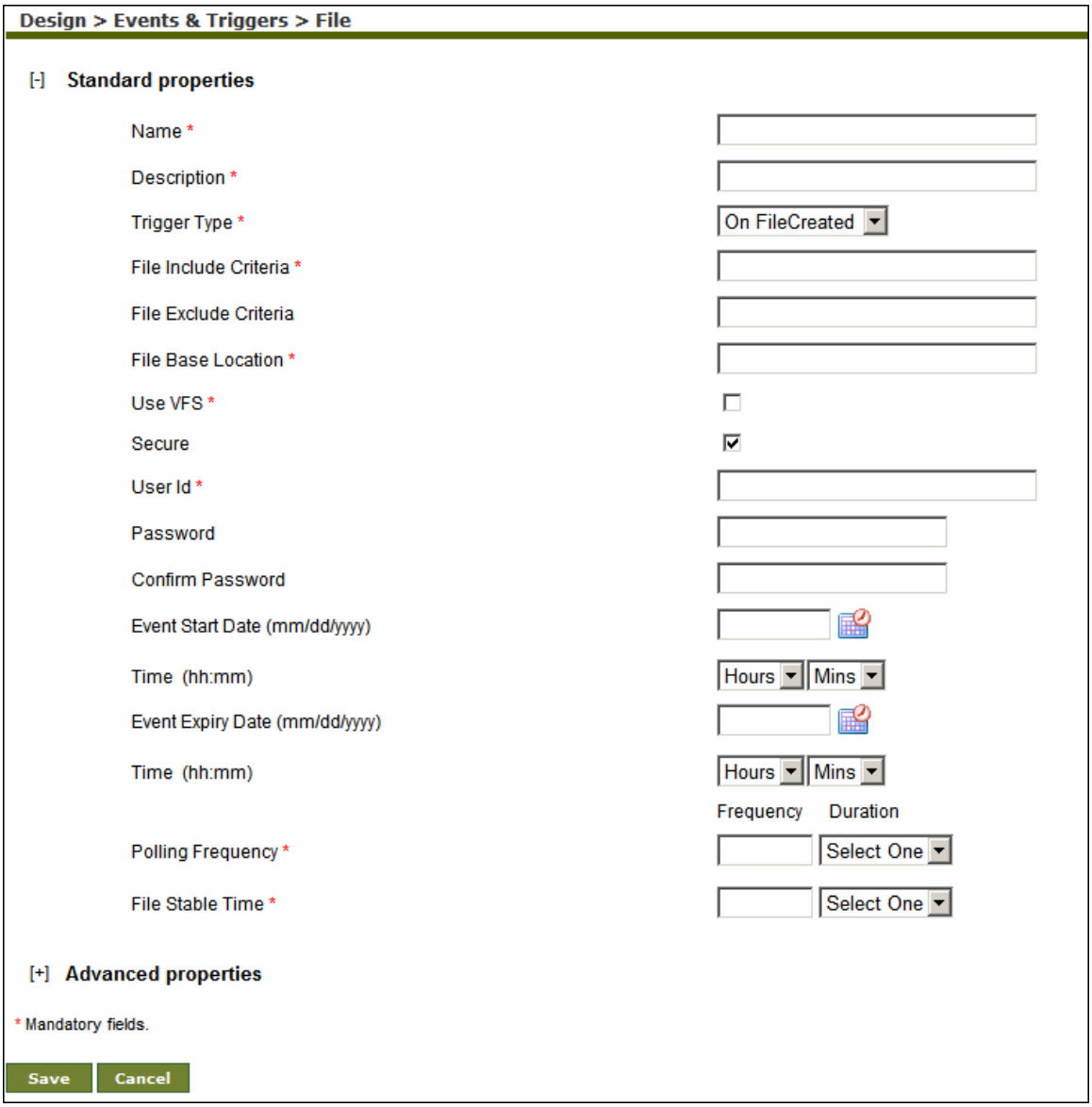


Figure 17.8: Create File Event

4. Enter the name and the description of the new File Event activity in the *Name* and *Description* fields respectively.

5. Select the trigger type from *Trigger Type* drop-down list. The effect on the selection is listed in the table below.

Table 17.2: Trigger Type Selection Values

Trigger Type Selection	Description
On FileCreated	To configure the file event to check for the creation of a new file(s).

On FileExists	To configure the file event to check for the existence of the file(s)
On FileModified	To configure the file event to check for any modification in file(s)

6. Enter the file name that the file event needs to verify, in the *File Include Criteria* field.

7. Enter the name of file that file event does not need to verify, in the *File Exclude Criteria* field. For example **.txt* is entered in *File Include Criteria*, but two files *Gdata.txt* and *Gdata1.txt* file are not required to be verified by File Event. Then *Gdata.txt* and *Gdata1.txt* file name need to be entered separated by comma in *File Exclude Criteria* field. To specify more than one file in *File Include Criteria* and *File Exclude Criteria*, you can use regular expressions listed in the table below.

Table 17.3: Expressions used in File Include Criteria and File Exclude Criteria

Expression	Description
.	For all files with some extension
*	For all files in a directory
a*.txt	For files starting with a and having extension txt (e.g. arch.txt)
a??????.txt	For files starting with a and have 6 more character followed by txt extension (e.g. archive.txt)
a[1-9]	For a1, a2 ,a3 ……..a9
b[aiu]t	For bat, bit or but
a.txt, a.doc	For two files named as a.txt and a.doc

If more than one file is specified in the *File Include Criteria* field, process flow will triggered for each file.

8. Enter the path of file in *File Base Location* field. Example c:/Gmdata

You can also use regular expression for folders, in *File Include Criteria* and *File Exclude Criteria* field. For example, if you enter *h*/*.txt* in *File Include Criteria* field and *C:/Gmdata* in *File Base Location* field, it will search for all .txt file inside all directories which starts from h under C:\Gmdata.

9. When Adeptia Server is installed on Windows Operating System, File Event uses windows service to connect to remote machine to access any file. It just connects once and uses the same connection with the same User ID and Password (which is stored in the cache) every time. If you want to enforce the validation of User ID and Password every time while accessing the file on a remote machine, check *Use VFS* checkbox.

10. If the File Event is secured i.e. username and password are required to access it, then check the *Secure* checkbox and enter the username and password required to access the file in the *User ID* and *Password* fields respectively. This option is applicable only when the file specified is located on a remote machine.

11. Enter the date from which file event will start triggering, in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click calendar icon and select the required date from the calendar.
12. Enter the start time from the *Time* drop-down list.
13. Enter the date on which file event will stop triggering, in the *Event Expiry Date* field. The date must be in *mm/dd/yyyy* format. Click calendar icon and select the required date from the calendar.
14. Enter the expiry time from the *Time* drop-down list.
15. Enter the time interval for which file event will check for the arrival of any file or upon modification of existing file in the *Polling Frequency* field. Enter the digit in the *Frequency* field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

	Recommended minimum Polling Frequency is 30 seconds.

16. Enter the file stable time in the *File Stable Time* field. This is applicable only when user selects On FileCreated or On FileModified in trigger type. Trigger will wait for the specified time for the file to become stable.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

17. Click **Save** button. This displays a screen confirming that the file event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the file event (refer to Figure 3.6).
18. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

19. Click **OK** to save the comments. This displays a screen confirming that the file event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a file event, by clicking the process flow displayed under *Associated Process Flows* on the Manage File Event screen.

CREATING FTP EVENT

The FTP Event enables you to specify when and how frequently a process flow should be executed based on either creation of a new file, or existence of a file(s) on a FTP Server or upon its modification.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a FTP Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **FTP**. The Mange FTP Event screen is displayed (see Figure 17.9).

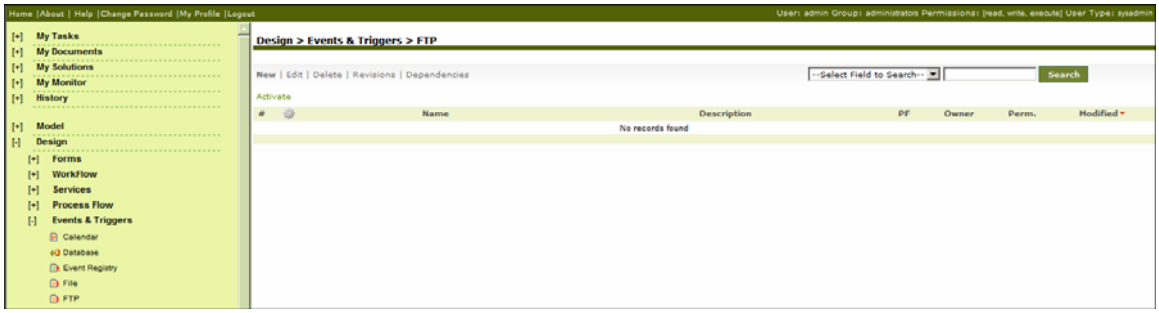


Figure 17.9: Manage FTP Event

3. Click the **New** link. The Create FTP Event screen is displayed (see Figure 17.10).

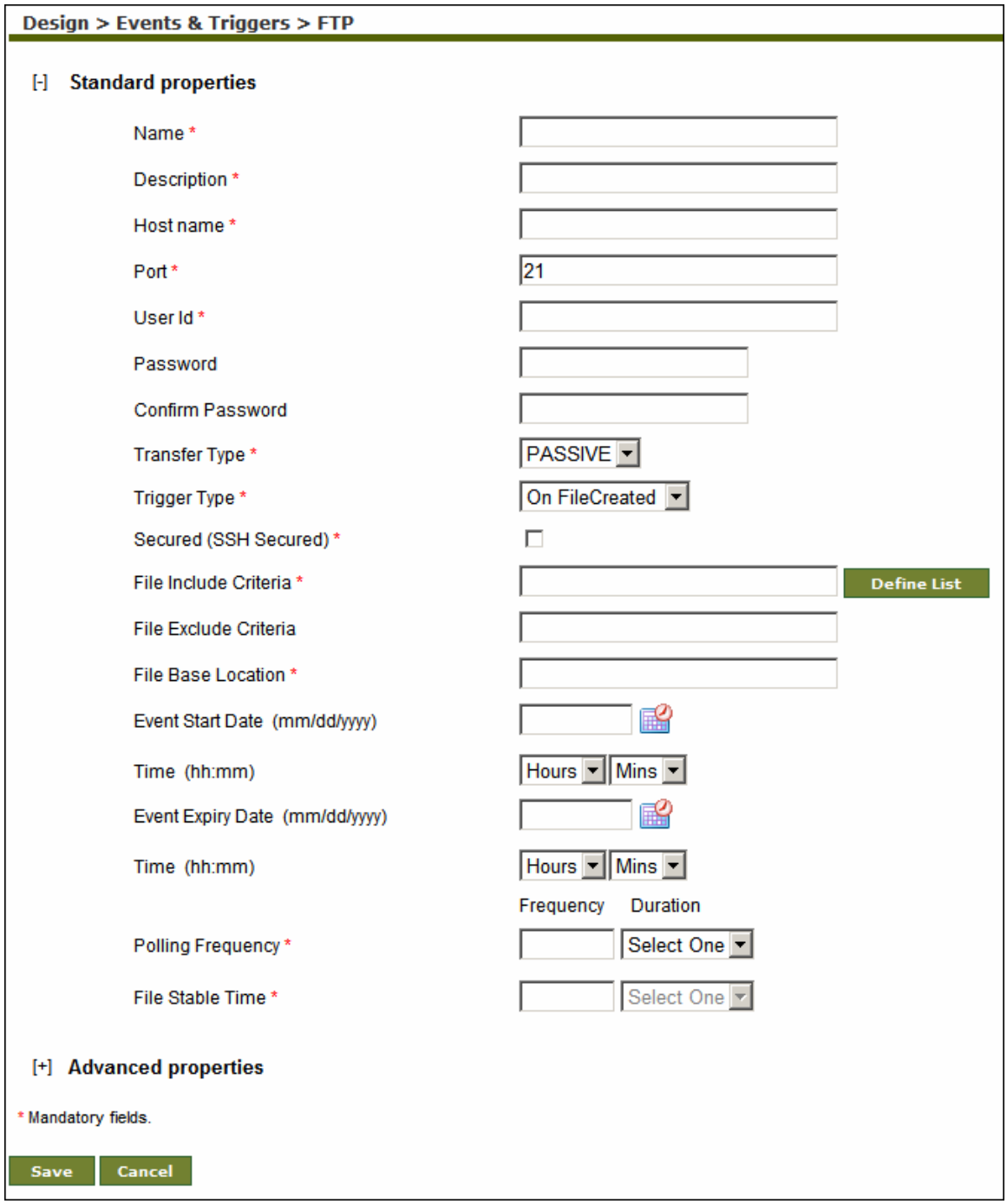


Figure 17.10: Create FTP Event

4. Enter the name and description of the new FTP Event in the *Name* and *Description* fields respectively.

5. Enter the name and port number of the FTP Server in the *Host Name* and *Port* field respectively.

6. Enter username and password required to access FTP Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the Confirm Password field.

7. Select the transfer type as either *Active* or *Passive* from the *Transfer Type* drop-down list. Active transfer is more secure since the client only initiates communication to the server on one port whereas in case of Passive transfer the client initiates communication to the server over two ports. Passive mode is useful when you are behind a firewall or a proxy.

8. Select the trigger type from the *Trigger Type* drop-down list. For details of the selection, refer to Table 17.2.

9. Check the *SSH (Secure Shell)* checkbox if the FTP event accesses a secure FTP Server. When SSH is used to protect FTP Server, the control connection between the FTP client and Server is encrypted.

10. Enter the name of file that FTP event needs to verify, in the *File Include Criteria* field.

In File Include Criteria and File Exclude Criteria you can also give the folder name along with the file name. For example suppose you have entered *C:\Gmdata* in *File Base Location* field. There are two sub-folders *Purchase* and *Purchase1*in *Gmdata* folder. Now suppose you don't know whether the file is in *Purchase* or *Purchase1* folder. To handle this scenario you can define File Include Criteria as defined below:

Purchase/*.txt,Purchase1/*.txt.

You can also use regular expression in the folder name as given below:

Pur*/*.txt

This path includes both the folder *Purchase* and *Purchase1*.

If you have large number of paths that need to be define, you can also use *Define List* option.

11. To define path in *File Include Criteria,* click *Define List* button. The *File Include Criteria List* screen is displayed (see Figure 17.11).

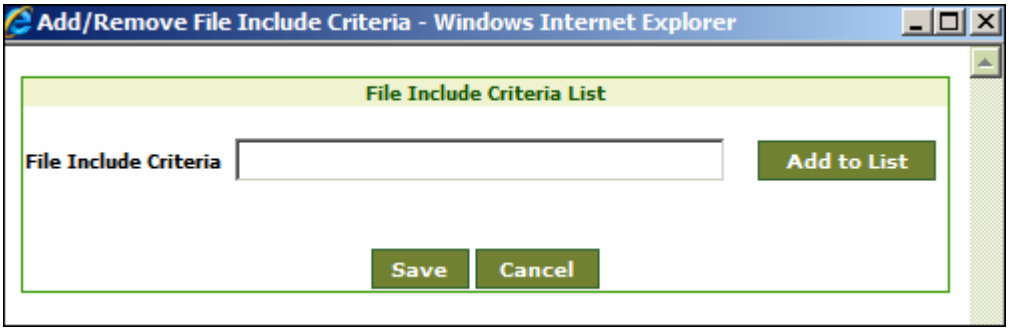


Figure 17.11: Add Include Criteria List

12. To add path enter the path in *File Include Criteria* field and click *Add to List* button. The added path are shown in *File Include Criteria List* (see Figure 17.12).

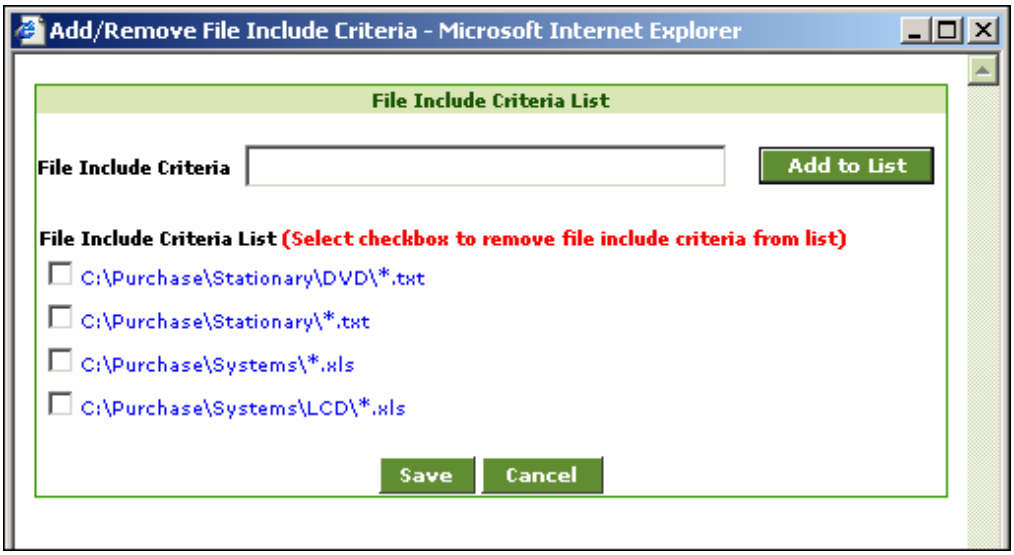


Figure 17.12: File Include Criteria List

13. Click *Save* button to save the added path in *File Include Criteria* field. The added paths are populated in the *File Include Criteria Field* in the *Create FTP Event* page.

	If you want to remove some path form the *File Include Criteria,* Check the checkbox adjacent to the required path and then click *Save.* The selected path is not populated in the *File Include Criteria* field of the *Create FTP Event* page.

14. Enter the name of file that file event does not need to verify, in the *File Exclude Criteria* field. For example **.txt* is entered in *File Include Criteria*, but two files *Gdata.txt* and *Gdata1.txt* file are not required to be verified by FTP Event. Then *Gdata.txt* and *Gdata1.txt* file name need to be entered separated by comma in *File Exclude Criteria* field. To specify more than one file in *File Include Criteria* and *File Exclude Criteria*, you can also use regular expressions. These are listed in Table 17.3

	If more than one file is specified in the *File Include Criteria* field, process flow will triggered for each file.

15. Enter the path of file in *File Base Location* field. Example c:/Gmdata.
16. Enter the date from which FTP event will start triggering, in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click calendar icon and select the required date from the calendar.
17. Enter the start time from the *Time* drop-down list.
18. Enter the date on which FTP event will stop triggering, in the *Event Expiry Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon and select the required date from the calendar.
19. Enter the expiry time from the *Time* drop-down list.

20. Enter the time interval for which FTP event will check for the arrival of any file or modification of existing file in the *Polling Frequency* field. Enter the digit in the Frequency field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

	Recommended minimum Polling Frequency is 30 seconds.

21. Enter the file stable time in the *File Stable Time* field. This is applicable only when user selects On FileModified in trigger type. Trigger will wait for the specified time for the file to become stable.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

22. Click **Save** button. This displays a screen confirming that the FTP event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the FTP event (refer to Figure 3.6).
23. Enter the comments in the *Add Comments* 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 FTP event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a FTP event, by clicking the process flow displayed under *Associated Process Flows* on the Manage FTP Event screen.

CREATING HTTP TRIGGER

The HTTP trigger enables you to schedule a process flow to be triggered when an HTTP request is made to Adeptia Server. It also allows the request to pass the data to the process flow. The trigger can be used by a HTTP client application to integrate with process flow, deployed on the Adeptia Server.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a HTTP Trigger

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **HTTP**. The Manage HTTP Trigger screen is displayed (see Figure 17.13).

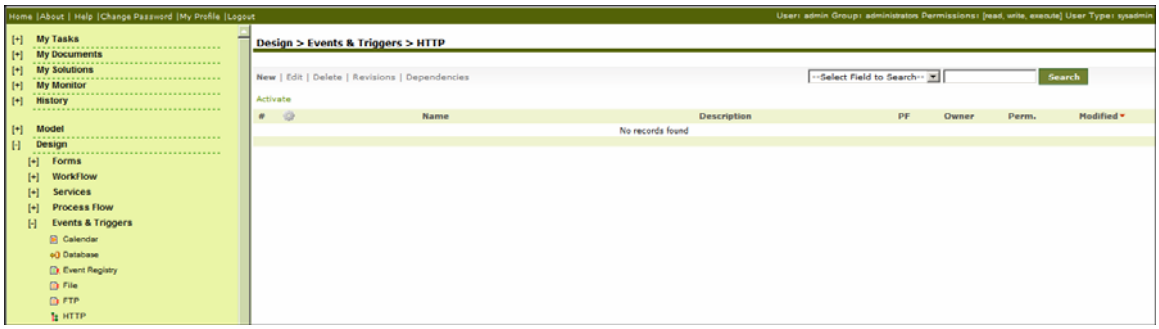


Figure 17.13: Manage Http Trigger

3. Click the **New** link. The Create HTTP Trigger screen is displayed (see Figure 17.14).

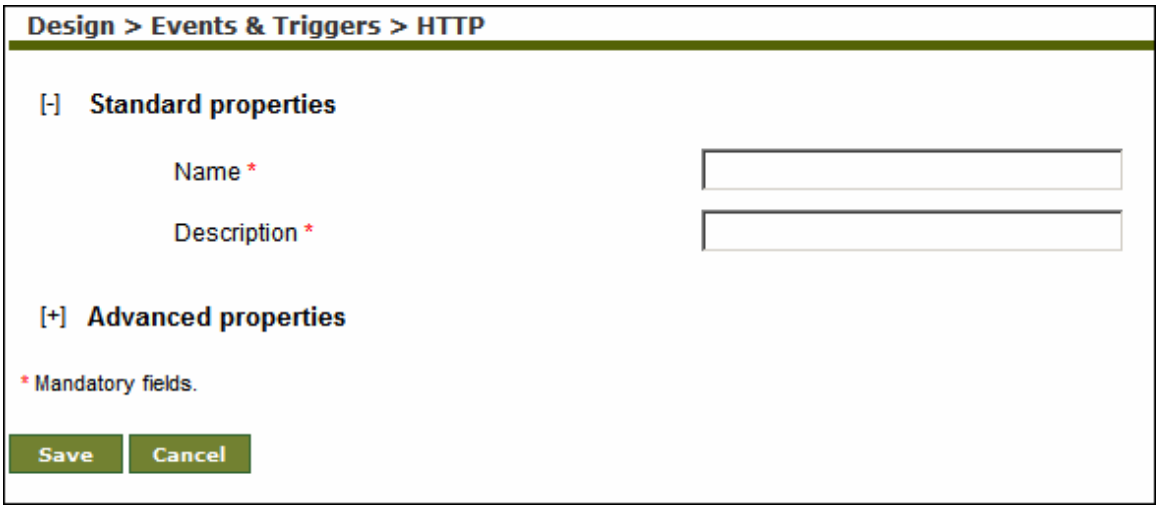


Figure 17.14: Create Http Trigger

4. Enter the name and description of the new HTTP Trigger activity in the *Name* and *Description* fields respectively.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

5. Click **Save** button. This displays a screen confirming that the HTTP event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the HTTP event (refer to Figure 3.6).
6. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a HTTP event, by clicking the process flow displayed under *Associated Process Flows* on the Manage HTTP Event screen.

Usage Recommendation

To read the context variable, within a process flow following steps are required:

1. While creating the process flow, create a process flow variable. To know, how to create Process Flow Variable, refer to the section **Creating Process Flow Variable**.
2. Use this process flow variable as Context Source within process flow. To know, how to use a Process Flow Variable as Context Source, refer to the section **Using Context Source and Context Target.**
3. Change the value of the Property Event Context Enabled to yes.
4. Post the data using your HTTP client application to the following URL:

   http://<ServerName>:<ServerPort>/adeptia/receiveservlet?activityID=<EntityID>&userID=<LoginName>&password=<LoginPassword>&group=IndigoGroup:<Group ID>

where

ServerName : *Name of the server where Adeptia Server is running*
ServerPort : *Port at which Adeptia Server is running. By default, it is 8080.*
EntityID : *30 digit ID of the HTTP Trigger activity. To view Entity ID of the HTTP Trigger, click View in the HTTP Trigger Page.*
LoginName : *User ID of the Adeptia Server*
LoginPassword : *Password of the Adeptia Server*
Group ID : *30 digit ID of the group, the user belongs to. To view Group ID of the group, click View in the Manage group page.*

CREATING JMS EVENT

The JMS Event enables you to specify when and how frequently the process flow should be executed if any message is updated in a queue or topic of a JMS Server.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Prerequisites:

- *JMS Provider* activity must be created before creating *JMS Event* activity.

Steps to create a JMS Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **JMS**. The Manage JMS Event screen is displayed (see Figure 17.15).

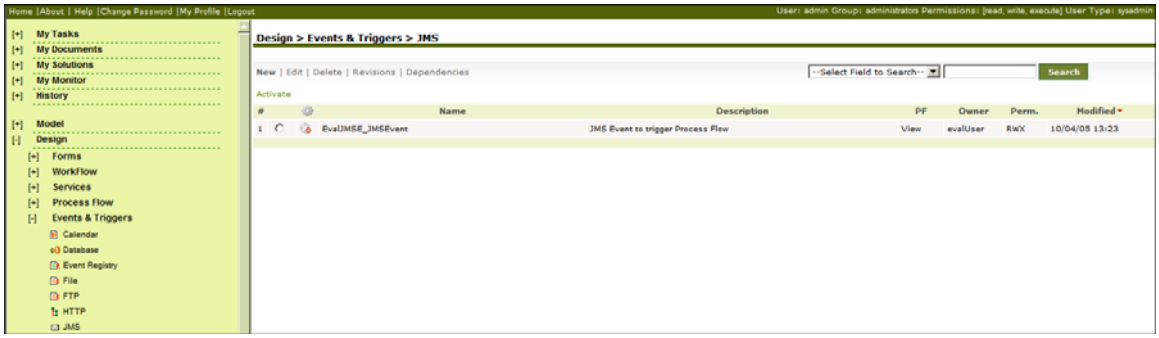


Figure 17.15: Manage JMS Event

3. Click the **New** link. The Create JMS Event screen is displayed (see Figure 17.16).

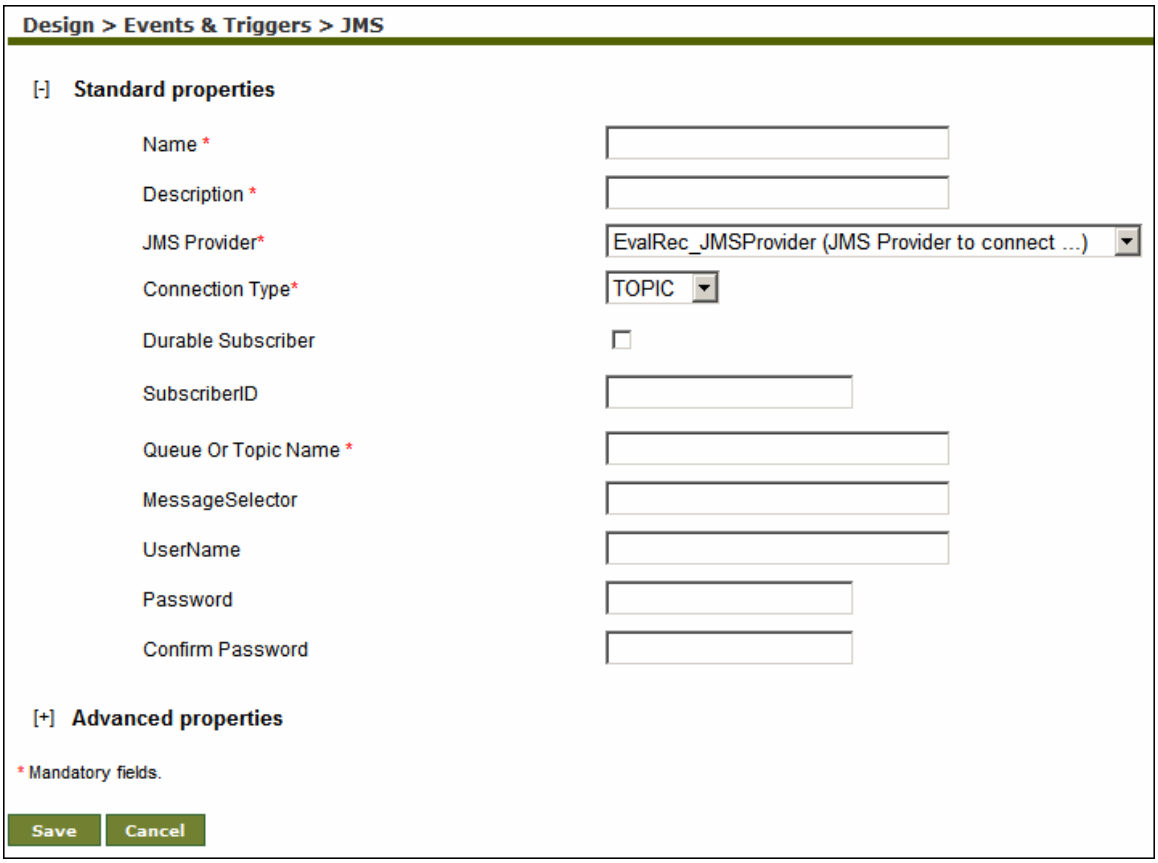


Figure 17.16: Create JMS Event

4. Enter the name for the new JMS Event in the *Name* field. Then, enter the description for the JMS Event in the *Description* field.

5. Select the JMS Provider activity from the *JMS Provider* drop-down list.

	To learn how to create JMS Provider activity, refer to the section *Creating JMS Provider* in *Administrator Guide*.

6. Select the Connection Type as either *Topic* or *Queue* from the *Connection Type* drop-down list.

7. Check the *Durable Subscriber* check box if the JMS Subscriber is durable. 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.

8. Enter the subscriber ID in the *Subscriber ID* field.

9. Enter the name of queue or topic as configured in the JMS Server in the *Queue Or Topic Name* field.

10. If you want to select a specific message from the JMS Server, enter the message selector in the *Message Selector* field.

	The message selector is used to specify the filter criterion to receive a message that the user is interested in. The messages can be filtered based on only header references and properties references of the message. The message selector uses SQL92 query syntax to define the filter criteria. SQL92 is widely used to query the entire standard databases i.e. Oracle, SQL Server. The only difference between the database query and the message selector query is that the message selector uses, only a part of the query which is after the where clause.
	
	The following message selector selects messages with a message type of car and color of blue and weight greater than 2500 pounds:
	
	`JMSType = 'car' AND color = 'blue' AND weight > 2500`
	
	The following message selector selects message with the property Sport has value either as Basketball or Football.
	`Sport in ('Basketball','Football')`

11. Enter the username and password required to connect to JMS Server in the *UserName* and *Password* fields respectively. Then, re-enter the password in the *Confirm Password* field.

	To learn about Advanced Properties refer to **Changing Advanced Properties** section.

12. Click **Save** button. This displays a screen confirming that the JMS event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the JMS event (refer to Figure 3.6).
13. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

14. Click **OK** to save the comments. This displays a screen confirming that the JMS event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a JMS event, by clicking the process flow displayed under *Associated Process Flows* on the Manage JMS Event screen.

CREATING MAIL EVENT

The Mail Event allows you to schedule a process flow to be triggered when a specified mail arrives on the mail Server.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	

Steps to create a Mail Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Mail**. The Manage Mail Event screen is displayed (see Figure 17.17).

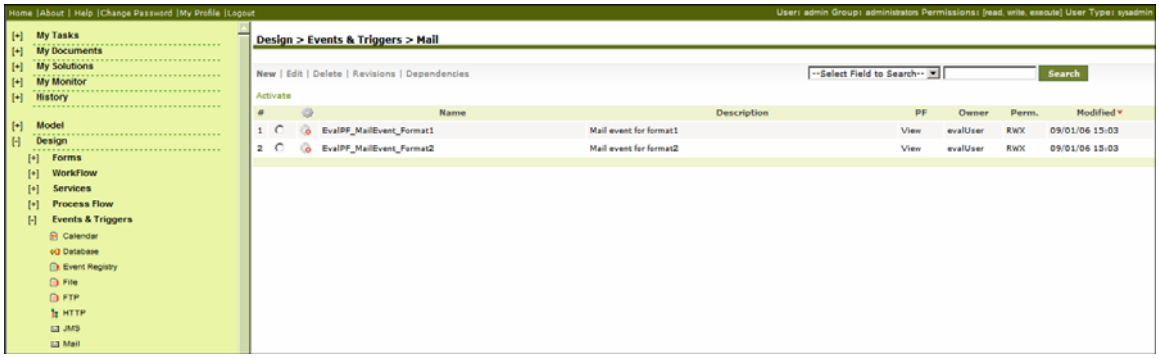


Figure 17.17: Manage Mail Event

3. Click the **New** link. The Create Mail Event screen is displayed (see Figure 17.18).

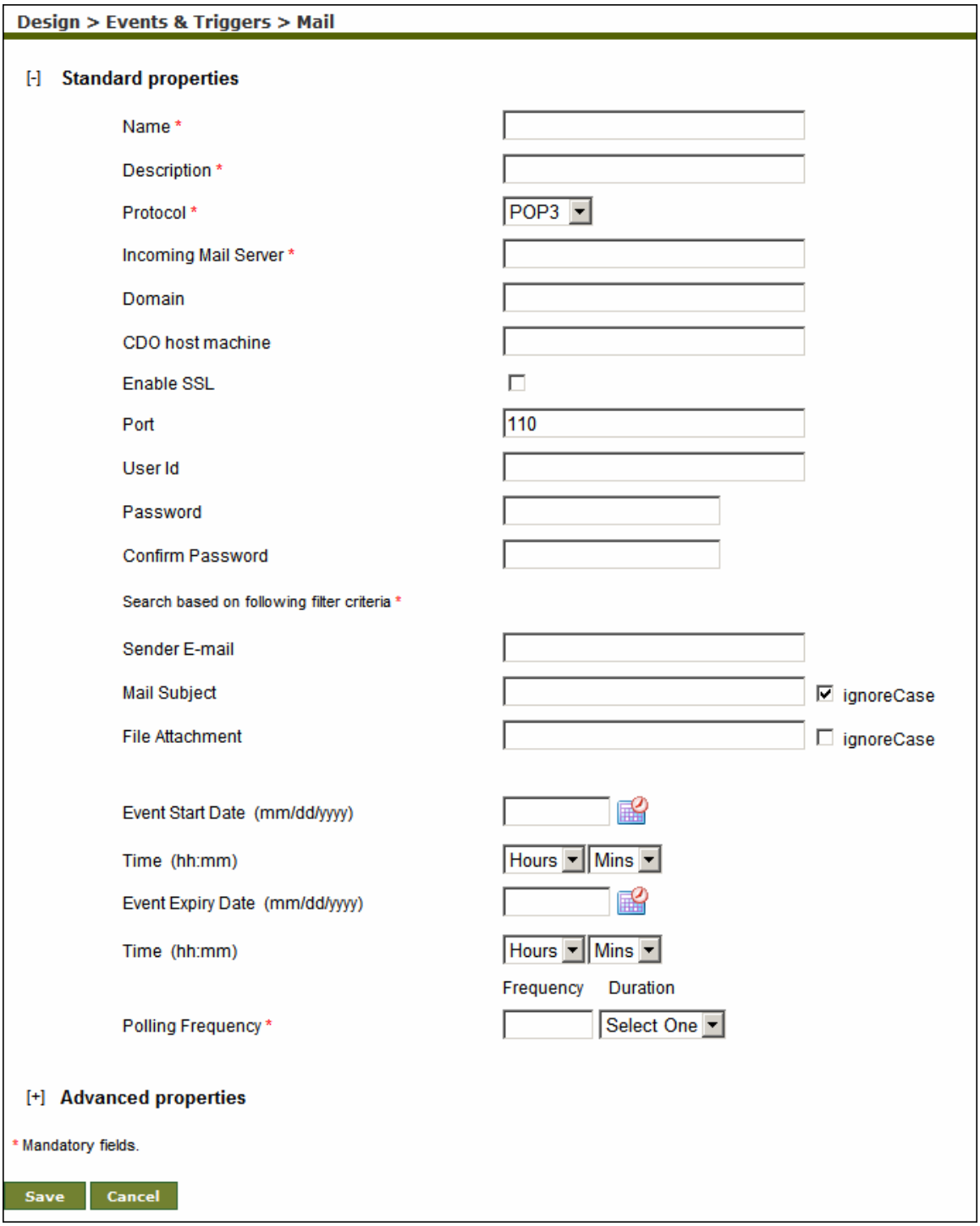


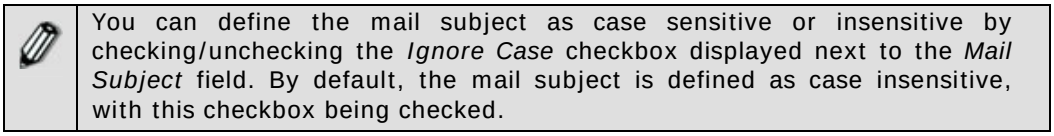
Figure 17.18: Create Mail Event

4. Enter the name and description of the new Mail Event activity in the *Name* and *Description* fields respectively.

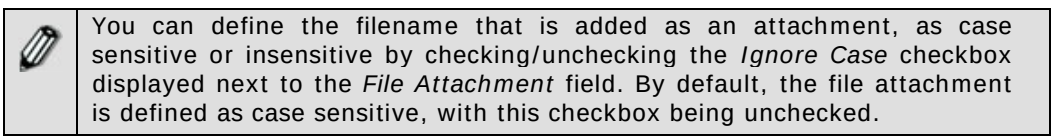
5. Select the Internet standard protocol to be used for retrieving incoming mails, from the *Protocol* drop-down list. You can select the POP3 protocol, IMAP4 protocol or the MAPI protocol. Based on the selected protocol, the default port number for that protocol is displayed in the Port field. The MAPI protocol is used to access mails on the Microsoft Exchange Server. It uses J-Integra as a bridge and uses a domain for exchanging mails, instead of a port. Thus, if the MAPI protocol is selected, then the port number field appears as disabled.
6. Enter the address of the incoming mail server in the *IncomingMail Server* field.
7. Enter the domain name that is configured for the Microsoft Exchange Server in the *Domain* field. This field appears as enabled only when the MAPI protocol is selected in the *Protocol* drop-down list.
8. Enter the host machine name on which CDO is installed and configured, in the *CDO Host Machine* field. This field appears as enabled only when the MAPI protocol is selected in the Protocol drop-down list. When Java applications use J-Integra to exchange mails on Micorosoft Exchange Server, then CDO serves as the intermediary between the Java application and Microsoft Exchange Server.
9. Check the *Secure* checkbox, if the specified incoming mail server is SSL enabled.
10. The default port number for the selected protocol is displayed in the Port field. If you want to change this port number, enter the new port number in the *Port* field. If MAPI protocol is selected, then the port number field appears as disabled.
11. Enter the username and password required to access the mail Server in the *User ID* and *Password* fields respectively. Then, re-enter the password in the Confirm Password field.
12. Select any of the following filter criteria:
    - Sender E-mail
    - Mail Subject
    - File Attachment

    You may select more than one filter criteria.

13. Enter the sender's email address and subject of email in the *Sender Email* and *Mail Subject* fields respectively.



14. Enter the name of the file attached with mail, in the *File Attachment* field.



15. Enter the date from which Mail event will start triggering; in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click calendar icon and select the required date from the calendar.
16. Enter the start time from the *Time* drop-down list.

17. Enter the date on which the Mail event will stop triggering; in the *Event Expiry Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon and select the required date from the calendar.
18. Enter the expiry time from the *Time* drop-down list.
19. Enter the time interval, the Mail event will check for the existence of the mail, in the *Polling Frequency* field. Enter the digit in the *Frequency* field and select the unit of time i.e. seconds, minutes or hours etc from the *Duration* drop-down list.

	Recommended minimum Polling Frequency is 30 seconds.

20. To specify maximum number of emails to be processed at a time, click **[+]** to expand Advanced Properties and enter the required value in the *Mail Process Concurrency* field.

	If there is large number of emails in the mailbox, which meet the search criteria of the mail event, all the mails will be processed at a time. If you want to limit the number of emails to be process at a time with this event, enter the appropriate value in the *Mail Process Concurrency* field. Now mail event will process only the specified number of emails at a time. Remaining emails will be processed at the next polling frequency. Mails are processed on First In First Out (FIFO) basis. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

21. Click **Save** button. This displays a screen confirming that the mail event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the mail event (refer to Figure 3.6).
22. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

23. Click **OK** to save the comments. This displays a screen confirming that the mail event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	If a mail event is deactivated and then activated again, it will trigger an event for an existing email. You can view details of a process flow associated with a mail event, by clicking the process flow displayed under *Associated Process Flows* on the Manage Mail Event screen. A mail event can trigger multiple process flows at a time. Each process flow uses a mail source. When multiple process flows use a mail source at a time, errors can occur. Thus it is advisable to limit the number of process flows triggered by a mail event. You can also set the number of retries and the sleep time between each retry, in case an error occurs when using a mail box. For details on these settings, refer to Appendix A in Administratoristrator Guide.

CREATING TIMER EVENT

The Timer Event enables you to specify the date, time and frequency at which a process flow should be executed.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√		√	√

Steps to create a Timer Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Timer**. The Manage Timer Event screen is displayed (see Figure 17.19).

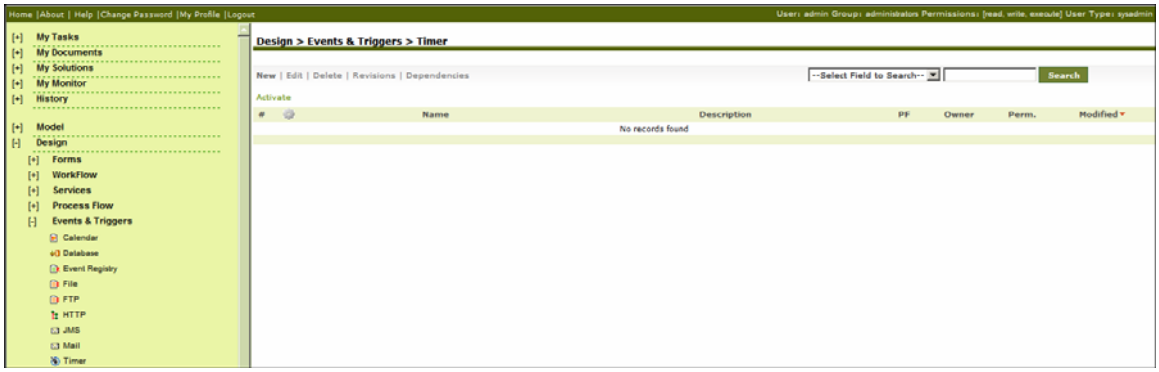


Figure 17.19: Manage Timer Event

3. Click the **New** link. The Create Timer Event screen is displayed (see Figure 17.20).

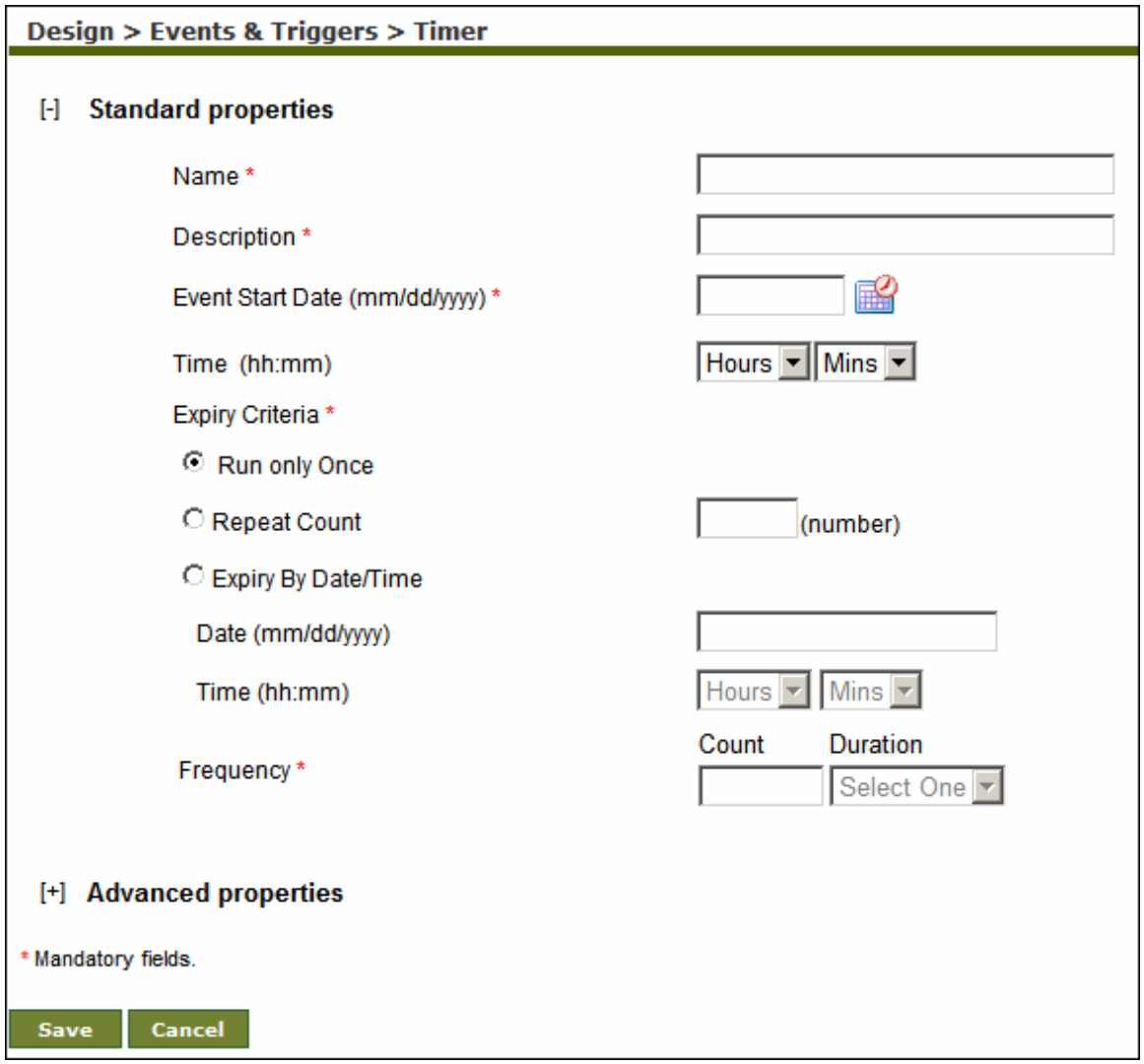


Figure 17.20: Create Timer Event

4. Enter the name and description of the new Timer Event activity in the *Name* and *Description* fields respectively.
5. Enter the date from which the Timer event will start triggering, in the *Event Start Date* field. The date must be in *mm/dd/yyyy* format. Click the calendar icon to select the required date from the calendar.
6. Enter the start time from the *Time* drop-down list.
7. Select one of the *Expiry Criteria* displayed in the table below.

Table 17.4: Expiry Criteria

Expiry Criteria	Description
Run Only Once	Select this option if the process flow needs to be triggered only once.

Repeat Count	Select this option if the process flow needs to be triggered for given number of times. Enter the required number in the Repeat Count field. Progress flow is triggered Repeat Count + 1 times.
Expiry By Date/ Time	Select this option if the process flow needs to be triggered up to the given date and time on a given interval. To select the expiry date click on the calendar and select the required date. Select the expiry time using Hours and Mins drop-down list.

8. If *Repeat Count* or *Expiry By Date/Time* option is selected as expiry criteria, enter the time interval in the *Frequency* field.

	Recommended minimum Polling Frequency is 30 seconds. To learn about Advanced Properties refer to **Changing Advanced Properties** section.

9. Click **Save** button. This displays a screen confirming that the timer event has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments for creating the timer event (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

11. Click **OK** to save the comments. This displays a screen confirming that the timer event has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	You can view details of a process flow associated with a timer event, by clicking the process flow displayed under *Associated Process Flows* on the Manage Timer Event screen.

TRIGGERING PROCESS FLOW USING WEB SERVICE TRIGGER

Web Service Trigger is an interface, which allows a Web Service client to trigger any process flow in Adeptia Server. The Web Service client needs to pass the ID of the process flow and any input parameters that need to be passed into the Process Flow.

To trigger a Process Flow, the Web Service client needs to know details of the Web Service published and format in which input parameter can be passed.

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite

√	√	√	

Steps to get the information required by Web Service Client to trigger a Process Flow

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **Web Service**. The Web Service Trigger screen is displayed (see Figure 17.21).

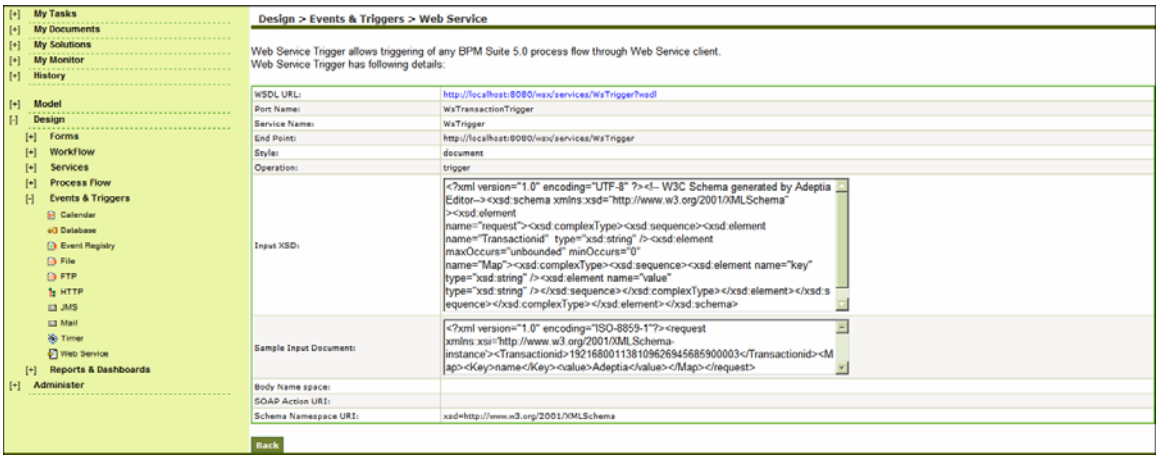


Figure 17.21: Web Service Trigger

3. This screen shows the details of the Web Service published. All the parameters needed by the Web Service client to invoke the service are displayed in this screen.
4. Use the WSDL URL and Sample Input Document displayed in the Figure 17.21 to trigger the Process Flow.
5. A Sample Input Document which is to be used to trigger the process flow is displayed below (see Figure 17.22).

```
<?xml version="1.0" encoding="ISO-8859-1"?><request
xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance'><Transactionid>19216800113810962694568590003</Transactionid>
<Map><Key>name</Key><value>Adeptia</value></Map></request>
```

Figure 17.22: Sample Input Document used to Trigger Process Flow

6. Make sure to replace the Transaction ID with the ID of the Process Flow, which is to be triggered.

	While creating the Web Service trigger, you need to ensure that Transport security type is selected, as this trigger does not support Message security type.

7. To pass the parameter to the process flow, enter the key(name) and the value of the parameter in the *Map* tag of the XML shown above.

	To know the ID of a Process Flow, In the Manage Process Flow page, click **View** link against the process flow.

ACTIVATING TRIGGER EVENTS

Steps to Activate a Trigger Event

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Events & Triggers** to expand the tree and then click **File**. The Manage File Event screen is displayed with the list of existing events (see Figure 17.23).

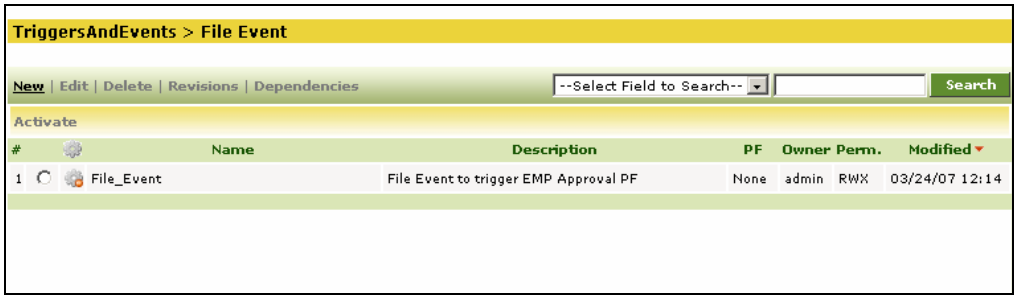


Figure 17.23: Manage File Event

3. To activate the trigger event, select the radio button adjacent to the event and click **Activate**. A screen is displayed confirming the trigger event activity has been activated successfully.

18 USING REPORTS & DASHBOARD

DashBoard collects data from various data sources, even outside Adeptia Server. As a user executes DashBoard, it displays the results in a graphical format with four different components clubbed together to form a single DashBoard. Monitoring DashBoard provides Adeptia Server user with real-time visibility into the performance of decisive services.

Monitoring DashBoard enables Administratoristrators to analyze discrepancies between expected and actual performance and to monitor compliance with IT operational best practices in real-time.

DashBoard enables the user to represent data in the form of *PIE chart*, *Bar Chart*, *String chart* and *Table chart*.

DashBoard applet represents data using the four components:

- Bar Chart
- String Chart
- Table Chart
- PIE Chart

DashBoard enables business users to view business level information like the number of orders received per customer (Bar Chart), Orders processed per business users (Bar Chart), Orders received per item (PIE Chart), List of new customers today (Table Chart), maximum order of the day (String Chart).

Using DashBoard can be broadly classified into three parts:

- **Creating DashBoard Component**
- **Designing DashBoard**
- **Executing DashBoard**

Prerequisites

- To create any DashBoard activity, *Database Driver* and *Database Info* should be already created and table used for selection of columns for customized DashBoard creation should exist.
- JRE 1.5 needs to be installed on your system to open the Dashboard applet.
- The *Pop-up Blocker* needs to be disabled in the web browser, to open the Dashboard applet. By default, the *Pop-up Blocker* is enabled.

CREATING CUSTOM REPORT

A Custom Report can be used to generate customize report of any set of data. Using Custom Report, you can also generate report for already executed or running process flows.

A custom report activity is created using the custom report template (jrxml file), which is generated by the *iReport* software. This custom report activity is further used to select the process flow and generate reports into PDF, XLS, CSV, TXT and HTML format.

iReport is a third party software which provides a graphical tool to design report template. For detailed information about iReport software refer to following website.

http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/ireport/

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√		

Steps to create a custom report activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Custom Report**. The Manage Custom Report screen is displayed (see Figure 18.1).

Figure 18.1: Manage Custom Report

3. Click the **New** link. The Create Custom Report screen is displayed (see Figure 18.2).

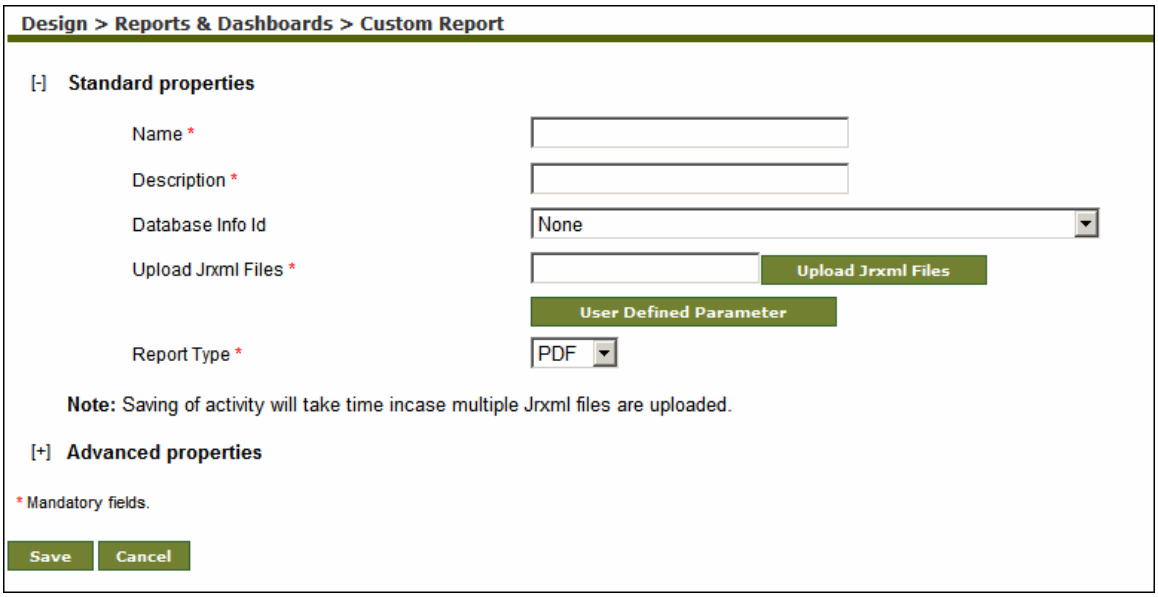


Figure 18.2: Create Custom Report Activity

4. Enter the name of Custom Report in the *Name* field. Then, enter the description for the Custom Report in the *Description* field.

5. Select the Database Info activity which points to the database Server to fetch the data, from the *Database Info ID* drop-down list.

	To learn how to create Database Info activity, refer to *Creating Database Info* section in the *Administrator Guide*.

6. To upload the Jrxml file (custom report template) generated using *iReport* software, click **Upload Jrxml File** button. The Upload Jrxml File screen is displayed (see Figure 18.3).

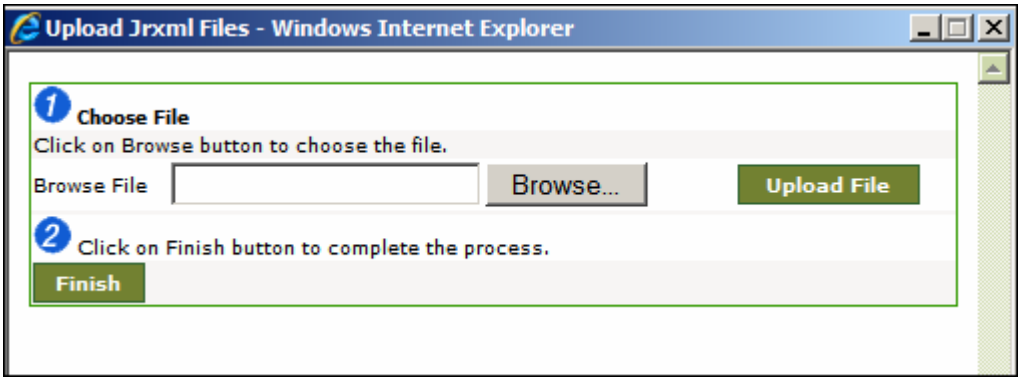


Figure 18.3: Select Jasper File

7. Click **Browse** button and select the Jrxml file. The path of the Jrxml file is displayed in the *Browse* File field (see Figure 18.4).

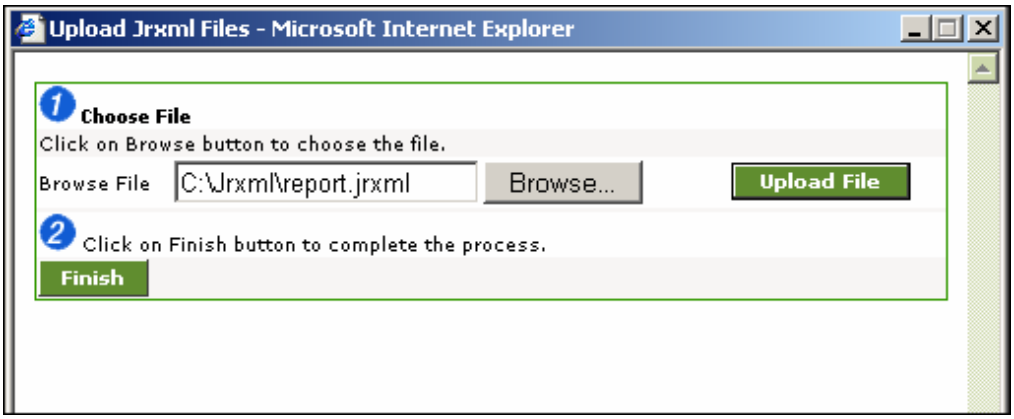


Figure 18.4: Select Jasper File

8. Click **Upload File** button. The file name is displayed in the *File Names* list (see Figure 18.5).

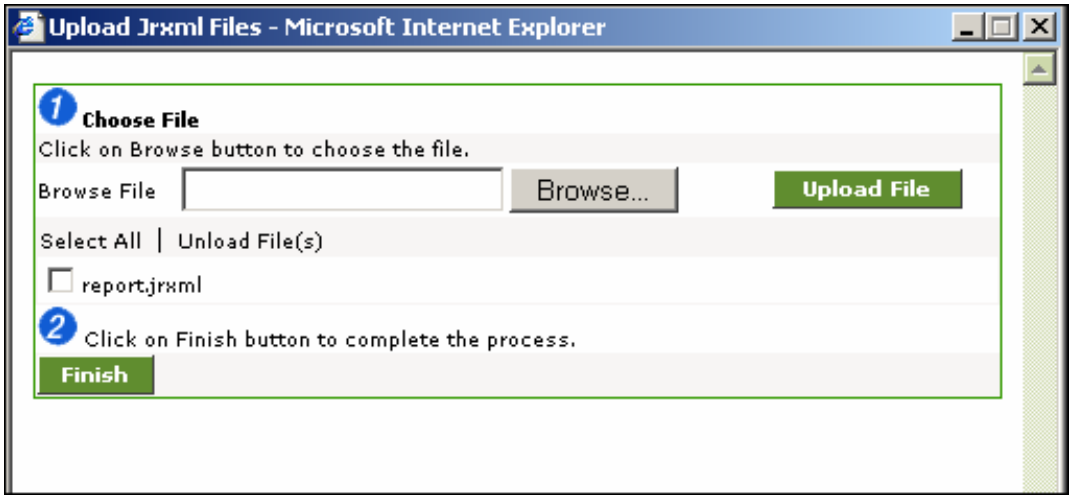


Figure 18.5: Jasper File Uploaded

9. Click **Finish** button. The uploaded *iReport* file is displayed in the *Upload Jrxml files* field (see Figure 18.6).

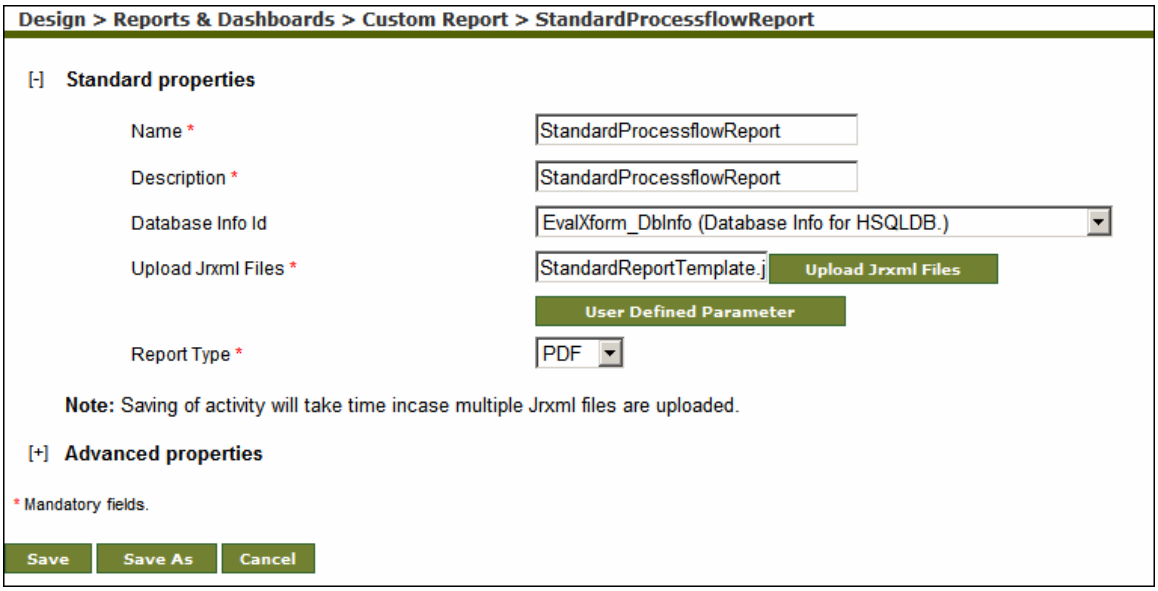


Figure 18.6: Uploaded Jasper Files in Create Custom Report

10. To change value of user defined Parameter defined in the Jrxml file, click **User Defined Parameter** button. The Jasper Report Parameter screen is displayed with list of parameters defined in the Jrxml file (see Figure 18.7).

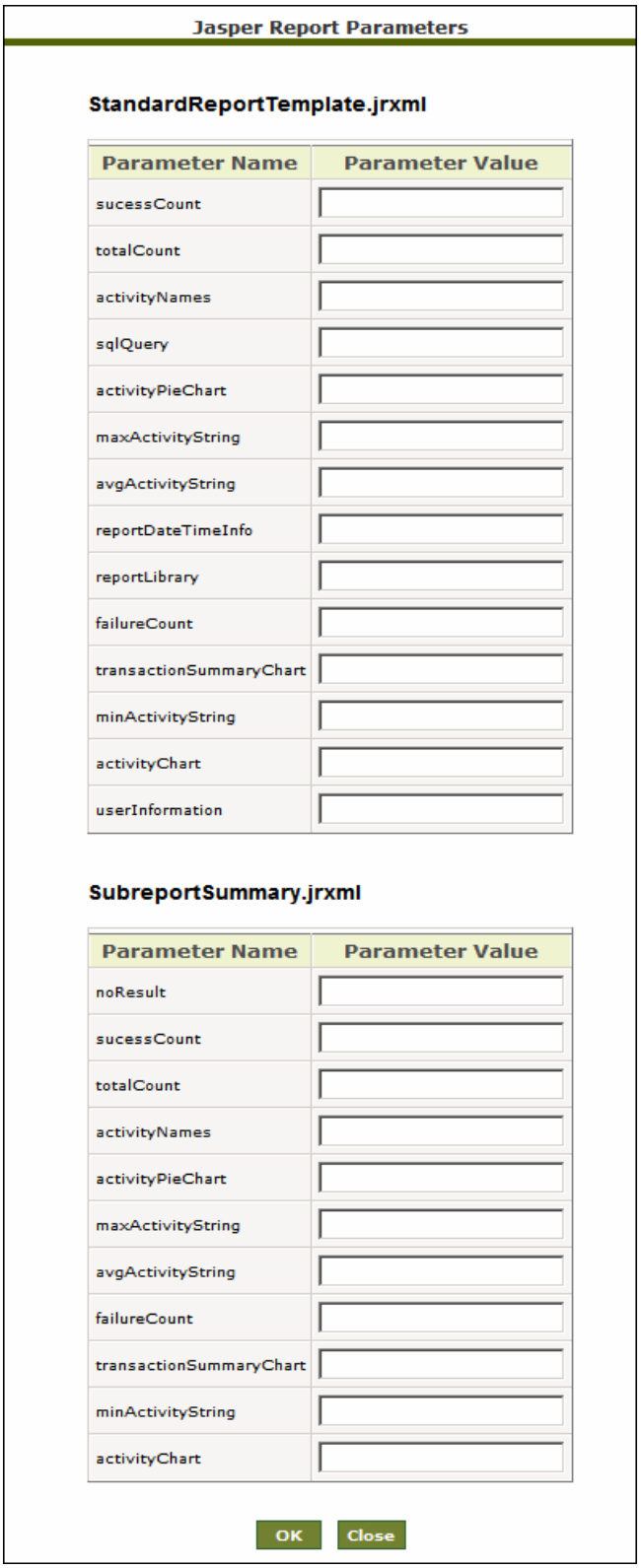

Jasper Report Parameters
StandardReportTemplate.jrxml
Parameter Name
Parameter Value
sucessCount
totalCount
activityNames
sqlQuery
activityPieChart
maxActivityString
avgActivityString
reportDateTimeInfo
reportLibrary
failureCount
transactionSummaryChart
minActivityString
activityChart
userInformation
SubreportSummary.jrxml
Parameter Name
Parameter Value
noResult
sucessCount
totalCount
activityNames
activityPieChart
maxActivityString
avgActivityString
failureCount
transactionSummaryChart
minActivityString
activityChart
OK
Close

Figure 18.7: Define Jasper Report Parameters

11. Enter the value of the parameter in the *Parameter Value* field.

	In *Parameter Value* field you can enter any constant value or any value which is set in context. To enter the constant value, directly enter the value in the *Parameter Value* field. To enter the value from the context, enter the name of the context variable starting and ending with $$. For example $$Eid$$.

12. Click **OK** to close the Jasper Report Parameter screen.
13. Select the format in which the custom report is to be generated, from the *Report Type* drop-down list.
14. Click **Save** button. This displays a screen confirming that the custom report activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the custom report (refer to Figure 3.6).
15. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

16. Click **OK** to save the comments. This displays a screen confirming that the custom report activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	A custom report can be used as a service in a process flow. For details, refer to the section **Using Custom Report in Process Flow**. A custom report activity named *StandardprocessflowReport* comes pre-created with the Adeptia Server. You can use that activity to generate the report for a Process Flow.

CREATING DASHBOARD COMPONENT ACTIVITY

A DashBoard Component activity can be created using one of the following:

- **Bar Chart**
- **String Chart**
- **Table Chart**
- **PIE Chart**

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite

√	√		

Creating Bar Chart Activity

Steps to create Bar Chart

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Component**. The Manage DashBoard Component screen is displayed (see Figure 18.8).

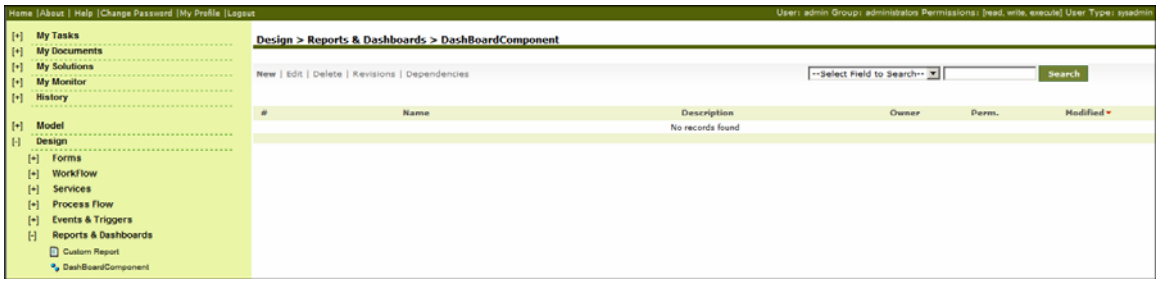


Figure 18.8: Manage Dashboard Component

3. Click the **New** link. The Create DashBoard Component screen is displayed (see Figure 18.9).

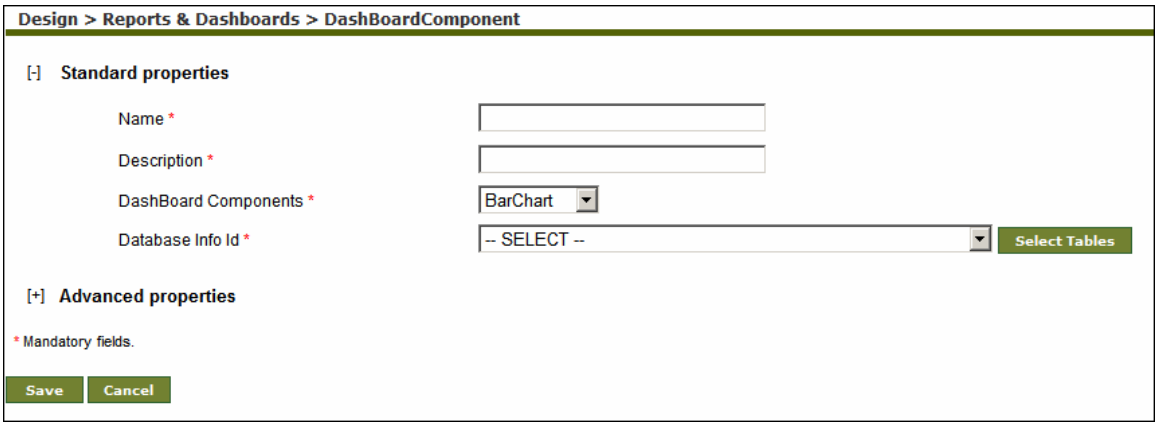


Figure 18.9: Create Dashboard Component

4. Enter the name and description of the new DashBoard Component in the *Name* and *Description* fields respectively.
5. Select Bar Chart from the *DashBoard Component* drop-down list.
6. Select the appropriate Database Info, depending on the database you want to use, from the *Database Info Id* drop-down list.
7. Click **Select Tables** button. The Select Table screen is displayed (see Figure 18.10).

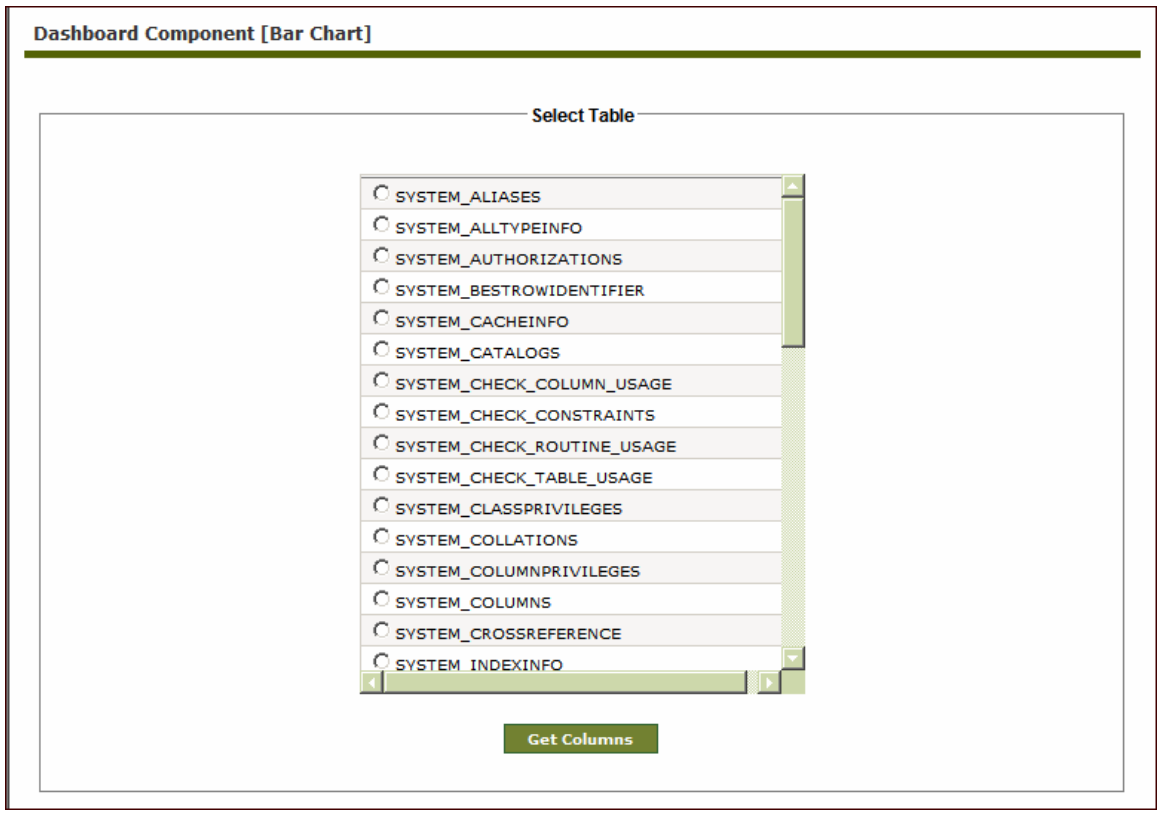


Figure 18.10: Select Table

8. Select the required table and click **Get Columns** button. The Columns in Bar Chart screen is displayed (see Figure 18.11).

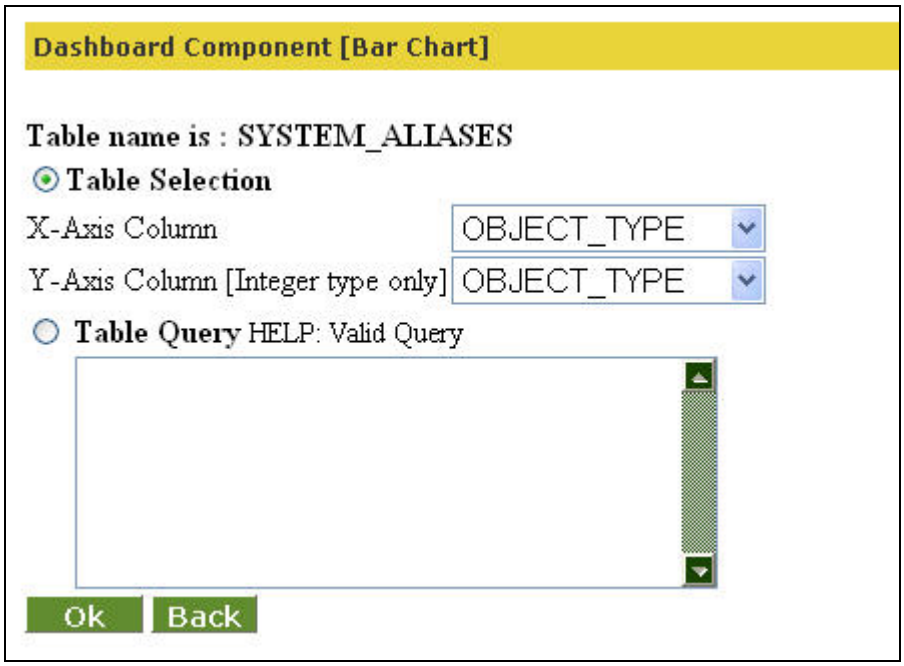


Figure 18.11: Select Columns for Bar Chart

9. Select the X-Axis and Y-Axis components for the Bar Chart from the *X-Axis Column* and *Y-Axis Column [Integer type only]* drop-down lists respectively.

	In Bar Chart Component Y-Axis can take only integer values or values which could be evaluated to an integer e.g. a string with value "2". Any other value will cause erroneous behavior. You can write, SQL query to define X axis and Y axis of Bar chart. To write SQL query, click Table Query radio button and enter your query in the Table Query field.

10. Click **Save** button to save the Bar Chart information. A screen is displayed confirming that the DashBoard Component activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the dashboard component (refer to Figure 3.6).
11. Enter comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

12. Click **OK** to save the comments and return to the Create Dashboard Component screen.
13. Click **Save** button. A screen is displayed confirming that the DashBoard Component activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Creating String Chart Activity

Steps to create String Chart

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Component**. The Manage DashBoard component screen is displayed (refer to Figure 18.8).
3. Click the **New** link. The Create DashBoard Component screen is displayed (refer to Figure 18.9).
4. Enter the name and description for the new DashBoard Component in the *Name* and *Description* fields respectively.
5. Select String Chart from the *DashBoard Components* drop-down list.
6. Select the appropriate Database Info, depending on the database user wants to use, from the *Database Info Id* drop-down list.
7. Click **Select Tables** button. The Select Tables screen is displayed (refer to Figure 18.10).
8. Click **Get Columns** button. The Select Fields screen is displayed (see Figure 18.12).

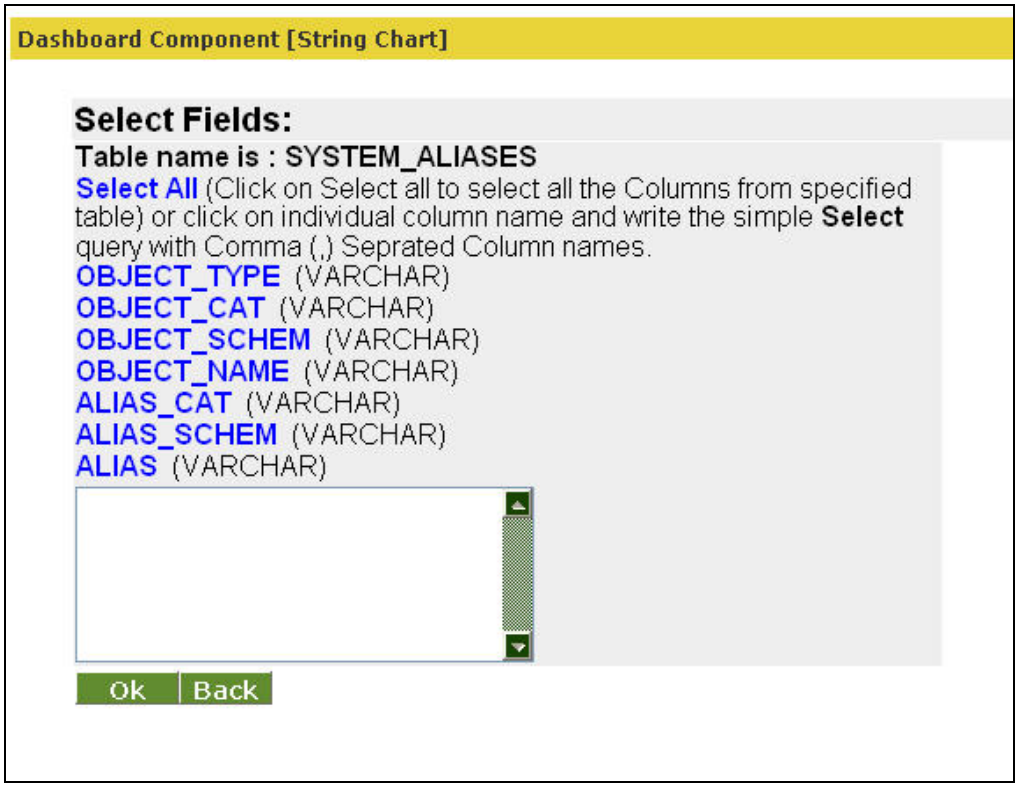


Figure 18.12: Select Fields

9. Click **Select All** to select all the columns from specified table or click individual column name and write the simple Select query with Comma (,) Separated Column names and click **OK** button to return to the Create Dashboard Component screen.

10. Click **Save** button. This displays a screen confirming that the DashBoard Component activity has been created successfully. If the *Comments* property is enabled then clicking **Save** will display a screen where you need to enter comments related to creating the dashboard component (refer to Figure 3.6).
11. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

12. Click **OK** to save the comments. This displays a screen confirming that the DashBoard Component activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Creating Table Chart Activity

Steps to create a Table Chart

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Component**. The Manage DashBoard component screen is displayed (refer to Figure 18.8).
3. Click the **New** link. The Create DashBoard Component screen is displayed (refer to Figure 18.9).
4. Enter the name and description for the new DashBoard Component in the *Name* and *Description* fields respectively.
5. Select Table Chart from the *DashBoard Components* drop-down list.
6. Select the appropriate Database Info, depending on the database user wants to use, from the *Database Info Id* drop-down list.
7. Click **Select Tables** button. The Select Table screen is displayed (refer to Figure 18.10).
8. Click **Get Columns** button. The Select Fields screen is displayed (refer to Figure 18.12)
9. Click **Select All** to select all the columns from specified table or click individual column name and write the simple Select query with Comma (,) Separated Column names and click **OK** button to return to the Create Dashboard Component screen.
10. Click **Save** button. This displays a screen confirming that the DashBoard Component activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the dashboard component (refer to Figure 3.6).
11. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

12. Click **OK** to save the comments. This displays a screen confirming that the DashBoard Component activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

Creating PIE Chart Activity

Steps to create a PIE Chart

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Component**. The Manage DashBoard component screen is displayed (refer to Figure 18.8).
3. Click the **New** link. The Create DashBoard Component screen is displayed (refer to Figure 18.9).
4. Enter the name and description for the new DashBoard Component in the *Name* and *Description* fields respectively.
5. Select PIE Chart from the *DashBoard Components* drop-down list.
6. Select the appropriate Database Info, depending on the database user wants to use, from the *Database Info Id* drop-down list.
7. Click **Select Tables** button. The Select Table screen is displayed (refer to Figure 18.10).
8. Click **Get Columns** button. The Select Columns screen is displayed (refer to Figure 18.11).
9. Select the X-Axis and Y-Axis for the PIE Chart from the *X-Axis Column* and *y-Axis Column [Integer type only]* drop-down lists respectively.

	In PIE Chart Y-Axis Column can take only integer values or values which could be evaluated to an integer e.g. a string with value “2”. Any other value will cause erroneous behavior. You can write, SQL query to define columns of PIE chart. To write SQL query, click Table Query radio button and enter your query in the Table Query field.

10. Click **Save** button. This displays a screen confirming that the DashBoard Component activity has been created successfully. If the *Comments* property is enabled, then clicking **Save** will display a screen where you need to enter comments related to creating the dashboard component (refer to Figure 3.6).
11. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

12. Click **OK** to save the comments and return to the Create Dashboard Component screen.
13. Click **Save** button. A screen is displayed confirming that the DashBoard Component activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

CREATING DASHBOARD DESIGN ACTIVITY

In the Adeptia Suite this feature is available in:

BPM Suite	Workflow Suite	Integration Suite	ETL Suite
√	√		

Steps to create a DashBoard Design activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Design**. The Manage DashBoard Design screen is displayed (see Figure 18.13).

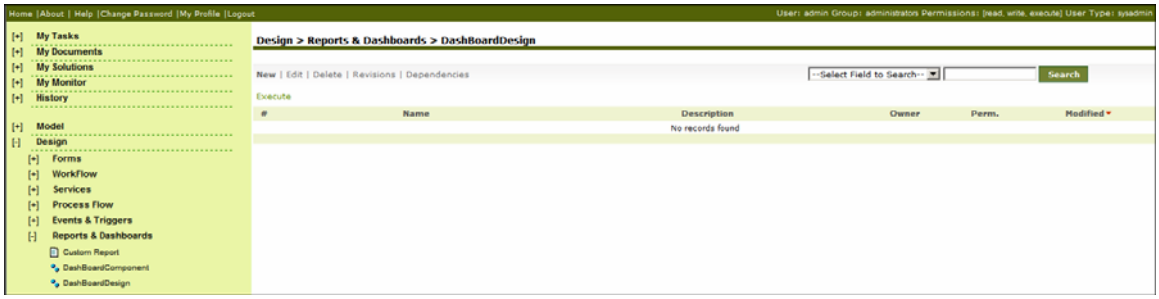


Figure 18.13: Manage Dashboard Design

3. Click the **New** link. The Create DashBoard Design screen is displayed (see Figure 18.14).

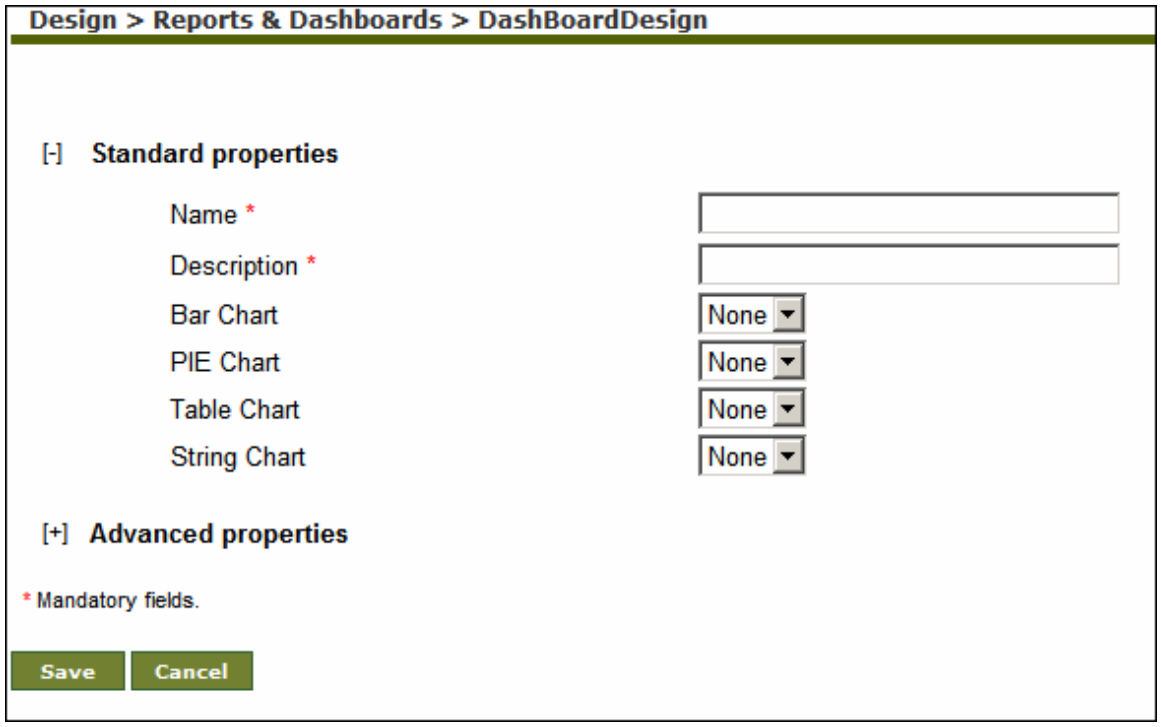


Figure 18.14: Creating Dashboard Design

4. Enter the name and description of the new DashBoard Design in the *Name* and *Description* fields respectively.
5. Then select any of the components from the *Bar Chart, PIE Chart, Table Chart* and *String Chart* drop-down lists. You may select one component from each chart and any number of charts.
6. Click **Save** button. This displays a screen confirming that the DashBoard Design activity has been created successfully. If the *Comments* property is enabled then clicking **Save** will display a screen where you need to enter comments related to creating the dashboard design (refer to Figure 3.6).
7. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

8. Click **OK** to save the comments. This displays a screen confirming that the DashBoard Design activity has been created successfully.

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

EXECUTING DASHBOARD DESIGN

Steps to execute a DashBoard Design activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Reports & Dashboards** to expand the tree and then click **Dashboard Design**. The Manage DashBoard Design screen is displayed (refer to Figure 18.13).
3. Select the radio button adjacent to required dashboard design activity that you want to execute and then click **Execute** link.
4. A DashBoard applet appears displaying the component that was chosen while creating the DashBoard Design activity (see Figure 18.15).

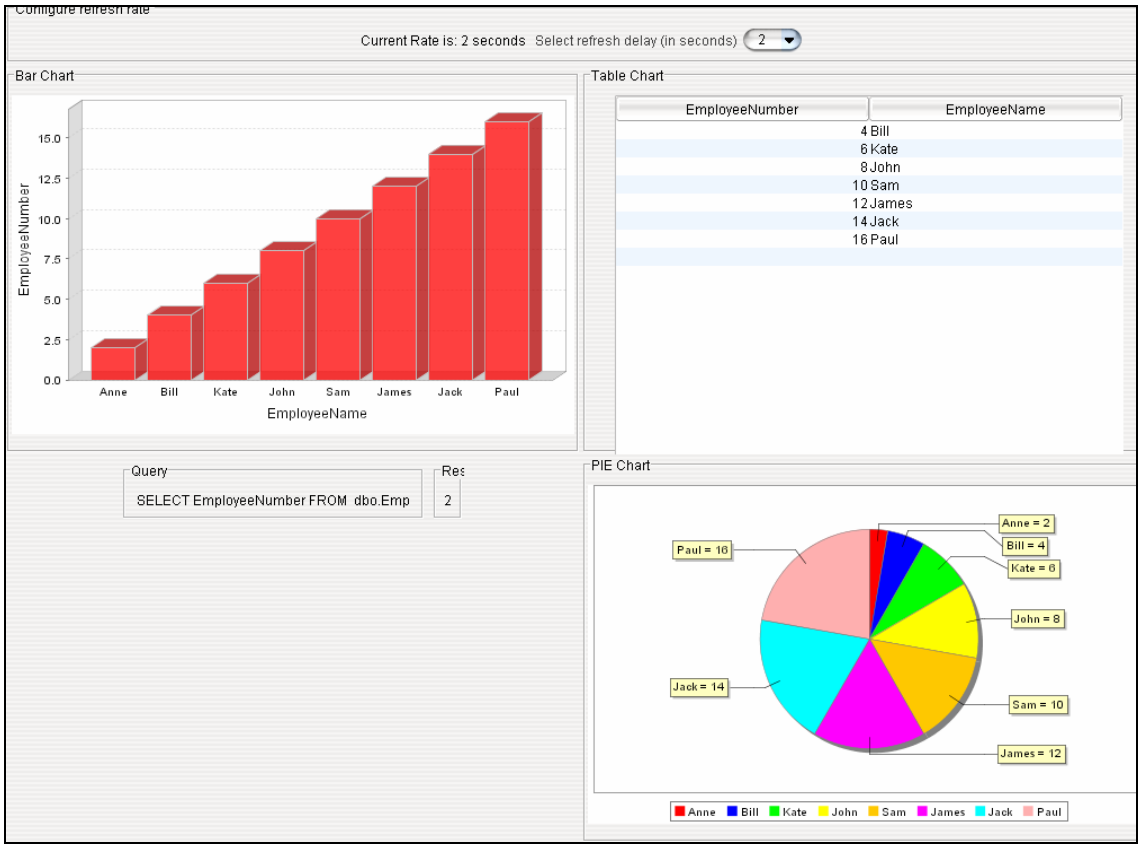


Figure 18.15: View Dashboard

5. To configure the refresh time, select the refresh time from the *Select refresh delay (in seconds)* drop-down list.

19 MANAGING ACTIVITIES

Managing activities involves creating new activities, editing or deleting the existing activities, and saving another instance of those activities. It also includes viewing a revision history of activities and a list of related activities associated with an activity. Creation of various Adeptia Server activities has been covered in previous sections.

This section allows you to perform the following tasks related to an activity:

- Search an Activity
- View an Activity
- Edit an Activity
- Delete an Activity
- View Revision History of an Activity
- View Dependant Activities of an Activity
- Change Advanced Properties of an Activity

For editing, deleting and saving another instance of activities, viewing revision history and list of related activities, refer to the sections below.

SEARCHING AN ACTIVITY

An activity can be searched by its name or description from the list of activities.

Steps to search an activity (for example Text Schema activity)

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (see Figure 19.1).

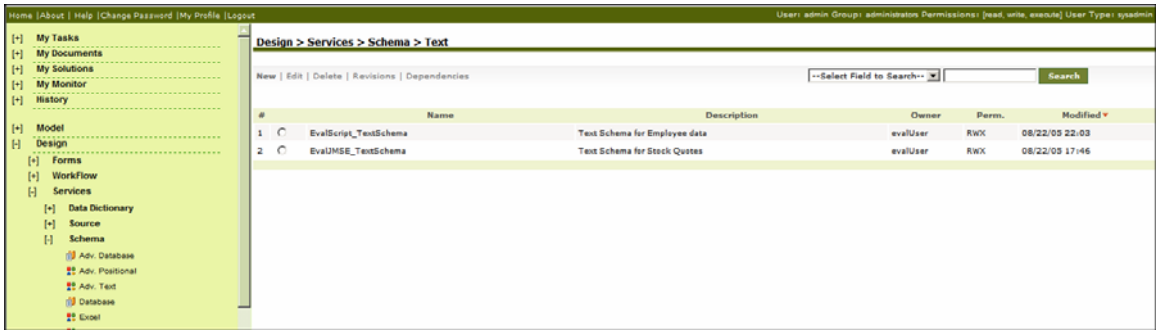


Figure 19.1: Manage Activity

4. Select the field to search from *Select Field* drop-down list.
5. Enter the search criteria in the *Search* field. For example, if Name is selected in *Select Field* drop-down list then enter the name of the *Search* field.

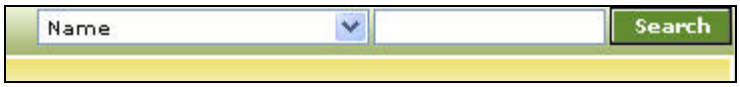


Figure 19.2: Enter Search Criteria

6. Click **Search** button. The searched activity is displayed.

Using Wildcards in Search

If you do not remember the entire name or description of the activity, you can use wildcard characters to search the activity. Once you select the option from the *Select Search Option* drop-down list, you can use a wildcard character in the *Criteria* field. The wildcard characters supported by Adeptia are described in the table below.

Table 19.1: Wildcard characters supported by Adeptia

Wildcard Character	Description	Example
?	Signifies one character in the string	Eval?MSE_TextSchema Searches for a string which has a character between Eval and MSE_TextSchema. Thus it displays EvalJMSE_TextSchema. Eval???E?TextSchema Searches for a string which has three characters after Eval and one character after E. Thus it displays EvalJMSE_TextSchema.

*	Signifies multiple characters in a string	Eval*E_TextSchema Searches for a string which has one or more characters after Eval and before E_TextSchema. Thus, it displays EvalJMSE_Text Schema. Eval * Searches for a string which has one or more characters after Eval. In such a case, it can display more than one strings such as *EvalJMSE_TextSchema, EvalScript_TextSchema* *EvalXForm_ExcelSchema*, etc. You can select the string that you want from this list.

VIEWING ACTIVITY PROPERTIES

Steps to View Activity Properties (for example Text Schema activity)

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Click the activity name. A screen is displayed displaying the properties of the selected activity (see Figure 19.3).

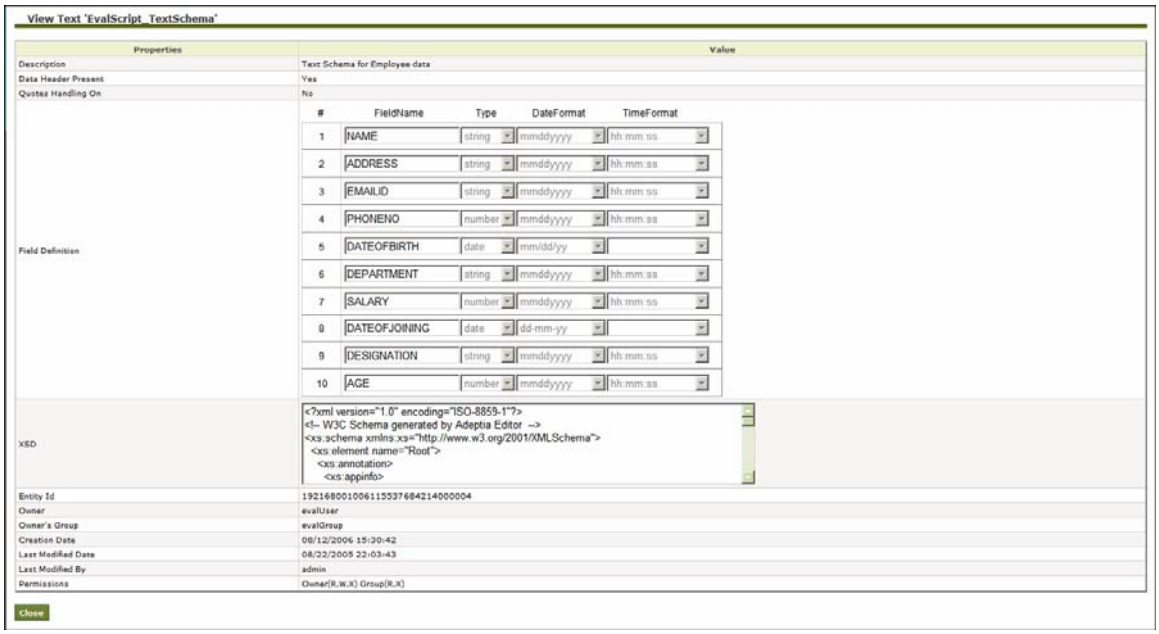


Figure 19.3: View Activity Properties

5. Click **Close** button to return to the Manage Activity screen.

EDITING AN ACTIVITY

Steps to edit an activity (for example Text Schema Activity)

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Select the radio button adjacent to the activity, which you want to edit and then click the **Edit** link. A screen is displayed that allows you to change the properties of the selected activity (see Figure 19.4).

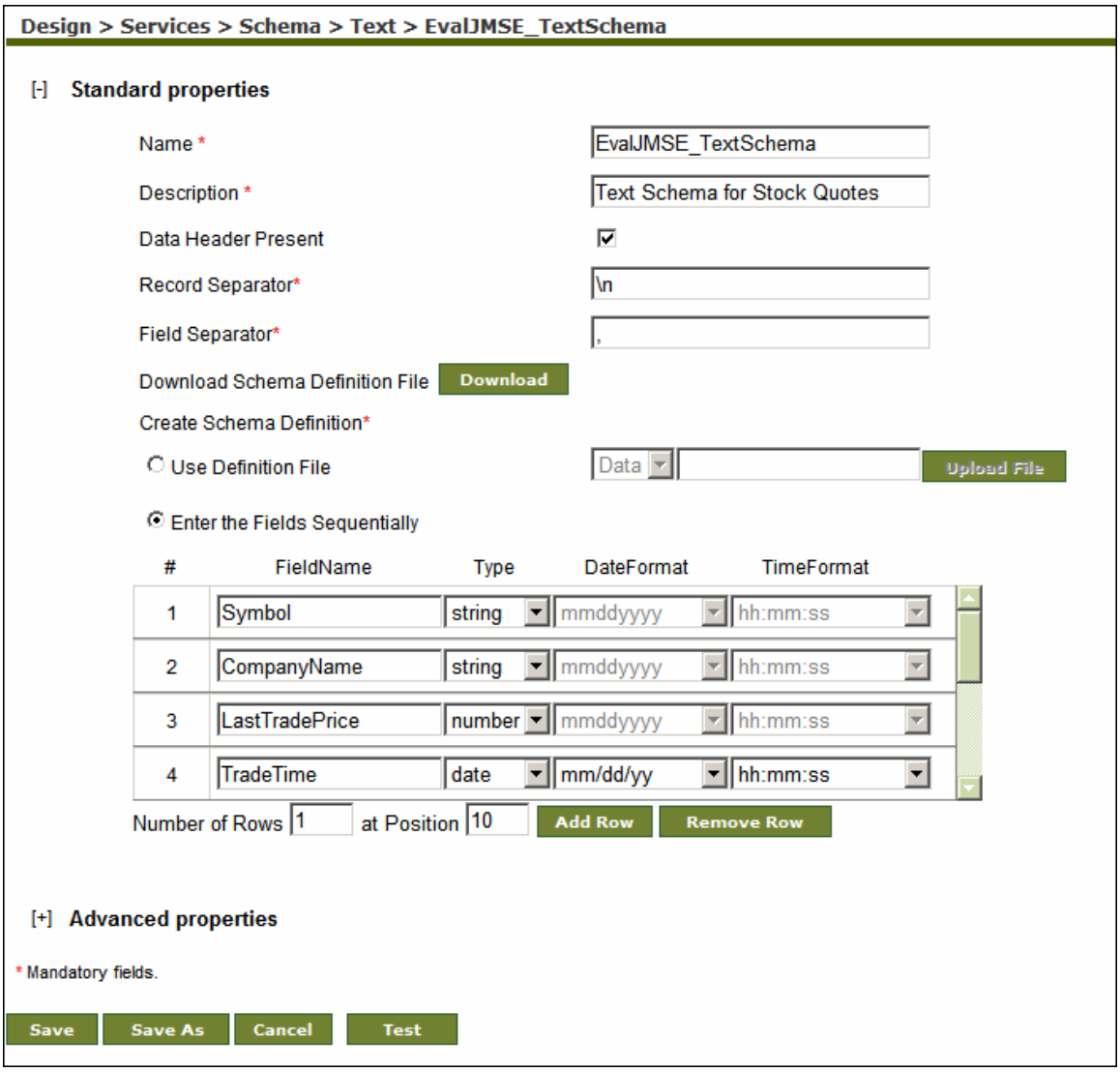


Figure 19.4: Edit Activity

	You can edit only those activities, in which you have write permission.

5. After changing the properties, click **Save** button to save the changes. This displays a screen confirming that the 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 editing the activity (refer to Figure 3.6).
6. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

	By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

	To save another instance of the edited activity with different name, change the activity name in the *Name* field and then click **Save As** button. A screen is displayed confirming that the activity has been created successfully.

	You can verify a source or target activity at design time. For this, click **Test**. This verifies the values in the fields of the activity and checks whether the source or target actually exists in the specified location. The verifications on the fields vary with each activity.

DELETING AN ACTIVITY

Steps to delete an activity

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Select the radio button adjacent to the activity that you want to delete and then click the **Delete** link. A screen is displayed asking for the confirmation to delete an activity (see Figure 19.5).

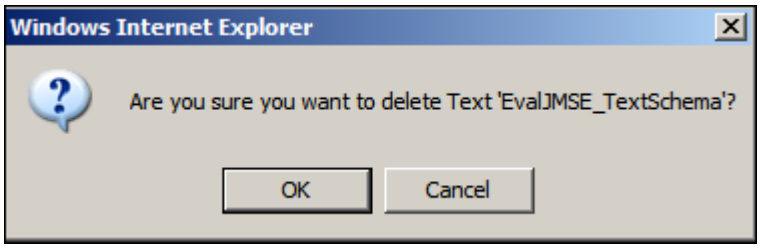


Figure 19.5: Delete an Activity

5. Click **OK** button if you are sure to delete the activity. If the *Comments* property is enabled, then clicking **OK** will display a screen where you need to enter comments related to deleting the activity (refer to Figure 3.6).
6. Enter the comments in the *Add Comments* field.

	The comment should be at least 1 character in length.

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

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

VIEWING REVISION HISTORY

The Revision History of an activity displays a log of actions that have been performed on the activity.

Steps to view Revision History of an activity (for example Text Schema activity)

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Select the radio button adjacent to the activity whose revision log you want to view, and then click the **Revisions** link. A screen is displayed that shows a list of actions performed on the selected activity, in descending order (see Figure 19.6).

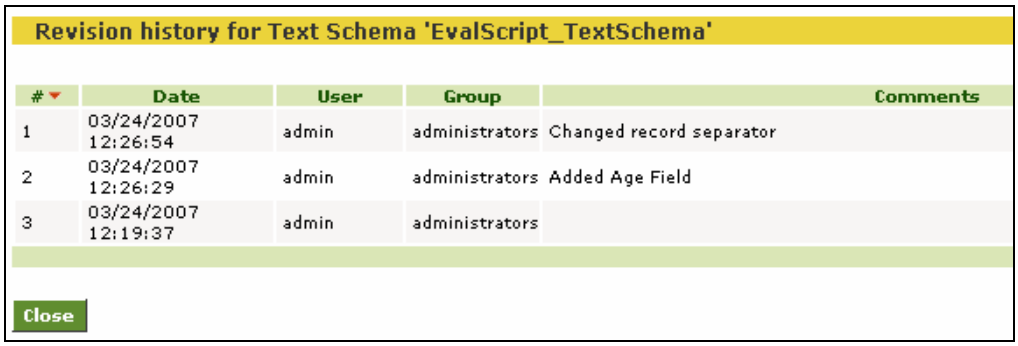


Figure 19.6: View Revision History

5. This screen displays the date and time of the action, the User's name and group of the user who performed the action and also the comments (if any) entered for an action.
6. Click **Back** to go to the Manage Text Schema screen.

The *Revision History* option is dependent on the property "*abpm.appmanagement.logRetainTime*". For example, to check last year's history of activities, you need to change the system date and then click the **Revision History** option.

VIEWING DEPENDENT ACTIVITIES

The Dependent Activities of an activity displays a list of activities that are using or are dependent on the activity. It includes activities that are directly or indirectly dependent upon the selected activity. If an activity is edited, then all the related activities will, in turn, be affected.

For example, there is a database driver namely DBDriver1. This driver is used by the database info DBInfo. The database info DBInfo is further used by the database schema DBSchema. The DBSchema is loaded while mapping source and target elements.

When the mapping activity is used in a process flow, it extracts the elements from the DBSchema. This implies that the mapping activity uses the DBSchema directly and the process flow indirectly. The DBSchema further extracts information from the DBInfo. This implies that the DBSchema uses the DBInfo directly and the process flow indirectly. The DBInfo further locates the driver DBDriver1. This implies that the DBInfo uses the DBDriver1 directly and the process flow indirectly.

The Dependent Activities will display the process flow and a list of all activities that are directly related to the selected activity. In the above example, the related activities for DBDriver1 will display the process flow and DBInfo. The related activities for DBInfo will display the process flow and DBSchema. The related activities of DBSchema will further include the process flow and the mapping activity.

Steps to view Dependent Activities of an activity (for example Text Schema activity)

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Select the radio button adjacent to the activity, whose list of dependant activities you want to view, and then click the **Dependencies** link. A screen is displayed that shows a list of activities directly and indirectly dependent on the selected activity (see Figure 19.7).

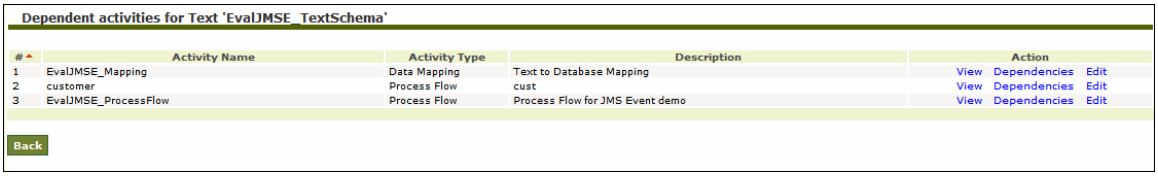


Figure 19.7: View Dependant Activities

5. This screen displays the name and type of the activity that is using this activity. It also displays the activity description. In addition to the **View** link, it also shows a **Dependencies Edit** link, against each activity which further have activities dependent upon them. For example, click **Dependencies Edit** against the activity. A screen is displayed which lists all activities dependent upon the activity (see Figure 19.8).

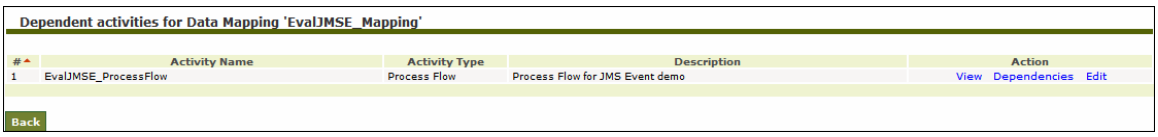


Figure 19.8: View Related Activities

6. Click **Back** to go to the Manage Text Schema screen.

	Related Activities cannot be viewed for Process Flows, WebDAV Folder and the Dashboard. It is also not displayed for the Users and Groups as all activities are dependent upon Users and Groups.

CHANGING ADVANCED PROPERTIES

Advanced properties of an activity show the name of the Owner, Creation Date, Last Modified Date, Modified By and Permissions.

The advanced properties vary for each activity. However, the process of changing advanced properties is similar for all activities. The process of changing advanced properties for a Text schema is explained below.

To change the permission of an activity:

1. In the Adeptia Suite homepage menu, click **[+] Design** to expand the tree. All the items in the **Design** category are displayed.
2. Click **[+] Services** to expand the tree. All the items in the **Services** category are displayed.
3. Click **[+] Schema** to expand the tree, and then click **Text**. The Manage Text Schema screen is displayed (refer to Figure 19.1).
4. Select the radio button adjacent to the activity that you want to edit and then click the **Edit** link. The Edit Text Schema screen is displayed in edit mode. (refer to Figure 19.4).
5. To change the advance properties, click **[+] Advanced Properties**. All the fields of advance properties are displayed (see Figure 19.9).

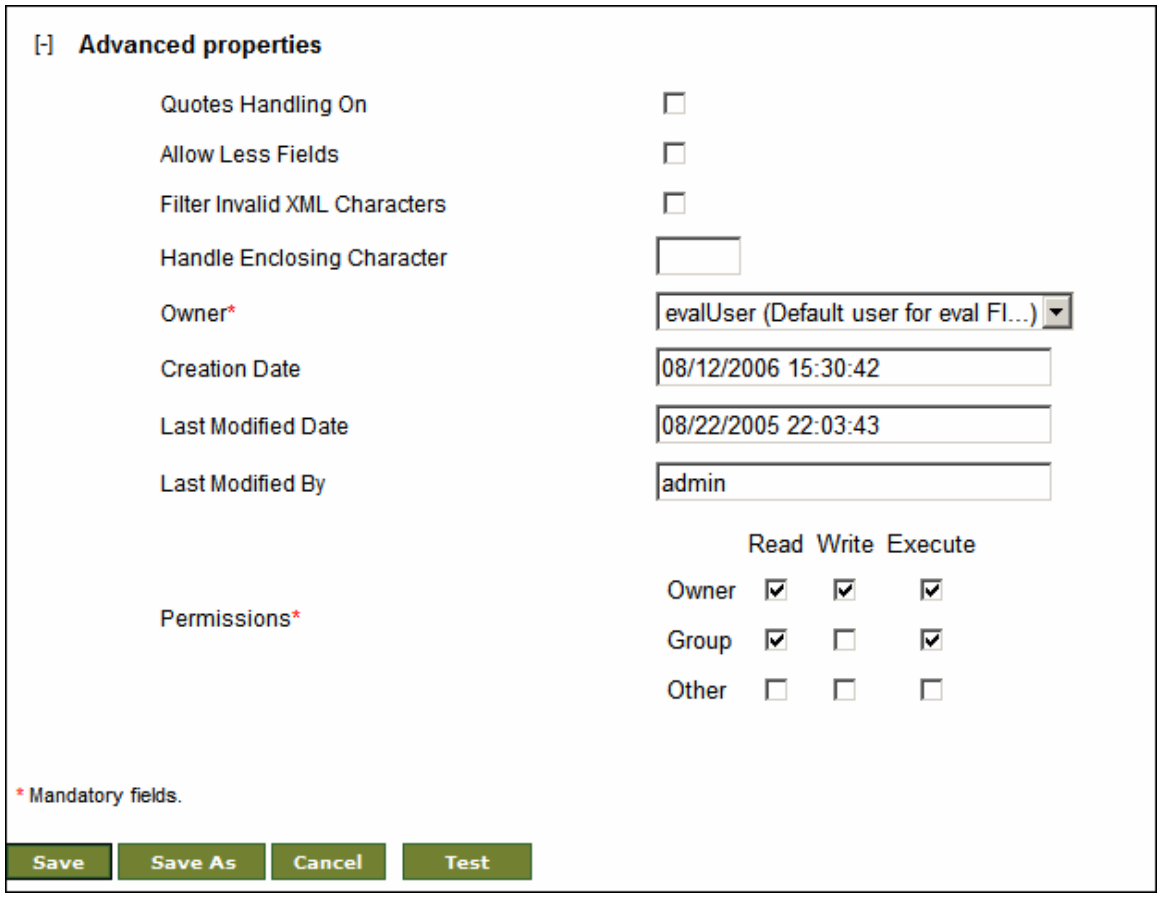


Figure 19.9: View Advanced Properties

6. To change the ownership of the activity, select the owner from the *Owner* drop-down list.

	A user can select another user within its group only as the owner of the activity.

7. The Creation Date, Last Modified Date and Last Modified By fields are non editable. A user cannot change the values of these fields.

8. To change the permissions, check the required checkboxes beside Permissions according to the following tables.

Table 19.2: Owner Permissions

Owner	
Read	Read permission allows the owner user to view the activity. The Read checkbox is pre-selected and cannot be unchecked.
Write	Write permission allows the owner user to Edit the activity.
Execute	Execute permission allows the owner user to Execute the activity.

Table 19.3: Group Permissions

Group	
Read	Read permission allows the other users of the owner's group to view the activity.
Write	Write permission allows the other users of the owner's group to Edit the activity.
Execute	Execute permission allows the other users of the owner's group to Execute the activity.

Table 19.4: Other Permissions

Other	
Read	Read permission allows the users of the other group to view the activity.
Write	Write permission allows the users of the other group to Edit the activity.
Execute	Execute permission allows the users of the other group to Execute the activity.

9. After changing the properties, click **Save** button to save the changes. This displays a screen confirming that the 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 editing the activity (refer to Figure 3.6).
10. Enter the comments in the *Add Comments* field.

The comment should be at least 1 character in length.

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

By default, the *Comments* property is disabled. To enable it, refer to the *Updating System Properties* section in *Administrator* Guide.

20 ABOUT ADEPTIA INC.

Adeptia, an enterprise software company headquartered in Chicago, Illinois, provides a business process integration technology to easily and quickly automate business processes using industry-specific standards. Adeptia's unique product combines business process management with business-to-business integration. Adeptia's reusable and highly scalable technology has been deployed by Fortune 1000 companies. For more information, visit **http://www.adeptia.com** .

Adeptia Headquarters
443 North Clark St, Suite 350
Chicago, IL 60654
USA
Email: info@adeptia.com

Adeptia India R&D Centre
D-74, Sector 63,
Noida, U.P. - 201301
India

21 TABLE OF FIGURES

22 TABLE OF TABLES