SUN SEEBEYOND

eVISION™ STUDIO USER'S GUIDE

Release 5.1.2

Part Number: 819-7427-10

Version 20061006120714

Contents

Chapter 3

Chapter 4

Chapter 8

Chapter 9

Chapter 12

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an overview of the Sun SeeBeyond eVision™ Studio document.

What's in This Chapter

1.1 What's New in This Release

This document includes the following new features and changes:

- Chapter 4 **"Using the Page Layout Designer"** includes a description of the new Preview with Data icon on the Page Layout Designer Toolbar. The tool enables you to view an XML input example as data. This allows you to see exactly how the page will render with real data at runtime.
- Chapter 8 **"Creating Custom Objects"** includes a new section that describes how to create custom objects using the Custom icon in the objects pallette.

1.2 About This Document

1.2.1 What's in This Document

This document contains the following information:

- **Chapter 1 "Introduction"** provides an overview of this document.
- **Chapter 2 "Overview of eVision Studio"** provides an overview of eVision Studio, including how the product is integrated with the Sun Java™ Composite Application Platform Suite.
- **Chapter 3 "Installing eVision Studio"** describes how to install eVision Studio.
- **Chapter 4 "Using the Page Layout Designer"** describes how to use the Page Layout Designer to create eVision Studio web pages, as well as Page Links.
- **Chapter 5 "Object Palettes"** describes the object palettes in the Page Layout Designer.
- **Chapter 6 "Using the Page Flow Designer"** describes how to use the Page Flow Designer to create an eVision Studio web application, which consists of Page Layouts and specialized design elements that are linked together in a Page Flow.
- **Chapter 7 "Creating Charts"** describes how to use the Chart object.
- **Chapter 8 "Creating Custom Objects"** describes how to create custom objects for use in Page Layouts.
- **Chapter 9 "Generating Portlets"** describes how to create portlets that are compliant with Java™ Specification Request (JSR) 168.
- **Chapter 10 "Authentication, Authorization, and Error Handling"** describes how to add authentication to eVision Studio web applications, how to return preconfigured pages for certain errors, and how to authorize users based on roles.
- **Chapter 11 "eVision Studio Tutorial"** guides you through various tasks in the creation and deployment of an eVision Studio web application.
- **Chapter 12 "Troubleshooting"** provides guidance for responding to various problems that you might encounter while using eVision Studio.
- **Appendix A "Method Palette"** describes each method that appears in Method Palette of the Business Rule Designer.
- **Appendix B "Conversational State"** describes the support for conversational state in eVision Studio web applications.

1.2.2 Scope

This document explains how to use eVision Studio to create and deploy user interfaces for composite applications.

1.2.3 Intended Audience

This document is intended for experienced computer users who need to create and deploy user interfaces for composite applications.

Using the style sheet feature requires knowledge of Cascading Style Sheets (CSS).

Creating a custom object requires knowledge of Extensible Markup Language (XML) and JavaServer Pages™ (JSP™) technology.

1.2.4 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	▪ Click **OK**. ▪ On the **File** menu, click **Exit**. ▪ Select the **eGate.sar** file.
`Monospaced`	Command line arguments, code samples; variables are shown in ***bold italic***	`java -jar` ***filename***`.jar`
Blue bold	Hypertext links within document	See **Text Conventions** on page 20
Blue underlined	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.2.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.3 Related Documents

The following documents provide additional information of interest to eVision Studio users:

- *Java Composite Application Platform Suite Installation Guide*
- *Sun SeeBeyond eGate Integrator System Administration Guide*
- *Sun SeeBeyond eGate Integrator User's Guide*
- *Sun SeeBeyond eInsight Business Process Manager User's Guide*
- *Sun SeeBeyond eVision Studio Release Notes*

1.4 Sun Microsystems, Inc. Web Site

The Sun Microsystems web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

http://www.sun.com

1.5 Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

CAPS_docsfeedback@sun.com

Chapter 2

Overview of eVision Studio

This overview includes a summary of features and a description of how the product is integrated with Java CAPS.

What's in This Chapter

2.1 Summary of Features

eVision Studio is a graphical environment that enables you to rapidly create interfaces for composite applications, without the need for advanced programming abilities. When you perform drag-and-drop tasks using the eVision Studio tools, the underlying code is automatically generated.

eVision Studio provides two main tools: the Page Layout Designer and the Page Flow Designer. Both tools are integrated within Enterprise Designer.

The Page Layout Designer is used to create web pages. This tool includes a design canvas, GUI object palettes, a properties window, and a toolbar. Among the objects in the palettes are check box groups, text boxes, hyperlinks, image maps, progress bars, submit buttons, and charts. The properties window enables you to specify a variety of attributes for the GUI objects and the overall web page. The toolbar includes functions such as cut, copy, and paste; left align, right align, top align, and bottom align; and preview.

The Page Flow Designer is used to manage the web pages within a composite application. The web pages are combined with specialized elements to create a Page Flow. Page Flows can include data mappings from one component to the next. These mappings are configured using a business rule designer.

You can use eVision Studio in a variety of ways as a front end to Java CAPS integration solutions. For example:

- In conjunction with Sun SeeBeyond eInsight™ Business Process Manager, eVision Studio enables users to participate in business processes. The users perform workflow tasks with eVision Studio web pages that are tailored for specific organizational roles.
- eVision Studio can provide the login pages in which users enter their user name and password, which are then authenticated.

2.2 eVision Studio and the Sun Java™ Composite Application Platform Suite

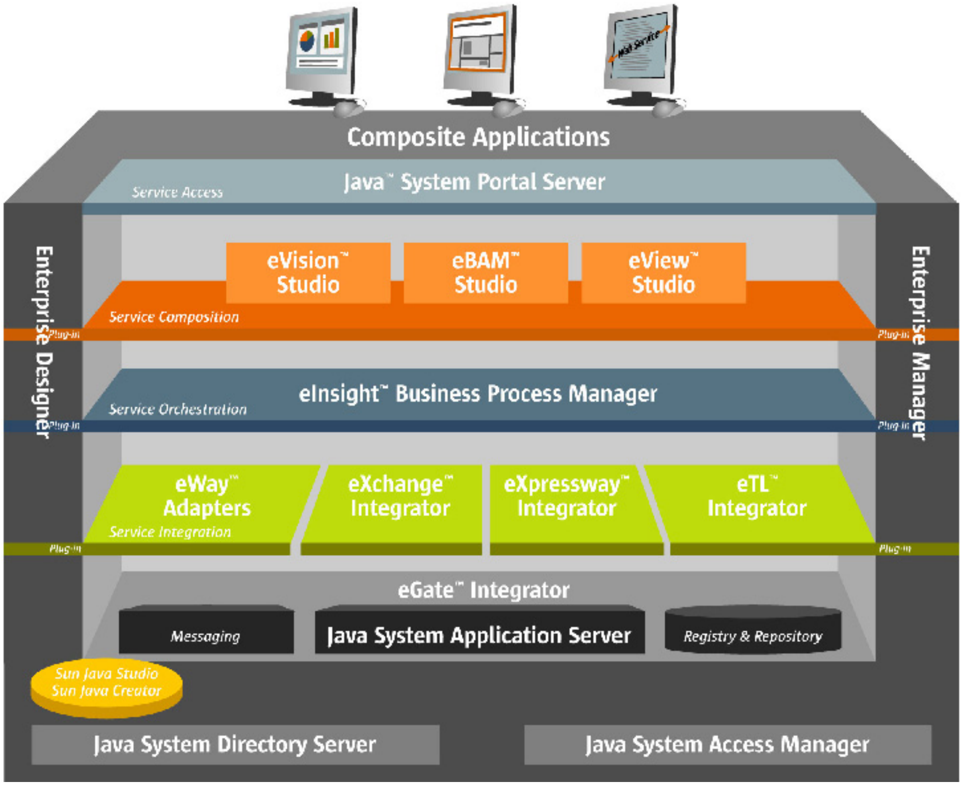
eVision Studio is part of Java CAPS. eVision Studio provides web application design and deployment features and functions to Java CAPS.

2.2.1 Java CAPS Integration

eVision Studio is tightly integrated with Java CAPS and runs as a component within the Java CAPS environment. The Page Layout Designer and Page Flow Designer run as components within Enterprise Designer. eVision Studio web applications are stored in the Sun SeeBeyond Repository.

Figure 1 shows how eVision Studio is integrated with Java CAPS.

Figure 1 Integration with Java CAPS



- eVision Studio runs as a component within Enterprise Designer.
- Page Layouts, Page Flows, GUI objects, and Deployment Profiles are stored in the Sun SeeBeyond Repository.
- The Page Flow Engine, which coordinates all process-related activity of a deployed project, runs in a Sun SeeBeyond Integration Server.

2.3 eVision Studio Overview

eVision Studio applications receive data from and send data to Java CAPS components, enabling users to interact with the data at runtime. Web applications enable the distribution of integrated business processes across the enterprise and allow real-time user interaction with those processes. eVision Studio enables the organization to present a single, unified view of enterprise data and applications to employees, customers, and partners.

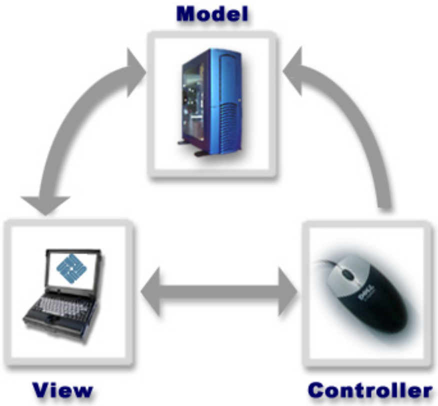
With eVision Studio, you can create personalized views of business information, enabling web application users to interact with running business processes in real time, while working with only the information that they need to see. Web applications can be structured to allow employees to see what tasks are assigned to them, then to interact with and complete the tasks.

2.3.1 MVC Architecture

eVision Studio applications are based on the Model/View/Controller (MVC) architecture. MVC is a software development paradigm that enhances the task of building software systems, particularly those that generate multiple, synchronized presentations of the same data. For example, MVC is ideal for the development of a graphical statistical presentation application that requires simultaneous rendering of the same data in bar, line, and pie chart formats.

The MVC architecture consists of three types of objects: the Model, the View, and the Controller.

Figure 2 Model/View/Controller (MVC) Architecture



The Model object represents the data in a program, which manages behaviors and data within an application. The Model responds to requests for information about its current state (typically requested by the View) and responds to instructions to change its state (typically requested by the Controller).

The View object manages the visual display of the Model data; for example, displaying graphics and text to users in a browser.

The Controller object enables user interaction with the Model data; for example, mouse and keyboard inputs from the user, which instruct the Model and/or View to perform an action.

eVision Studio's application architecture fully supports the MVC paradigm. In an eVision Studio web application, user input, modeling of the external world, and visual feedback are managed by MVC objects, where each object is specialized for its task. For example:

- The Model, represented by the Page Flow, contains the business logic (Object Type Definitions and Collaborations) that interacts with the back-end system applications.
- The View contains the pages created with the JavaServer Pages™ technology ("JSP™ pages") that are generated with the Page Layout Designer.
- The Controller is the web-enabled Page Flow created with the Page Flow Designer. The Controller orchestrates the sequence of pages being sent to the browser in response to user actions.

2.4 Process Overview

The steps for the web application development process are:

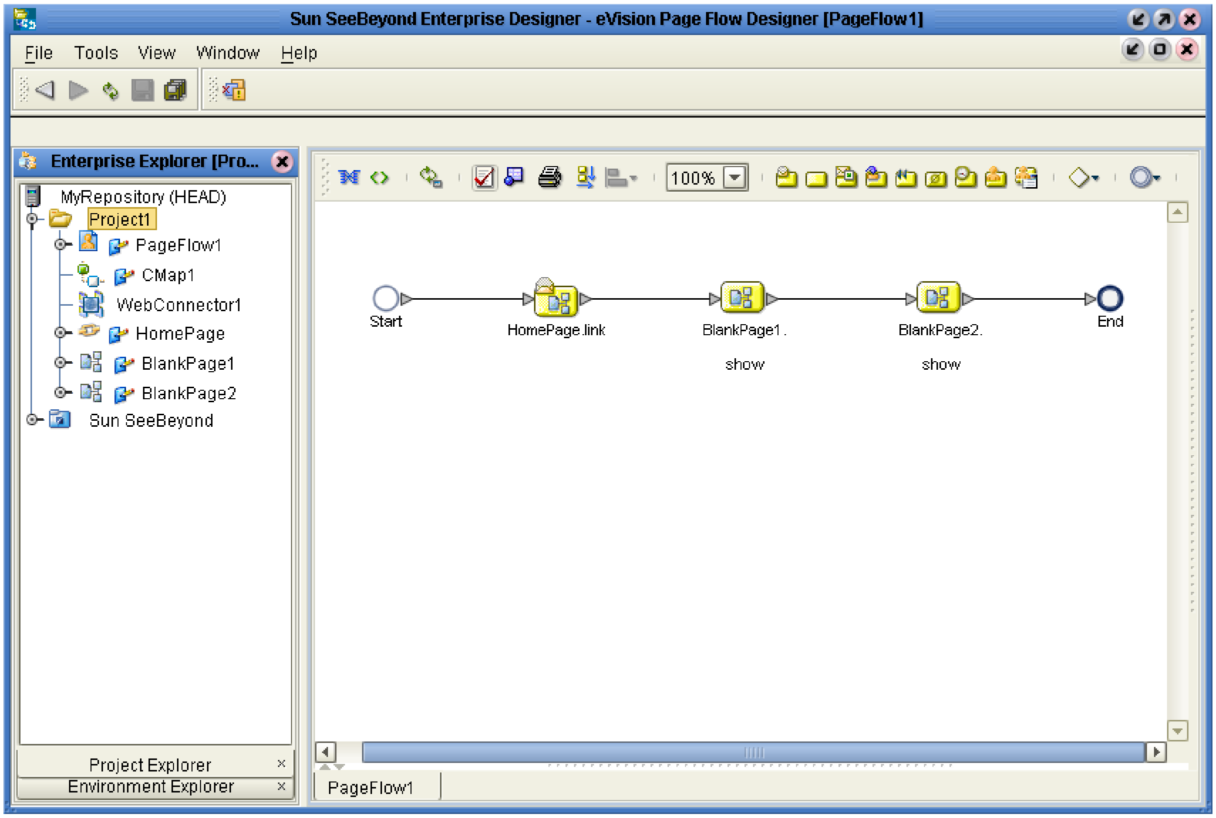
- Page Layout - Using the Page Layout Designer to create web pages with pre-built GUI objects.

- Page Linking - Using the Page Link Wizard to create links to web pages and pass parameters from one page to another.
- Page Flow - Using the Page Flow Designer to connect finished web pages to create a logical flow for the underlying business process.
- Binding and Deployment - Deploying web applications into an Environment for integration with other Java CAPS runtime components.

2.4.1 User Interface Components

eVision Studio leverages the familiar Enterprise Designer interface to create, manage, deploy, and integrate web applications.

Figure 3 eVision Studio in Enterprise Designer



Page Layout Designer

The Page Layout Designer enables you to specify the details of individual pages in the application using familiar drag-and-drop techniques to place GUI objects (web page design objects) onto a design canvas.

The Page Layout Designer provides a comprehensive collection of pre-built GUI objects and a web page editor.

Page Flow Designer

The Page Flow Designer facilitates the flow of web-based business activities. The functions include Page Flow modeling, monitoring, and execution, as well as the ability to analyze how data messages flow from page to page.

The Page Flow Designer enables you to lay out the user workflow of a web application by modeling the high-level, logical, page-by-page flow that users will follow through the web application to complete a given task (the Page Flow). You can lay out a web application's page flow using the Page Flow Designer's set of graphical modeling tools. At runtime, the Page Flow drives the display of eVision Studio web pages and orchestrates the interactions with back-end systems.

Through the deployment of web pages that are developed with the Page Layout Designer and Page Flow Designer, you can enable enterprise-wide, internal and external access to web applications across an intranet or the Internet.

2.4.2 Page Flow Engine

The Page Flow Engine orchestrates the system responses to the execution of web page component code. At runtime, the Page Flow Engine executes Page Links, receives and processes user input, and, based on human interaction, moves the viewer from page to page until the underlying process is complete.

2.5 UTF-8 Support

eVision Studio provides support for the UTF-8 encoding scheme during both design time and runtime. Therefore, an eVision Studio web application can include any character set that is supported by UTF-8, including Asian language character sets.

Chapter 3

Installing eVision Studio

In the installation procedure for eVision Studio, you upload files to the Repository and then update Enterprise Designer.

What's in This Chapter

3.1 Installing eVision Studio from Media

The installation procedure for eVision Studio is similar to the installation procedure for other Java CAPS products. The *Java Composite Application Platform Suite Installation Guide* contains general product installation instructions.

You can install eVision Studio at the same time as eGate Integrator or at a later time.

If you want to install eVision Studio at the same time as eGate Integrator, then follow the instructions in the *Java Composite Application Platform Suite Installation Guide*. After uploading the **eGate.sar** file to the Repository, upload the **eVision.sar** file.

If you want to install eVision Studio at a later time, then follow the instructions in this section.

Before you begin, ensure that the Repository is running.

3.1.1 Uploading eVision Studio to the Repository

eVision Studio consists of two **.sar** files:

- The **eVision.sar** file contains the product software.
- The **eVisionDocs.sar** file contains the documentation and sample files.

Before you can install the **eVision.sar** file, the **eInsight.sar** file must already be installed. If you try to upload the **eVision.sar** file and the **eInsight.sar** file has not been installed, then the Installer prompts you to upload the **eInsight.sar** file.

The *Java Composite Application Platform Suite Installation Guide* lists the locations of these files.

The following procedure must be performed by the **Administrator** user, or by another Repository user that has the **administration** role.

To upload the product software to the Repository

1. Start Internet Explorer.
2. In the **Address** field, enter the following URL:

   ```
   http://hostname:portnumber
   ```

   Set the host name to the TCP/IP host name of the computer where the Repository is installed. Set the port number to the base port number of the Repository.

   The login window of the Installer appears.
3. Enter your Repository user name and password.
4. Click **Login**.

   The **Administration** page of the Installer appears.
5. Click the **Click to install additional products** link.

   The list of product categories appears.
6. Expand the **Core Product** node.
7. Locate the **eVision** row and select the check box.
8. At the bottom of the page, click **Next**.

   The **Selecting Files to Install** window appears.
9. If eInsight has not been installed, then you are prompted to install eInsight. In the **eInsight** field, do one of the following, and then click **Next**:
   - Enter the fully qualified name of the **eInsight.sar** file.
   - Click **Browse** to select the **eInsight.sar** file.
10. In the **eVision** field, do one of the following, and then click **Next**:
    - Enter the fully qualified name of the **eVision.sar** file.
    - Click **Browse** to select the **eVision.sar** file.

    The **Installation Status** window indicates the status of the upload.

    When the installation is finished, a green check mark appears.
11. Click the **Administration** page again.

    eVision Studio now appears in the list of products that have been installed.

To upload the documentation and sample files to the Repository

1. From the **Administration** page of the Installer, click the **Click to install additional products** link.
2. Expand the **Documentation** node.
3. Locate the **eVisionDocs** row and select the check box.

4 At the bottom of the page, click **Next**.

The **Selecting Files to Install** window appears.

5 In the **eVisionDocs** field, do one of the following, and then click **Next**:

- Enter the fully qualified name of the **eVisionDocs.sar** file.
- Click **Browse** to select the **eVisionDocs.sar** file.

The **Installation Status** window indicates the status of the upload. When the installation is finished, a green check mark appears.

6 Click the **Documentation** page. You can now access the documentation and sample files.

3.1.2 Adding eVision Studio Components to Enterprise Designer

The following procedure adds eVision Studio components to Enterprise Designer.

To add eVision Studio components to Enterprise Designer

1. On each client computer where Enterprise Designer is installed, start Enterprise Designer.
2. On the **Tools** menu, click **Update Center**.

   The Update Center Wizard appears.
3. Click **Next**.

   Step 2 of the wizard appears.
4. Click the **Add All** button.

   The items in the **Available Updates and New Modules** box move to the **Include in Install** box.
5. Click **Next**.

   The **License Agreement** window appears.
6. Click **Accept**.

   Progress bars indicate the status of the download.

   When the download is complete, the message **Done** appears below the progress bars.
7. Click **Next**.

   The Update Center Wizard displays a list of modules.
8. Click **Finish**.

   The **Restart the IDE** dialog box appears.
9. Click **OK**.

   When you log into Enterprise Designer again, you can start using eVision Studio.

3.2 Installing the eVision Enterprise Manager Plug-In

To monitor eVision Studio Projects at runtime, you must first install the eVision Enterprise Manager Plug-In application. You perform this task from Sun SeeBeyond Enterprise Manager.

The following procedure must be performed by an Enterprise Manager user that has the **Manager** role.

To install the eVision Enterprise Manager Plug-In

1. Ensure that the Repository is running.
2. In the Explorer panel of Enterprise Manager, click the **Configuration** icon.
3. Click the **Web Applications Manager** tab.
4. Click the **Auto-Install from Repository** tab.
5. Connect to the Repository.
6. In the row that lists the eVision Enterprise Manager Plug-In application, select the check box.
7. Click **Install**.

   After the installation process is complete, the **Results** area indicates whether the installation succeeded.

3.3 Upgrading eVision Studio from Release 5.1.1 to 5.1.2

To upgrade eVision Studio from release 5.1.1 to 5.1.2, follow the instructions in the *Java Composite Application Platform Suite Installation Guide*.

Note: *You do not need to export and import Projects.*

3.4 IBM AIX Configuration Changes

If you are using a Repository that is running IBM AIX, you must perform the following procedure in order to monitor eVision Studio Projects in Enterprise Manager.

To make the configuration changes on an IBM AIX Repository

1. Go to the computer on which the Repository is installed.
2. If the Repository is running, shut it down.
3. Set the **DISPLAY** environment variable to **somehost:0.0**, where **somehost** is the hostname or IP address of one of the computers that will be using Enterprise Manager. The UNIX user that starts the Repository must perform this step.

   Here is an example for the **csh** shell:

   ```
   setenv DISPLAY 10.1.192.13:0.0
   ```

   Here is an example for the **sh** shell:

   ```
   DISPLAY=10.1.192.13:0.0
   export DISPLAY
   ```

4. Open the **startserver.sh** file in the **Sun_JavaCAPS_install_dir/repository** directory.
5. Add the following command to the **JAVA_OPTS** environment variable:

   ```
   -Djava.awt.headless=true
   ```

6. Save the file.
7. Start the Repository.

Chapter 4

Using the Page Layout Designer

You use the Page Layout Designer to create eVision Studio web pages, as well as Page Links.

What's in This Chapter

4.1 Page Layout Designer Basics

Through the use of drag-and-drop techniques and text-based property sheets, the Page Layout Designer enables you to add graphics, text, and programmatic content to the web pages in your application.

You drag pre-built or custom objects from the object palettes and position them on the design canvas. An object's property sheet enables you to specify attributes, such as the object's internal name and the display text.

4.1.1 Adding a Page Layout to a Project

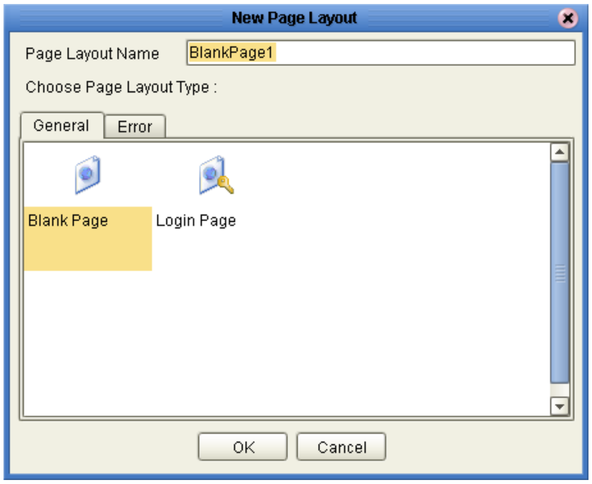
You can add one or more Page Layouts to a Project.

To add a Page Layout to a Project

1. In the Project Explorer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page Layout**.

The **New Page Layout** dialog box appears.

Figure 4 New Page Layout Dialog Box


New Page Layout

Page Layout Name: BlankPage1

Choose Page Layout Type :

3 Click one of the Page Layout types.

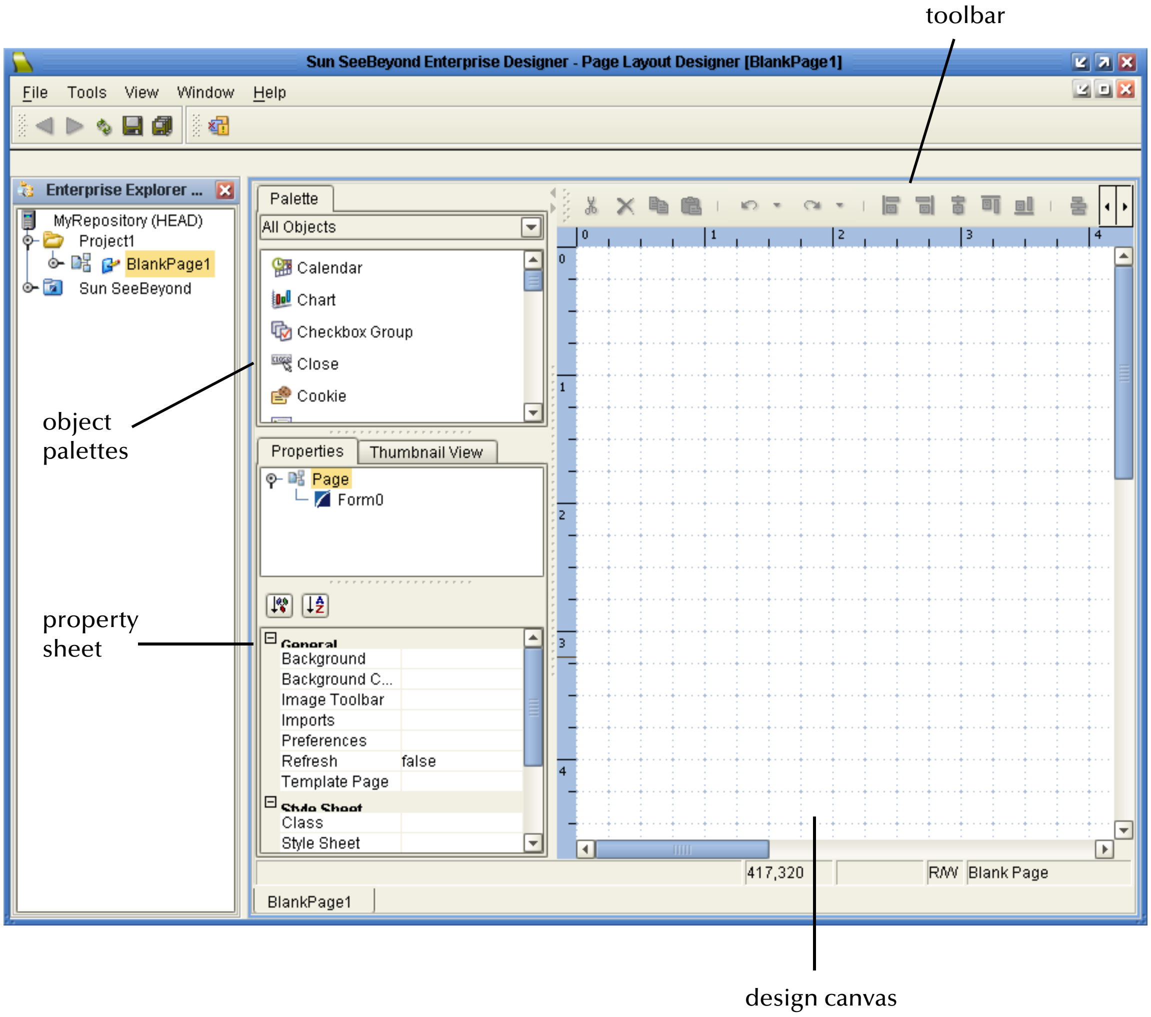
Note: Chapter 10 **"Authentication, Authorization, and Error Handling"** *describes the Login Page, Access Denied Error, Login Error, No Such Resource Error, and Internal Server Error types.*

4 In the **Page Layout Name** field, you can change the default name of Page Layout. The characters must be alphabetic or numeric. You cannot enter a name that is already used by another Page Layout in the same Project.

5 Click **OK**.

The Page Layout is added to the Project Explorer, and the Page Layout Designer appears with a blank design canvas.

Figure 5 Page Layout Designer



For information about the toolbar, see **“Page Layout Designer Toolbar” on page 37**.

For information about the object palettes, see **“Page Layout Objects” on page 38**.

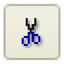
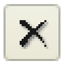
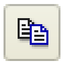
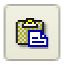
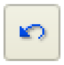
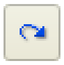
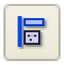
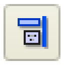
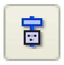
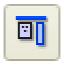
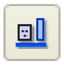
For information about the property sheet, see **“Object Properties” on page 40**.

For information about various features of the design canvas, see **“Design Canvas Features” on page 43**.

4.1.2 Page Layout Designer Toolbar

The Page Layout Designer toolbar enables you to manipulate the objects that you place on the design canvas. Table 2 describes each tool.

Table 2 Page Layout Designer Tools

Tool	Name	Description
	Cut Components	Removes the selected object from the canvas and places it on the clipboard.
	Delete Components	Deletes an object permanently.
	Copy Components	Copies the selected object and places it on the clipboard.
	Paste Components	Pastes an object from the clipboard onto the canvas.
	Undo	Rolls back the most recent action. If you click the arrow to the right of the icon, you can select the number of actions to roll back.
	Redo	Reverses the most recent rollback. If you click the arrow to the right of the icon, you can select the number of rollbacks to reverse.
	Left Align	Aligns two or more selected objects with the left border of the primary selected component (the component that has the yellow selection handles). **Note:** If you want to change which component is the primary selected component, then click the desired component.
	Right Align	Aligns two or more selected objects with the right border of the primary selected component (the component that has the yellow selection handles).
	Center Align	Aligns two or more selected objects with the center of the canvas (the center of the web page).
	Top Align	Aligns two or more selected objects with the top border of the primary selected component (the component that has the yellow selection handles).
	Bottom Align	Aligns two or more selected objects with the bottom border of the primary selected component (the component that has the yellow selection handles).

Tool	Name	Description
	Vertical Spacing	Creates equal vertical spacing between objects in a group of three or more, based on an averaging algorithm.
	Horizontal Spacing	Creates equal horizontal spacing between objects in a group of three or more, based on an averaging algorithm.
	Resize Left to Align	Resizes the left boundaries of the selected objects to match the object that is furthest to the left.
	Resize Right to Align	Resizes the right boundaries of the selected objects to match the object that is furthest to the right.
	Resize Top to Align	Resizes the top boundaries of the selected objects to match the object that is furthest up.
	Resize Bottom to Align	Resizes the bottom boundaries of the selected objects to match the object that is furthest down.
	Install Custom Object	Enables you to import the files for a custom object. For more information, see Chapter 8 **“Creating Custom Objects”**.
	Replace Properties	Enables you to set property values for multiple selected objects at the same time.
	Preview	Previews the Page Layout in your browser.
	Preview with Data	Enables you to preview an XML input example as data, to see how it would appear at runtime. The XML must be in the same XSD format that the page is expecting. This allows you to see exactly how the page will render with real data at runtime.
	Print	Prints the Page Layout.

4.1.3 Page Layout Objects

The Page Layout Designer provides standard web design objects, such as check boxes, text boxes, drop-down lists, graphics containers, horizontal lines, and submit buttons. In addition, you can create custom objects.

When you place an object on the canvas, the object’s property sheet enables you to change the characteristics of the object.

In the Page Layout Designer, the upper left pane contains the object palettes. The lower left pane contains the property sheets.

By default, the **All Objects** palette appears. This palette contains all of the predefined and custom objects. To display another palette, select the appropriate option from the drop-down list. Figure 6 shows the **Custom Objects** palette.

Figure 6 Custom Objects Palette

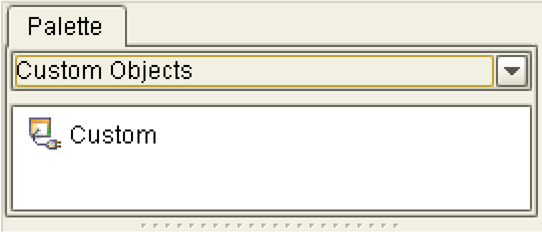


Figure 7 shows the **Extension Objects** palette.

Figure 7 Extension Objects Palette

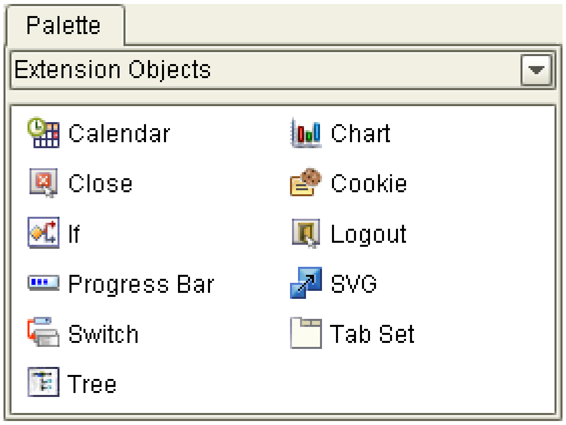


Figure 8 shows the **Form Objects** palette.

Figure 8 Form Objects Palette

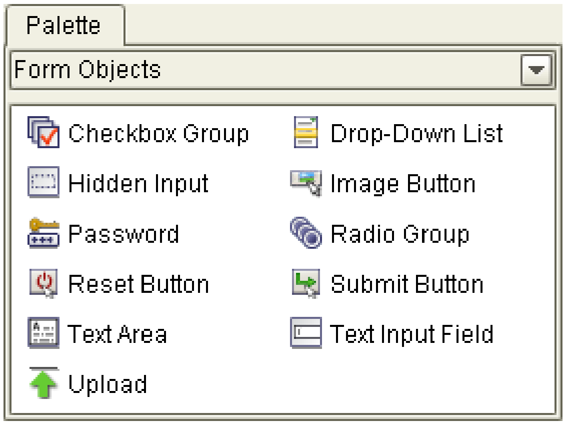
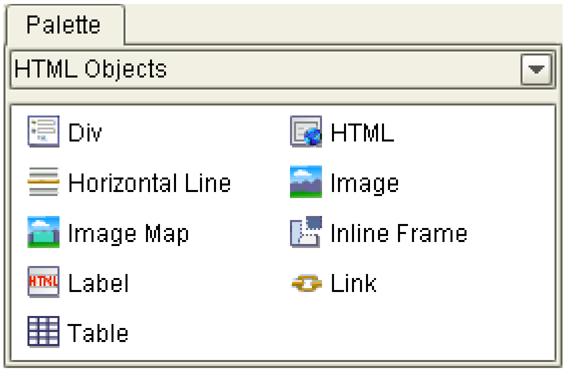


Figure 9 shows the **HTML Objects** palette.

Figure 9 HTML Objects Palette

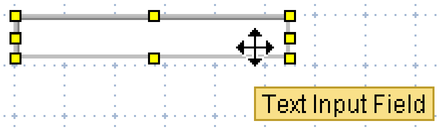


To add an object to the canvas, select an object in a palette, drag the object onto the canvas, and release the mouse button.

After you place an object on the canvas, you can customize its functional and visual presentation properties in the **Properties** tab.

When you place the mouse pointer over an object on the canvas, a tooltip indicates what type of object it is.

Figure 10 Identifying the Object Type



For information about the **Template Objects** palette, see **"Creating Template Objects" on page 66**.

4.1.4 Object Properties

The **Properties** tab displays the property sheet of an object or of the overall Page Layout. The property sheet enables you to customize the behavior of the object or Page Layout.

Some of the properties are required, while other properties are optional.

Each object must have a logical name (the **LName** property). The Page Flow Designer uses the **LName** property as the identifier for Page Layout objects. Changing the default value of the **LName** property to a more descriptive value makes it easier for you to identify the object in the Page Flow Designer.

eVision Studio requires that the first character of the **LName** property be capitalized. If you enter a value that begins with a lowercase character, then eVision Studio capitalizes the character. For example, **myValue** is changed to **MyValue**.

A description of each property is provided in two ways:

- In a property description box at the bottom of the **Properties** tab
- As a tooltip when you place the mouse pointer over a property name

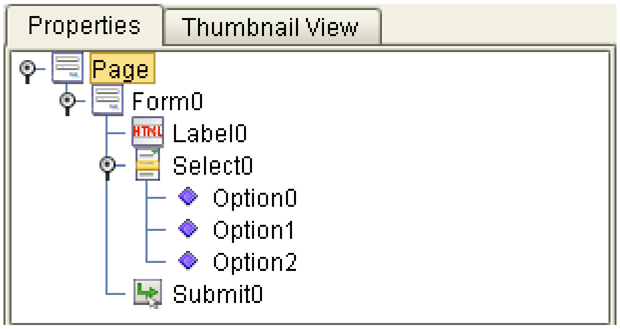
To show or hide the property description box, right-click the design canvas and then click **Show Property Description**.

Property Types

Properties are grouped into *property types*.

The upper portion of the **Properties** tab contains a tree structure that shows the hierarchy of property types.

Figure 11 Property Tree Hierarchy



The property type for the overall Page Layout is **Page**.

Each object has one or more property types. The top-level property type for each object is named after the value of the **LName** property. Figure 11 shows a Page Layout that contains a **Label** object, a **Drop-Down List** object, and a **Submit Button** object.

Property Categories

By default, the list of properties is divided into categories. For example, the categories for the **Page** property type are **General**, **Style Sheet**, **Accessibility**, and **JavaScript**.

You can specify whether to display the list of properties in category or alphabetical format. For category format, click the **Categorize** icon. For alphabetical format, click the **Alphabetize** icon.

Property Names

In the **Properties** tab, some of the property names appear in blue. The blue color indicates that these properties also appear in the mapper portion of the Page Flow Designer.

In the **Properties** tab, the internal property names are changed to more closely resemble the English language. For example, the **bgcolor** property appears as **Background Color**. The Page Flow Designer displays the internal property names.

Property Values

To set the value for a property, select the property name in the left column and then do one of the following in the right column:

- Enter a value.
- Select an option from the drop-down list.
- Click the **Command** button (**...**). In the dialog box that appears, enter text or specify options.

You can restore the original value of most properties by clicking the blue, circular arrow in the right column.

Setting Property Values for Multiple Selected Objects

The Page Layout Designer allows you to set a property value for multiple selected objects at the same time.

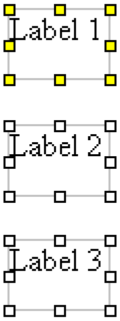
This feature works with container objects. For example, if a **Div** object contains another **Div** object, then the property value is set for both **Div** objects.

You can use this feature with the **Table** object to set a property value for all of the table rows, or for all of the table cells within a table row.

To set a property value for multiple selected objects

1. Select the objects. Figure 12 shows an example of three selected **Label** objects.

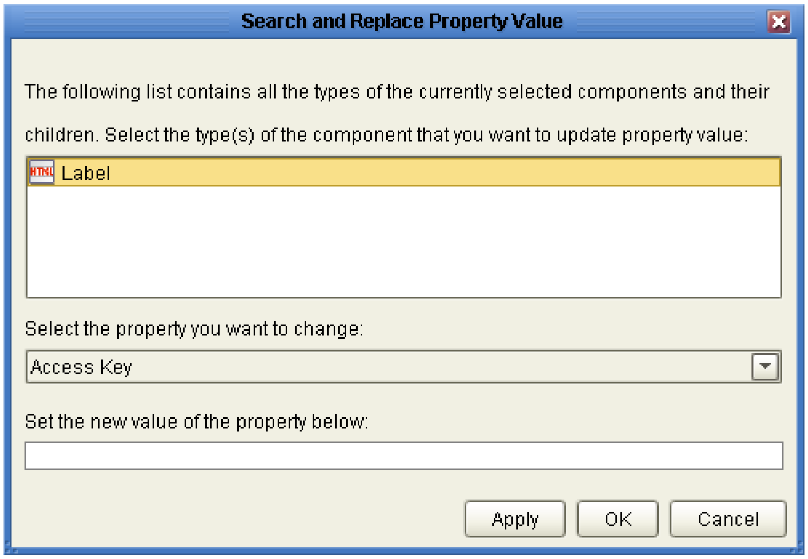
Figure 12 Multiple Selected Objects



2. On the Page Layout Designer toolbar, click the **Replace Properties** icon.

   The **Search and Replace Property Value** dialog box appears.

Figure 13 Search and Replace Property Value Dialog Box



Search and Replace Property Value

The following list contains all the types of the currently selected components and their children. Select the type(s) of the component that you want to update property value:

Label

Select the property you want to change:

Access Key

Set the new value of the property below:

3 If more than one object type appears, then select the object type.

4 Select the property from the drop-down list. The properties appear in alphabetical order.

5 Type the property value.

6 Click **OK**.

4.1.5 Design Canvas Features

The design canvas includes rulers, a background grid, and a status bar. You can display more of the design canvas by “floating” the object palettes and the property sheet. You can display a thumbnail view.

Rulers

You can display rulers on the upper and left borders of the design canvas. The measurements are displayed in inches.

To show or hide the rulers

1 Right-click the design canvas.

2 To show the rulers, select the **Show Ruler** check box.

3 To hide the rulers, clear the **Show Ruler** check box.

Background Grid

When you create a Page Layout, the background grid appears by default. You can specify that objects that you place or move on the design canvas are automatically aligned with the grid. This behavior is known as “snap to grid.”

You can control various grid properties, such as the width and height of each box within the grid.

To show or hide the background grid

1. Right-click the design canvas.
2. To show the grid, select the **Show Grid** check box.
3. To hide the grid, clear the **Show Grid** check box.

To turn on and off the “snap to grid” behavior

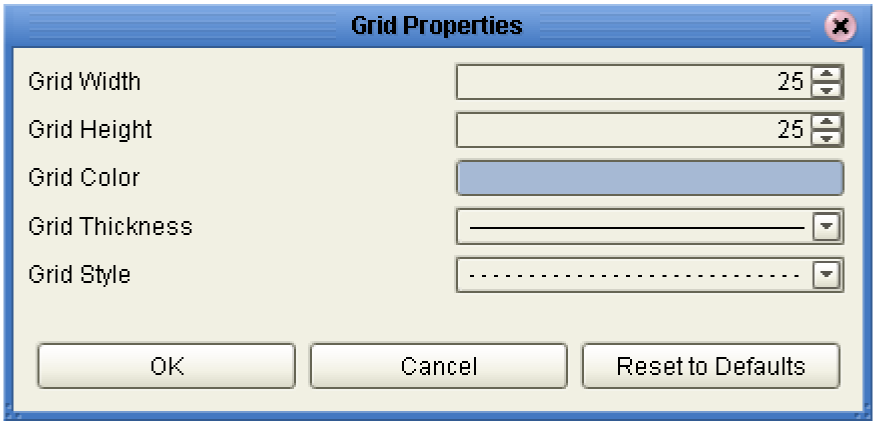
1. Right-click the design canvas.
2. To turn on the “snap to grid” behavior, select the **Snap to Grid** check box.
3. To turn off the “snap to grid” behavior, clear the **Snap to Grid** check box.

To edit grid properties

1. Right-click the design canvas and select **Grid Properties**.

   The **Grid Properties** dialog box appears.

Figure 14 Grid Properties Dialog Box



Grid Width 25
Grid Height 25
Grid Color
Grid Thickness
Grid Style

2. To change the width (in pixels) of each box within the grid, select a new value from the **Grid Width** spin box.
3. To change the height (in pixels) of each box within the grid, select a new value from the **Grid Height** spin box.
4. To change the color of the gridlines, do the following:

   A. Click the color that appears to the right of the **Grid Color** label. The **Choose a Color** dialog box appears.

   B. Specify the desired color.

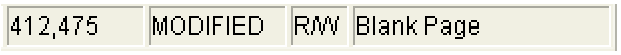
   C. Click **OK**.
5. To change the thickness of the gridlines, select the desired setting from the **Grid Thickness** drop-down list.

6. To change the style of the gridlines, select the desired setting from the **Grid Style** drop-down list.
7. To set all of the properties to their default values, click **Reset to Defaults**.
8. Click **OK**.

Status Bar

The status bar appears on the lower border of the design canvas.

Figure 15 Status Bar



412,475 | MODIFIED | R/W | Blank Page

The status bar displays the following information from left to right:

- The coordinates of the mouse pointer
- Whether the Page Layout has been modified since being opened or saved
- Whether the Page Layout is read/write or read only
- The Page Layout type

Floating Controls

If you want to display more of the design canvas on your screen, you can configure the object palettes and the property sheet to appear in a separate window.

To float controls

1. Right-click the design canvas.
2. Select the **Float Controls** check box.

   The object palettes and the property sheet appear in a separate window. The design canvas expands into the area formerly occupied by the object palettes and the property sheet.
3. To turn off this behavior, do either of the following:
   - In the window that contains the object palettes and the property sheet, click the close button.
   - Right-click the design canvas and clear the **Float Controls** check box.

Template Page

You can save the current Page Layout as a template page to quickly create another page from without having to do a **Save As** and maintain objects in the repository. After a page is saved as a template, it will show up in the New Page dialog under the tab templates. These pages are not present in the repository .

To save a Page Layout as a Template Page

1. Right-click the design canvas in an existing Page Layout.

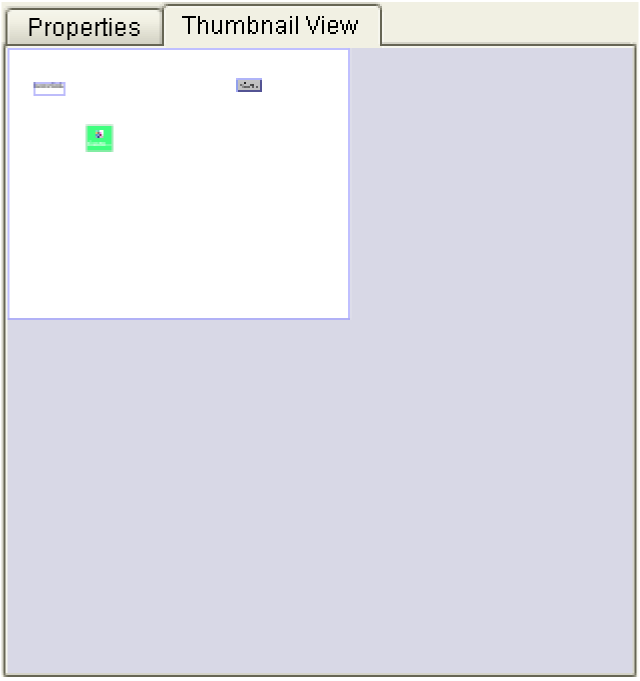
2 Select **Save as Template Page**.

The template page does not hold any references to Repository objects. For example, if there is a Page Link in the saved template page, it will no longer be linked to the Page Layout. This also holds true for images, which must be relinked due to the fact that they are not stored in the repository.

Thumbnail View

The design canvas might not show the entire web page. The **Thumbnail View** tab enables you to see which portion of the web page is currently displayed.

Figure 16 Thumbnail View Tab



You can move around the web page by by clicking the white portion and dragging it to another area in the **Thumbnail View** tab.

4.1.6 Importing Files

To make files (such as graphic images) accessible to a Page Layout, you must import the files into the Project.

To import a file

1 In the Project Explorer, right-click the Project.

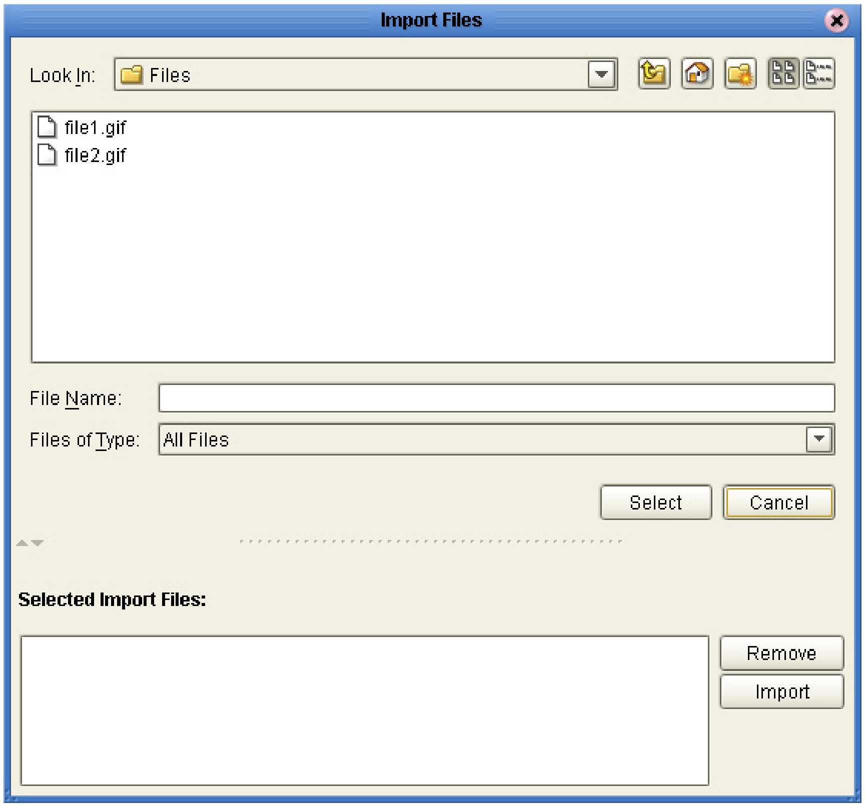
2 On the shortcut menu, point to **Import**, and then click **File**.

The **Import Files** dialog box appears.

3 Navigate to the directory that contains the file or files that you want to import.

The files can reside anywhere on your computer or a network.

Figure 17 Import Files Dialog Box



4 For each file that you want to import, select the file and click **Select**.

The file names appear in the **Selected Import Files** box.

5 To remove a file from the list, select the file name in the **Selected Import Files** box and click **Remove**.

6 When you are done, click **Import**.

The files are displayed under your Project in the Project Explorer.

4.2 Manipulating Objects on the Canvas

You can move objects, resize objects, and configure the z-direction of objects.

4.2.1 Moving a Single Object

To move a single object, select the object and drag it to any location on the canvas by using the mouse.

You can also move an object by using the arrow keys on your computer keyboard. When the object is where you want it, click elsewhere on the canvas.

4.2.2 Moving a Group of Objects

To choose two or more objects to be moved simultaneously, do one of the following:

- Click the canvas in a blank area near the objects, and then drag the mouse pointer over the objects. If the objects are inside a container, then hold down the Ctrl key beforehand.
- Hold down the Shift key and sequentially select the objects.

When the objects are selected, drag one of the objects. The other objects will follow.

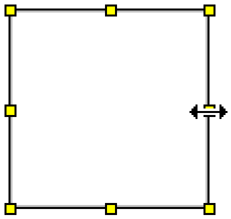
You can also move a group of objects by using the arrow keys on your computer keyboard. When the objects are where you want them, click elsewhere on the canvas.

4.2.3 Resizing Objects

You can resize objects vertically or horizontally, depending on the attributes of the object.

When an object is selected, the border contains one or more squares. Move the mouse pointer over a square. As shown in Figure 18, the pointer changes to a bidirectional arrow. Move the arrow in the appropriate direction.

Figure 18 Bidirectional Arrow



To maintain the object’s aspect ratio during resizing, hold down the Shift key. This feature is especially useful with images.

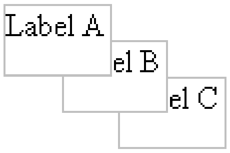
4.2.4 Configuring the Z-Direction of Objects

The standard two-dimensional graphical orientation consists of two values: **x** (horizontal) and **y** (vertical). A third value (**z**) implies another planar orientation, which is toward or away from the viewer. eVision Studio allows you to place objects under or on top of each other by using an integer to specify the "plane."

You can move an object backward or forward in the stack order by using the **Z-Index** property. The default value for this property is 0.

Figure 19 shows three overlapping **Label** objects. The **Z-Index** property for Label A is set to 1. The **Z-Index** property for Label B is set to 0. The **Z-Index** property for Label C is set to -1.

Figure 19 Label Objects with Different Z-Index Values



The shortcut menu of an object enables you to move the object in the z-direction one level at a time. This approach is the same as increasing or decreasing the **Z-Index** property by 1.

To move an object in the z-direction by using the Z-Index property

1. Select the object.
2. In the left column of the **Properties** tab, select the **Z-Index** property.
3. To move the object back (down) one or more levels, enter a negative integer in the right column.
4. To move the object forward (up) one or more levels, enter a positive integer in the right column. Do not include a plus sign.

To move an object in the z-direction one level at a time

1. Right-click the object.
2. To move the object back (down) one level, click **Send Backward**.
3. To move the object forward (up) one level, click **Bring Forward**.

4.3 Keyboard Shortcuts

You can use keyboard shortcuts to perform various tasks in the Page Layout Designer.

Table 3 lists the keyboard shortcuts that correspond to tools in the Page Layout Designer toolbar.

Table 3 Keyboard Shortcuts - Toolbar Tasks

Task	Shortcut
Cut Components	Ctrl+X
Delete Components	Delete
Copy Components	Ctrl+C
Paste Components	Ctrl+V
Undo	Ctrl+Z
Redo	Ctrl+Y
Left Align	Ctrl+L
Right Align	Ctrl+R
Center Align	Ctrl+I
Top Align	Ctrl+T
Bottom Align	Ctrl+B
Vertical Spacing	Ctrl+E
Horizontal Spacing	Ctrl+M
Resize Left to Align	Shift+L
Resize Right to Align	Shift+R
Resize Top to Align	Shift+T
Resize Bottom to Align	Shift+B
Preview	Ctrl+P

Table 4 lists the keyboard shortcuts for tasks that involve moving and resizing Page Layout objects. The primary selected component is the component that has the yellow selection handles.

Table 4 Keyboard Shortcuts - Moving and Resizing

Task	Shortcut
Move the selected object	Arrow key (left, right, up, or down)
Resize the selected object larger	Shift+Arrow key (left, right, up, or down)
Resize the selected object smaller	Alt+Shift+Arrow key (left, right, up, or down)

Table 5 lists the keyboard shortcuts for other tasks.

The term *focus cycle* refers to objects that have a parent-child relationship. For example, assume that a **Div** object contains two **Label** objects. The **Label** objects are children of the **Div** object. If the **Div** object has the focus, then you can navigate to the first child object by pressing Ctrl+Tab. Similarly, you can navigate from a child object to the parent object by pressing Ctrl+Shift+Tab.

Table 5 Keyboard Shortcuts - Other

Task	Shortcut
Save	Ctrl+S
Select all	Ctrl+A
Move the focus forward	Tab
Move the focus backward	Shift+Tab
Move the focus down to the next focus cycle	Ctrl+Tab
Move the focus up to the previous focus cycle	Ctrl+Shift+Tab
Display list of keyboard shortcuts	Ctrl+H

4.4 Style Sheets

Style sheets make it easier to apply a consistent look and feel to an eVision Studio Web application. Instead of specifying the formatting for each instance of a Page Layout object, you specify the formatting once for all instances of the object. For large Web applications, style sheets can save you a lot of time.

eVision Studio uses the Cascading Style Sheets (CSS) standard. In the CSS standard, a style sheet is a collection of style rules. Each style rule contains a selector and one or more property/value pairs. The following example contains a selector called **body** and two property/value pairs:

```
body {background-color: transparent; font-family: Serif;}
```

For detailed information about this standard, go to http://www.w3.org/Style/CSS/.

Note: *Different browsers might interpret a style rule in different ways. For example, one browser ignores the text alignment property, while another browser does not ignore the text alignment property.*

You can create a style sheet based on a template provided with eVision Studio. Alternately, you can import a style sheet that was created with a third-party tool.

4.4.1 Creating Style Sheets

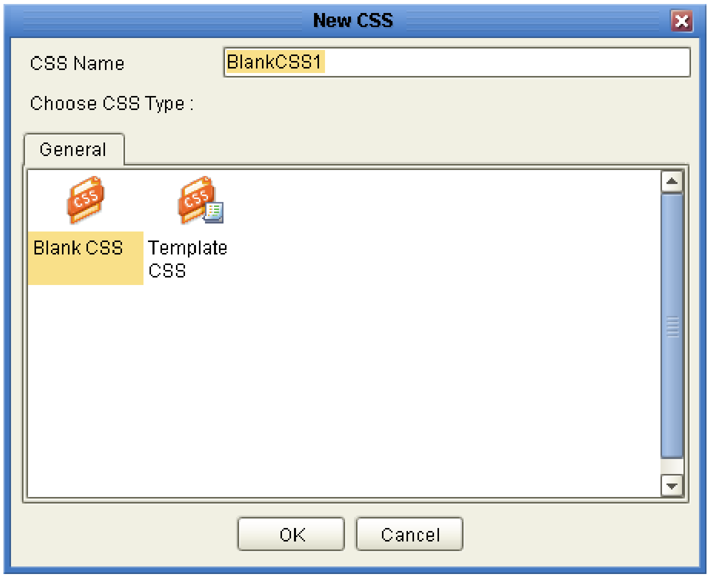
In the following procedure, you have the option of creating a blank style sheet or creating a style sheet based on a template provided with eVision Studio.

To create a style sheet

1. In the Project Explorer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page CSS**.

   The **New CSS** dialog box appears.

Figure 20 New CSS Dialog Box



New CSS

CSS Name: BlankCSS1

Choose CSS Type :

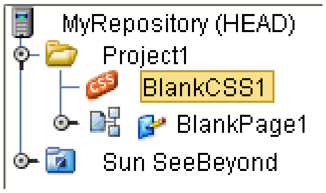
3 To create a blank style sheet, select **Blank CSS**. To create a style sheet based on the template provided with eVision Studio, select **Template CSS**.

4 In the **CSS Name** field, you can change the default name for the style sheet.

5 Click **OK**.

The style sheet appears in the Project Explorer.

Figure 21 Style Sheet in the Project Explorer



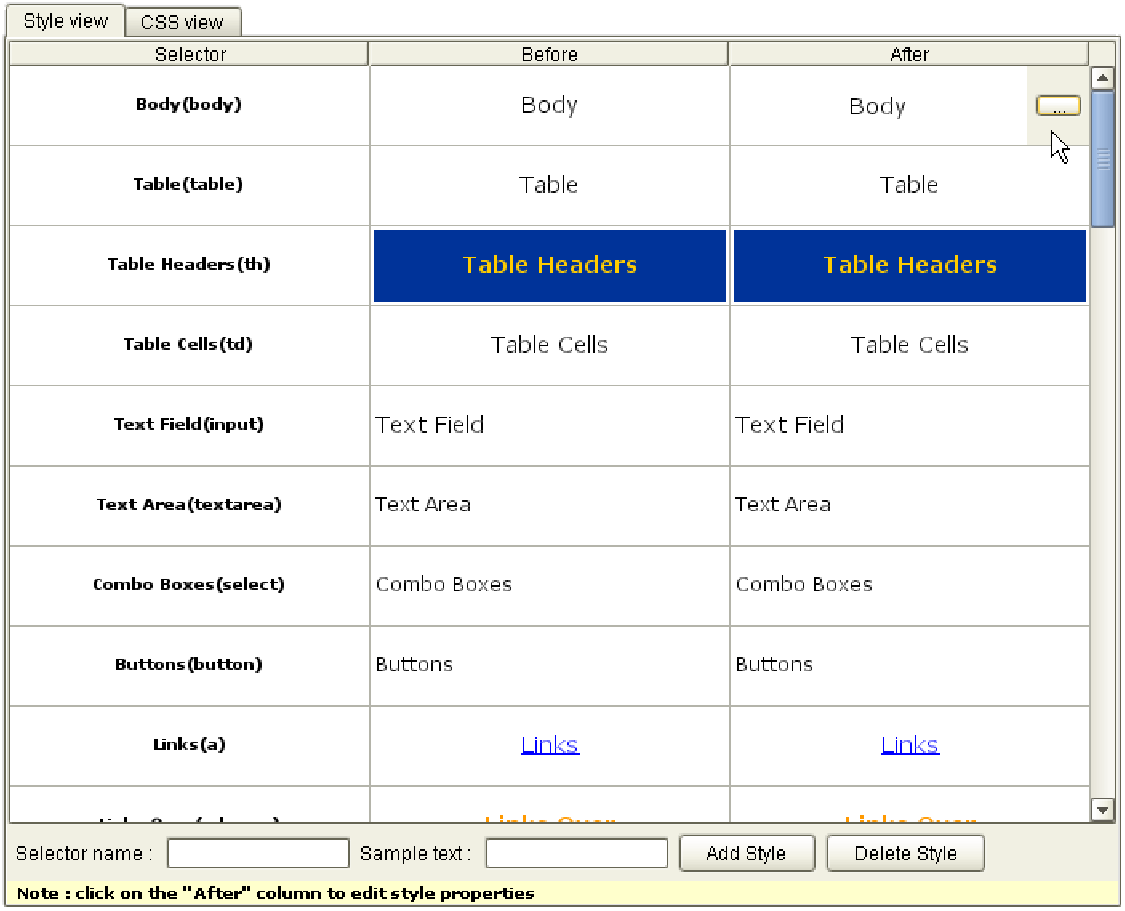
6 If you created a blank style sheet, then you can now copy the third-party style sheet into the Css Editor.

7 If you created a style sheet based on the template, then the Css Editor appears. **"Editing Style Sheets" on page 54** describes how to modify the default settings.

4.4.2 Editing Style Sheets

The Css Editor enables you to edit a style sheet based on the template provided with eVision Studio.

Figure 22 Css Editor



Selector	Before	After
Body(body)	Body	Body
Table(table)	Table	Table
Table Headers(th)	Table Headers	Table Headers
Table Cells(td)	Table Cells	Table Cells
Text Field(input)	Text Field	Text Field
Text Area(textarea)	Text Area	Text Area
Combo Boxes(select)	Combo Boxes	Combo Boxes
Buttons(button)	Buttons	Buttons
Links(a)	Links	Links

Note : click on the "After" column to edit style properties

The **Selector** column lists a descriptive name and the selector for each style rule in the style sheet.

The **Before** column shows the original look of the element.

The **After** column shows the current look of the element.

The bottom portion of the editor allows you to add and delete style rules:

- To add a style rule, enter the selector name and sample text, and click **Add Style**.
- To delete a style rule, select the style rule and click **Delete Style**.

If you want to examine the underlying style sheet, click the **CSS view** tab. When you are done, click the **Style view** tab.

Font Properties

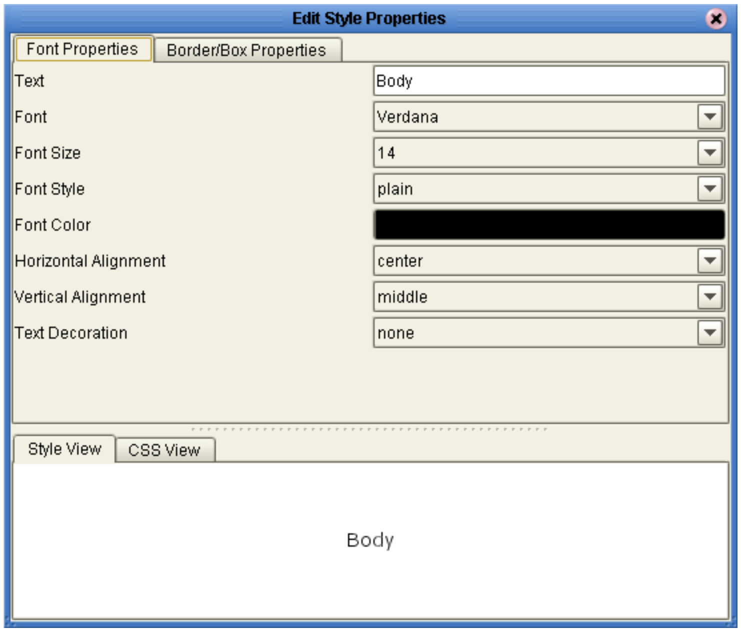
You can edit the font properties of an element from the Css Editor.

To edit font properties

1. In the row that lists the element, click anywhere in the **After** column, and then click the **Command** button (**...**).

   The **Edit Style Properties** dialog box appears.

Figure 23 Edit Style Properties Dialog Box - Font Properties



Text	Body
Font	Verdana
Font Size	14
Font Style	plain
Font Color	■
Horizontal Alignment	center
Vertical Alignment	middle
Text Decoration	none

2. The **Text** field enables you to enter new text for the **After** column.
3. The **Font** drop-down list enables you to specify the font family. The value **Inherit** indicates that the element uses the font family of its parent element.
4. The **Font Size** drop-down list enables you to select one of the predefined font sizes. The value is expressed in pixels.
5. The **Font Style** drop-down list enables you to apply bold and/or italic formatting.
6. The **Font Color** field enables you to specify the color of the element.
7. The **Horizontal Alignment** drop-down list enables you to specify the horizontal alignment: center, left, right, or justify.
8. The **Vertical Alignment** drop-down list enables you to specify the vertical alignment: middle, top, or bottom.
9. The **Text Decoration** drop-down list enables you to apply one of the following line effects: underline, overline, or line-through.
10. When you are done making changes, click the close icon in the upper right corner of the dialog box.

Border and Box Properties

You can edit the border and box properties of an element from the Css Editor.

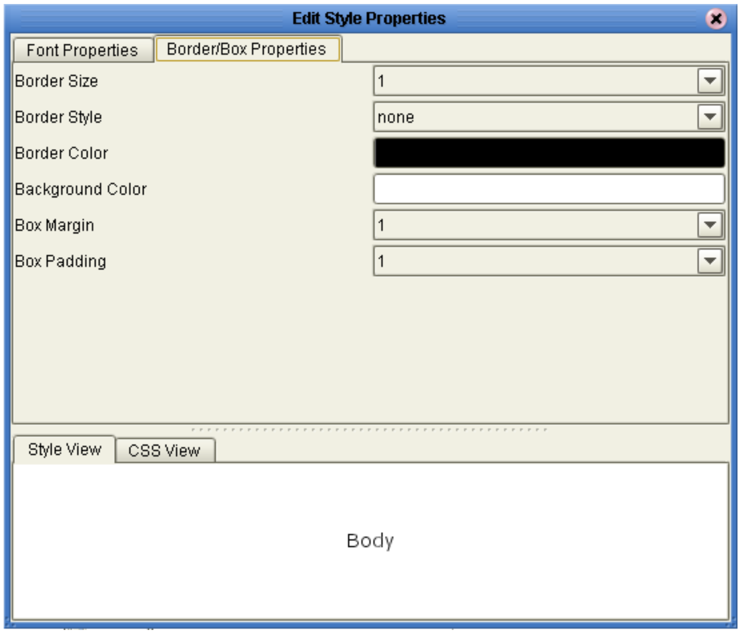
To edit border and box properties

1 In the row that lists the element, click anywhere in the **After** column, and then click the **Command** button (**...**).

The **Edit Style Properties** dialog box appears.

2 Click the **Border/Box Properties** tab.

Figure 24 Edit Style Properties Dialog Box - Border and Box Properties



Border Size 1

Border Style none

Border Color

Background Color

Box Margin 1

Box Padding 1

3 The **Border Size** drop-down list enables you to specify the thickness of the border. The value is expressed in pixels.

4 The **Border Style** drop-down list enables you to apply one of a variety of styles to the border.

5 The **Border Color** field enables you to specify the color of the border.

6 The **Background Color** field enables you to specify the color of the element's background.

7 The **Box Margin** drop-down list enables you to specify the amount of space that surrounds the element outside of the border. The value is expressed in pixels.

8 The **Box Padding** drop-down list enables you to specify the amount of space between the element and the border. The value is expressed in pixels.

9 When you are done making changes, click the close icon in the upper right corner of the dialog box.

4.4.3 Applying a Style Sheet to a Page Layout

When you are finished with a style sheet, you can apply the style sheet to one or more Page Layouts.

To apply a style sheet to a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **Style Sheet** property.
3. In the right column, click the **Command** button (**...**).

   The **Enter value for "Style Sheet"** dialog box appears.
4. Navigate to the Project folder and select the style sheet.
5. Click **Open**.

   The style sheet is applied.
6. If you want to remove the style sheet at a later time, right-click the design canvas and then click **Clear CSS Stylesheet**.

4.4.4 Applying a Local Style

You can apply a local style to an object on the design canvas. If a style sheet has been applied to the Page Layout, then the local style overrides the style sheet definition.

This procedure assumes that you know how to create a CSS style rule. For detailed information about CSS, go to http://www.w3.org/Style/CSS/.

As an example, assume that you want to create a table header row in which the background is red and the text is white. Set the **Background Color** property of the header row to red. Drag a **Label** object into each header cell. For each **Label** object, append the following local style:

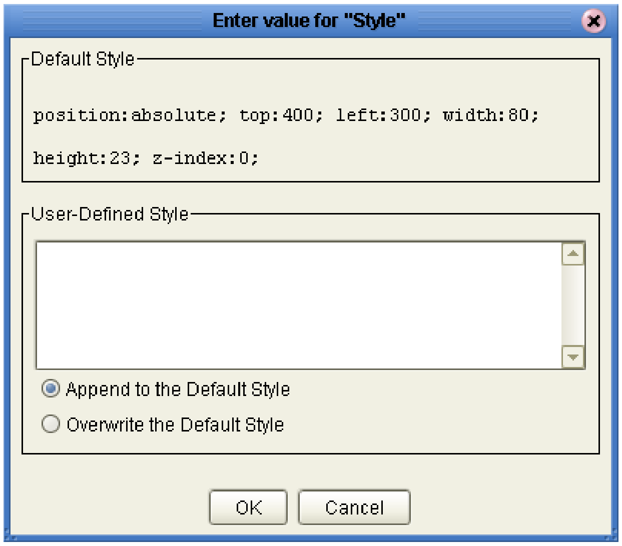
```
background-color:red; color:white;
```

To apply a local style

1. Select the object on the design canvas.
2. In the left column of the **Properties** tab, select the **Style** property.
3. In the right column, click the **Command** button (**...**).

   The **Enter value for "Style"** dialog box appears.

Figure 25 Enter value for “Style” Dialog Box



```
position:absolute; top:400; left:300; width:80;
height:23; z-index:0;
```

● Append to the Default Style

○ Overwrite the Default Style

4 In the **User-Defined Style** text field, enter the property, followed by a colon, followed by one or more values, followed by a semicolon. For example:

```
color:blue;
```

Note: *If you want to copy a dynamic version of the default style to the* **User-Defined Style** *text field, press F2.*

5 If you want to add the property/value pair to the default style, then select **Append to the Default Style**. If you want to replace the default style, then select **Override the Default Style**.

6 Click **OK**.

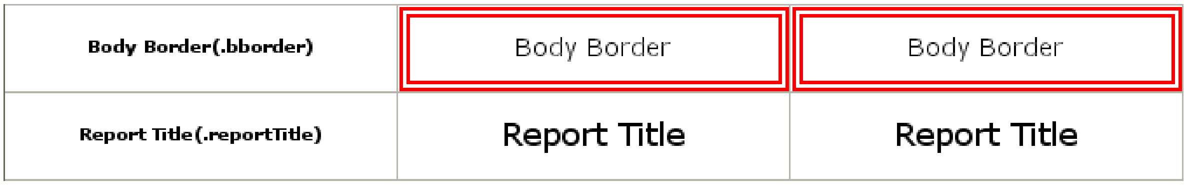
To remove a local style

- In the right column of the **Properties** tab, click the reset arrow for the **Style** property.

4.4.5 Applying Classes

The style sheet template provided with eVision Studio includes predefined classes. In the Css Editor, the classes contain a period before the class name (for example, **.bborder**). Figure 26 shows two of the classes.

Figure 26 Style Sheet Classes



Body Border(.bborder)	Body Border	Body Border
Report Title(.reportTitle)	Report Title	Report Title

You can modify the classes in a style sheet by using the Css Editor. See **"Editing Style Sheets" on page 54**.

After applying the style sheet to a Page Layout, you can apply a class to a Page Layout object.

To apply a class to an object

1. On the design canvas, select the object.
2. In the left column of the **Properties** tab, select the **Class** property.
3. In the right column, select a class from the drop-down list.

Note: *At runtime, the browser determines the final style based on inheritance and cascading rules. The final style might be different from what you assign by using the* **Class** *property.*

4.4.6 Referencing an External File

According to the CSS specification, a style sheet can include a reference to an external file. For example, the following property/value pair sets the background of an element to an image.

```
background-image:url("glblnav_selected.gif");
```

If you want to use a custom style sheet that includes references to one or more external file, then perform the following steps.

To incorporate a reference to an external file

1. Package the external file(s) in a **.zip** file. If the **url** reference points to a subdirectory, then be sure that the files in the **.zip** file use the same directory structure.
2. Import the **.zip** file into the Project. You can follow the procedure in **"Importing Files" on page 46**.
3. For each Page Layout where the style sheet is used, set the **Imports** property to the imported **.zip** file. To access the **Imports** property, select the **Page** property type in the **Properties** tab.

4.5 JavaScript

JavaScript is a client-side programming language that enables you to add a variety of enhancements to Web pages. For example, you can create a “rollover” effect, in which an image changes when the user moves the mouse pointer over the image. Or you can check whether the user has entered invalid characters in a form object.

This section describes the support for JavaScript in eVision Studio.

4.5.1 Event Handler Properties

Most of the Page Layout objects have a set of event handler properties. You can enter JavaScript code that will be executed when the event handler is triggered by the associated event.

Table 6 lists the complete set of properties and their associated events.

Table 6 Event Handler Properties

Property	Associated Event
On Blur	The object loses focus.
On Change	The user changes the default text, and the object loses focus.
On Click	The user clicks the object.
On Dbl Click	The user double-clicks the object.
On Focus	The object gains focus.
On Key Down	The object has focus, and the user presses a key.
On Key Press	The object has focus, and the user presses and releases a key.
On Key Up	The object has focus, and the user releases a key.
On Mouse Down	The user moves the mouse pointer over the object and then presses the mouse button.
On Mouse Move	The user moves the mouse pointer over the object and then moves the mouse pointer again within the object.
On Mouse Out	The user moves the mouse pointer over the object and then moves the mouse pointer away from the object.
On Mouse Over	The user moves the mouse pointer over the object.
On Mouse Up	The user moves the mouse pointer over the object, presses the mouse button, and then releases the mouse button.
On Select	The user selects all or part of the default text.

4.5.2 Applying External JavaScript Files to a Page Layout

You can import an externally created JavaScript file that contains functions, and then call the functions in the event handler properties.

Important: *Before you begin, you must import the file into the Project. For more information, see* **"Importing Files" on page 46**.

To apply an external JavaScript file to a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **JavaScript** property.
3. In the right column, click the **Command** button (...).

   The **Enter value for "JavaScript"** dialog box appears.
4. Select the file and click **Open**.

4.6 Additional Features

The Page Layout Designer includes a variety of additional features for enhancing the look and behavior of Page Layouts.

For example, you can do the following:

- Set the title of a Page Layout
- Set a background image for a Page Layout
- Create and apply a Page Layout template

4.6.1 Setting the Title of a Page Layout

The **Title** property of a Page Layout enables you to specify the text that appears in the title bar at runtime.

If you do not assign a value for the **Title** property, then the servlet context appears in the title bar. For information about the servlet context, see **"Servlet Context" on page 161**.

You can also set the title dynamically. In the Page Flow Designer, the **Title** property is exposed as a node of the Business Process Attributes.

To set the title of a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **Title** property.
3. In the right column, enter the value that you want to appear in the title bar.

4.6.2 Setting a Background Image or Color for a Page Layout

You can specify that the background of a Page Layout will contain an image or a color. The image appears in Preview mode, but not on the design canvas. The color appears both on the design canvas and in Preview mode.

Important: *Before you set a background image, you must import the image into the Project. For more information, see* **"Importing Files" on page 46**.

To set a background image for a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **Background** property.
3. In the right column, click the **Command** button (...).

   The **Enter value for "Background"** dialog box appears.
4. Select the image and click **Open**.
5. If you want to remove the background image at a later time, right-click the design canvas and select **Clear Background Image**.

To set a background color for a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **Background Color** property.
3. In the right column, click the **Command** button (...).

   The **Pick a Color** dialog box appears.
4. Select a color. If you want to have more control over the settings, click the **Advanced** button.
5. Click **OK**.

4.6.3 Creating and Applying Page Layout Templates

You can use a Page Layout as a template for other Page Layouts.

For example, assume that you want a logo to appear in the same place on five Page Layouts. Create a Page Layout and add the logo to the desired position. Then, go to the five Page Layouts and apply the Page Layout that has the logo. The logo will appear in the desired position in the five Page Layouts.

If you change the template, then any Page Layouts that use the template are updated automatically.

This feature is especially useful when you have a large number of Page Layouts to maintain.

Nesting of templates is supported. For example, assume that page A uses page B as a template, and page B uses page C as a template. In this scenario, page A will contain objects from both page B and page C.

At runtime, the **Z-Index** property of objects in the template interact properly with the **Z-Index** property of objects in the Page Layouts that use the template.

To create and apply a Page Layout template

1. Create the Page Layout that will be used as a template.
2. For each Page Layout that you want to use the template, do the following:

   A. In the **Properties** tab, select the **Page** property type.

   B. In the left column, select the **Template Page** property.

   C. In the right column, set the value to the Page Layout template.

   The objects from the template appear. The objects are read only. If you want to edit the objects, then you must do so from the template.

4.6.4 Automatically Resizing Page Layout Objects

When the **Auto Size** property of a Page Layout object is set to **true**, the object automatically expands or contracts as you change the object's text. You do not need to manually resize the object.

This feature is a design-time feature, not a runtime feature.

The following objects include the **Auto Size** property: **Checkbox Group**, **Drop-Down List**, **Hidden Input**, **HTML**, **Label**, **Link**, **Radio Group**, **Reset Button**, and **Submit Button**. By default, the value is set to **true**.

If you use the style sheet feature to set the size of an object, then the **Auto Size** property is ignored.

4.6.5 Automatically Refreshing Page Layouts

You can indicate that a Page Layout will automatically refresh after a specified period of time. A typical use for this feature is when the page contains dynamic content.

The following procedure requires you to set the **Refresh** and **Refresh Seconds** properties of the Page Layout. If the Page Layout is *not* going to be the first Page Layout in a Page Flow, then you must set the **Method** property of the immediately preceding Page Layout.

To automatically refresh a Page Layout

1. Open the Page Layout that you want to be automatically refreshed.
2. In the **Properties** tab, select the **Page** property type.
3. Set the value of the **Refresh** property to **true**.

   A new property called **Refresh Seconds** appears.
4. The **Refresh Seconds** property indicates how many seconds to wait before reloading the page. The default value is 60 seconds. If desired, change the value.
5. Save the Page Layout.
6. If the Page Layout does not arrive via a Page link, but rather a form Submission, then perform the following steps.

   A. Open the Page Layout that will immediately precede the current Page Layout in the Page Flow.

   B. In the **Properties** tab, select the **Form** property type.

   C. Set the value of the **Method** property to **get**.

   D. Save the Page Layout.

4.6.6 Using the Save As Feature

The Save As feature enables you to save a Page Layout in the same Project or in a different Project.

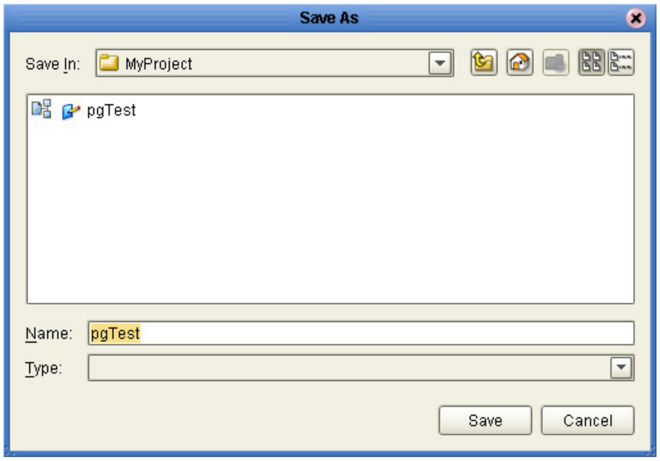
If you save the Page Layout in the same Project, you must enter a new name for the Page Layout. If you save the Page Layout in a different Project, the Page Layout can have the same name or a different name.

To use the Save As feature

1. In the Project Explorer, right-click the Page Layout.
2. On the shortcut menu, click **Save As**.

   The **Save As** dialog box appears.

Figure 27 Save As Dialog Box


Save In: MyProject

pgTest

Name: pgTest

Type:

3. If you want to save the Page Layout in a different Project, navigate to the desired Project.
4. If you want to change the Page Layout name, do so in the **Name** field.
5. Click **Save**.

4.6.7 Creating Template Objects

You can add any object on the design canvas to the **Template Objects** palette for later reuse. This feature is especially helpful when you make a large number of changes to the default behavior of an object, and you plan to use the object again.

When you create a template object, eVision Studio does not preserve any external references (such as images and links) that are defined in the object. For example, the **Src** property of the **Close** object defines the image that appears for the button. If you save the **Close** object as a template object, then the value of the **Src** property is cleared in the template object.

After creating a template object, you can edit and delete the object.

To create a template object

1. On the design canvas, right-click the object and then click **Save as Template Object**.
2. Enter a name for the object.
3. Click **OK**.

   The object is added to the **Template Objects** palette. You can now drag the object onto any Page Layout.

To edit a template object

1. In the **Template Objects** palette, right-click the object and then click **Edit**.

   The **Edit Properties** dialog box appears.
2. Change the value of one or more properties.
3. Click **OK**.

To delete a template object

1. In the **Template Objects** palette, right-click the object.
2. If you want to delete the object, then click **Delete**. If you want to delete all of the template objects, then click **Delete All Template Objects**.
3. When prompted to confirm this action, click **Yes**.

4.6.8 Editing the Default Property Values of Non-Template Objects

You can edit the default property values of non-template objects. You can subsequently reset the default property values to their original settings.

To edit the default property values of an object

1. In the object palette, right-click the object and then click **Edit Defaults**.

   The **Edit Properties** dialog box appears.

2. Change the default value of one or more properties.
3. Click **OK**.

   The object name in the object palette now appears in blue.

To reset the default property values to their original settings

1. In the object palette, right-click the object.
2. If you want to reset the default values for this object only, then click **Reset to Defaults**. If you want to reset the default values for all of the non-template objects, then click **Reset All to Defaults**.

   The object name no longer appears in blue.

4.7 Page Links

Page Links enable you to create links to other Page Layouts. A special type of Page Link points to the home page of an eVision Studio web application.

In the Page Layout Designer, you can drag a Page Link from the Project Explorer and drop it onto a Page Layout.

In the Page Flow Designer, you can drag the **link** operation of a Page Link from the Project Explorer and drop it onto a Page Flow.

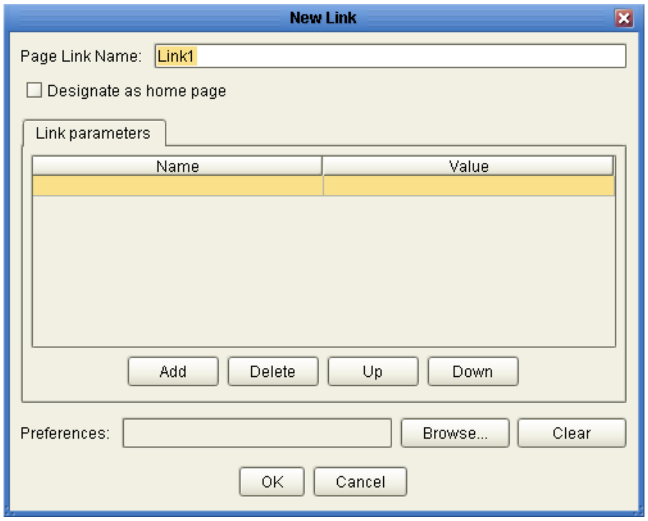
Note: *Do not confuse Page Links with the* **Link** *object. For information about the* **Link** *object, see* **"Links" on page 71**.

To create a Page Link

1. In the Project Explorer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page Link**.

   The **New Link** dialog box appears.

Figure 28 New Link Dialog Box



New Link

Page Link Name: Link1

☐ Designate as home page

Link parameters

Name	Value

Preferences:

3. The **Page Link Name** field enables you to change the default name.
4. If you want the link to point to the Page Flow's home page, then select the **Designate as home page** check box. The page will be the first page that appears when a user accesses the web application. In a Page Flow, you can designate only one page as the home page.

5 If desired, add a Page Link parameter.

A Page Link parameter is a data type that is passed by way of the link to an object on the linked destination page. You can pass one or more parameters.

Assume that you create a Project that includes the Page Flow shown in **"Event Based Decision" on page 128**. If the **plCredit** Page Link contains a parameter, then the parameter is passed to the **pgCredit** Page Layout when a user clicks the link.

In the **Name** column, enter the parameter's name (the internal system name used when specifying a target for the link). In the **Value** column, enter the data to be passed by way of the link to a target object on the destination page.

Note: *You must assign a value to the Page Link, even if the value is dynamically assigned at runtime.*

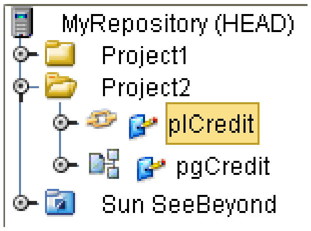
To add another parameter, click **Add** and enter the name and value. To delete a parameter, select the parameter and click **Delete**. To move a parameter up one level in the stack, select the parameter and click **Up**. To move a parameter down one level in the stack, select the parameter and click **Down**.

6 If you want to add a Preferences object, then click **Browse** to select the object. For more information, see **Chapter 9 "Generating Portlets"**.

7 Click **OK**.

The Page Link is added to your Project in the Project Explorer.

Figure 29 Page Link in the Project Explorer



Chapter 5

Object Palettes

The Page Layout Designer includes the following object palettes: HTML Objects, Form Objects, Extension Objects, and Custom Objects.

What's in This Chapter

5.1 HTML Objects

The **HTML Objects** palette enables you to drag and drop HTML-based objects onto the Page Layout Designer canvas. You can also access these objects from the **All Objects** palette.

Table 7 describes the HTML objects.

Table 7 HTML Objects

Object	Name	Purpose
	Div	Creates an area called a division. You can drop other objects into the division and format them with the same styles. For more information, see **"Divisions" on page 86**.
	Horizontal Line	Places a horizontal line on the canvas. For more information, see **"Horizontal Lines" on page 84**.
	HTML	Creates a container for HTML code. For more information, see **"HTML Containers" on page 87**.

Table 7 HTML Objects

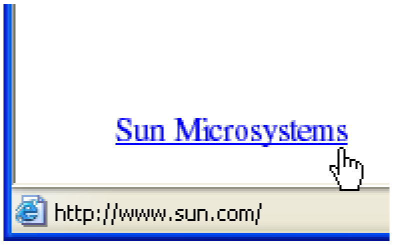
Object	Name	Purpose
🖼	Image	Holds a static graphic image. For more information, see **"Images" on page 82**.
🖼	Image Map	Enables you to create a "hotspot" on an image that a user can select to perform a link action, linking to another page or an area within the current page. You import the base image the same way that you import a standard graphic image. For more information, see **"Image Maps" on page 83**.
⬚	Inline Frame	Enables you to create a frame within a Page Layout. For more information, see **"Inline Frames" on page 85**.
HTML	Label	Creates a field on the canvas to hold static text, or a placeholder for dynamic text. You use this object for labels and general information on a page. For more information, see **"Labels" on page 81**.
🔗	Link	Creates an HTML link that points to an internal or external web site. For more information, see **"Links" on page 71**.
▦	Table	Enables you to organize information into rows and columns. You can add other objects to the table cells. A table can include dynamic rows, which are defined at runtime. For more information, see **"Tables" on page 73**.

5.1.1 Links

The **Link** object enables you to add an HTML link that points to an internal or external web site.

Figure 30 shows how a link to the Sun Microsystems web site appears in Preview mode.

Figure 30 Link Example (Preview Mode)



Note: *Do not confuse the* **Link** *object with Page Links. For information about Page Links, see* **"Page Links" on page 68**.

To create a link

1. From the **HTML Objects** palette, drag the **Link** icon onto the canvas.
2. You can enter text for the object now, or at a later time by using the **Text** property.

   For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.
3. The **Charset** property enables you to specify the character set of the target location.
4. Set the **HRef** property to the target location (for example, **http://www.sun.com**).
5. The **Rel** property enables you to describe the forward link to the target location. The **Rev** property enables you to describe a reverse link to the current page.
6. You can use the **Target** property to specify how the target location opens. Table 8 describes the valid values.

Table 8 Valid Values of target Property

Value	Description
_blank	The target location opens in a new window.
_self	The target location opens in the same frame.
_parent	The target location opens in the parent frame (that is, one level up in the frame hierarchy).
_top	The target location opens in the full body of the same window.

   If the value is empty, then the target location opens in the full body of the same window.
7. The **Type** property enables you to specify what type of content is available at the target location. You must enter a valid MIME type (for example, **text/html**).

   For information about the **Class** property, see **"Applying Classes" on page 59**.
8. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

9 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

10 The **HRef Language** property enables you to specify the base language (for example, **fr**) of the target location.

11 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

12 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

13 The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

14 The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code. For a list of language codes, see RFC 1766, “Tags for the Identification of Languages.”

Using JavaScript to Specify Properties of a New Window

You can use the JavaScript method **window.open()** to specify various properties of a link that opens in a new window.

To specify properties of a new window

1 Ensure that the **Target** property is empty, which is the default setting.

2 In the **On Click** property, enter the **window.open()** method. For example:

```
window.open(this.href,'reassignWindow1','toolbar=no,menubar=no,loc
ation=no,scrollbars=no,resizable=yes,status=no,width=600,height=40
0');return false;
```

The first parameter must be set to **this.href**.

The second parameter is an internal name for the window. You can set this parameter to any value. However, if you have multiple links on a page and you want them to open in separate windows at the same time, then the second parameter for each link must be unique.

The third parameter enables you to specify the properties, including the width and height.

5.1.2 Tables

The **Table** object enables you to organize information into rows and columns. You can add other objects to the table cells.

The **Table** object contains the following property types:

Table 9 Table Property Types

Property Type	Description
Divtable	Properties for the object’s container.
Table	Properties for the entire table.
Thead	Properties for the table header.
Tbody	Properties for the table contents.
Tfoot	Properties for the table footer.
Tr	Properties for a row.
Td	Properties for a cell.
Th	Properties for a cell in a header or footer.

Most of the properties described in this section are in the **Table** property type.

Creating Tables

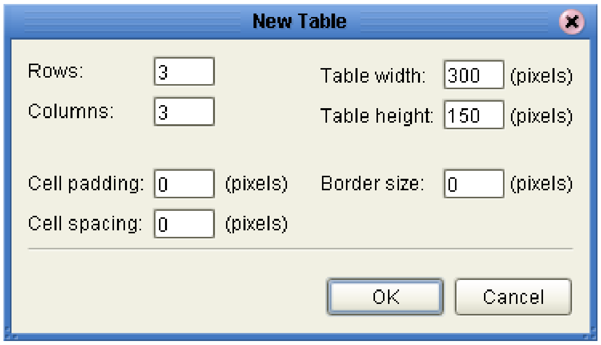
When you drag the **Table** icon onto the canvas, a dialog box enables you to specify basic properties.

To create a table

1 From the **HTML Objects** palette, drag the **Table** icon onto the canvas.

The **New Table** dialog box appears.

Figure 31 New Table Dialog Box



New Table

Rows: 3
Columns: 3
Table width: 300 (pixels)
Table height: 150 (pixels)
Cell padding: 0 (pixels)
Cell spacing: 0 (pixels)
Border size: 0 (pixels)

2 This dialog box enables you to specify basic properties, such as the number of rows and columns and the width. If desired, change the default values.

3 Click **OK**.

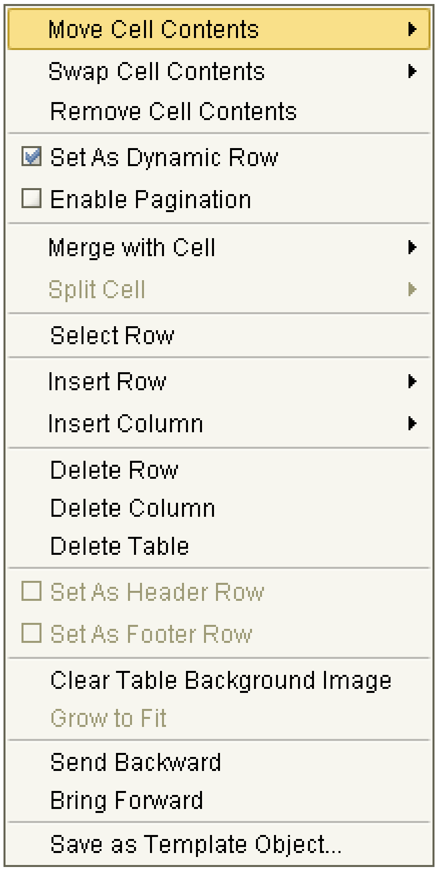
The table appears.

Modifying Tables

After the table appears on the canvas, you can use the **Properties** tab to modify any of the properties. These properties include basic characteristics, such as the number of rows and columns and the width. Additional properties include the background image and color.

You can use the shortcut menu to modify certain properties and to manipulate cell contents. Figure 32 shows the shortcut menu.

Figure 32 Table Shortcut Menu



To modify basic properties of a table

1. You can modify the number of rows and columns in one of the following ways:
   - Change the **Rows** and **Columns** properties in the **Properties** tab
   - Right-click a table cell, point to **Insert Row** or **Insert Column**, and then click the appropriate menu option
   - Right-click a table cell and then click **Delete Row** or **Delete Column**
2. To change the width of the table, select the **Table** property type and set the **Width** property to the desired value (in pixels).
3. To add space around the contents of the cells, select the **Table** property type and set the **Cell Padding** property to the desired number of pixels.

   To add space between the cells, select the **Table** property type and set the **Cell Spacing** property to the desired number of pixels.

4 To add a border to the table, select the **Table** property type and set the **Border** property to the desired thickness (in pixels).

To specify which outer borders are displayed, select the **Table** property type and set the **Frame** property to the desired value.

To specify which inner borders are displayed, select the **Table** property type and set the **Rules** property to the desired value.

5 To specify that the first row is the table header, right-click a cell in the first row and click **Set As Header Row**. The hierarchy of the property types in the **Properties** tab changes accordingly.

To specify that the last row is the table footer, right-click a cell in the last row and click **Set As Footer Row**. The hierarchy of the property types in the **Properties** tab changes accordingly.

To modify additional properties

1 The **Align** property enables you to specify the horizontal alignment of the table in the Page Layout. The alignment can be left, center, or right. When you set this property, you must also modify the **Style** property for the **Divtable** and **Table** property types so that the style uses relative positioning, rather than absolute positioning.

2 By default, the **LName** properties of the rows and columns are automatically renamed when you insert or delete a row or column. If you want to disable this behavior, then set the **Auto Rename** property to **false**.

3 To add a background image to a table, import the image into the Project, select the **Table** property type, and modify the **Background** property. **"Importing Files" on page 46** describes how to import an image into a Project.

To remove a background image, right-click a table cell and click **Clear Table Background Image**.

4 To add a background color to a table, select the **Table** property type and set the **Background Color** property to the desired color.

5 To remove a background color, click the reset arrow for the property.

For information about the **Class** property, see **"Applying Classes" on page 59**.

6 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

7 The **Summary** property enables you to specify alternate text for the table.

8 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

9 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

10 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

11 The **Auto Grow** property indicates whether the table resizes automatically when you add content to the cells. The default value is **true**. If you want to disable this behavior, then set the value to **false**. This property is located in the **Divtable** property type.

When the **Auto Grow** property is set to **false**, you can enable the behavior for an individual cell by right-clicking the table and then clicking **Grow to Fit**.

To manipulate cell contents

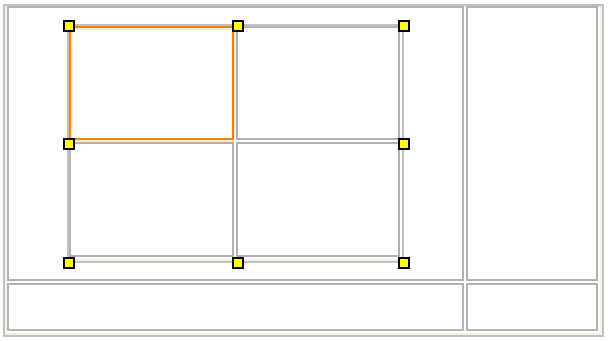
1. To move the contents of a table cell to another cell, right-click the cell, point to **Move Cell Contents**, and then click the appropriate menu option.
2. To swap the contents of a table cell with the contents of another cell, right-click either cell, point to **Swap Cell Contents**, and then click the appropriate menu option.
3. To remove the contents of a table cell, right-click the cell and then click **Remove Cell Contents**.
4. To merge a table cell with another table cell, right-click the cell, point to **Merge with Cell**, and then click the appropriate menu option.

   To split a merged table cell, right-click the cell, point to **Split Cell**, and then click the appropriate menu option.
5. To select a row, right-click any cell in the row and then click **Select Row**.

Nested Tables

You can add a **Table** object to a cell of another **Table** object. The inner table is called a *nested table*.

Figure 33 Nested Table



Dynamic Tables

In a dynamic table, one row is designated as “dynamic.” Both the total number of rows and the content of the rows are undefined. At runtime, table rows and content are dynamically generated by a repeating node in a Page Flow.

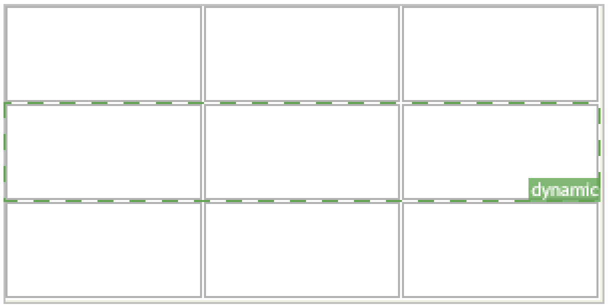
For information about mapping data into a dynamic table, see **“Dynamic Tables” on page 144**.

To define a dynamic row in a table

1. Select a cell in the row.
2. Right-click the cell, and then click **Set As Dynamic Row**.

   The row is marked as being dynamic.

Figure 34 Table With Dynamic Row



Note: *eVision Studio allocates 20 extra pixels to the width of a dynamic table. If a vertical scrollbar is needed at runtime, these pixels enable the scrollbar to appear correctly.*

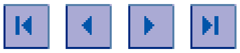
Adding Pagination to a Dynamic Table

You can add “page-forward, page-back”, “page-first, page-last” functionality to a row in a dynamic table. The pagination feature enables table cell data to be displayed incrementally. For example, if the table is set to read the results of a database query, the table will display the first 10 items returned by query, and enable the user to page forward to the second 10 items, and so on.

All rows are retrieved onto the client side, and the pagination is performed on the client side.

eVision Studio provides the following default image files for the pagination buttons: **firstImg.gif**, **prevImg.gif**, **nextImg.gif**, and **lastImg.gif**. You can replace the default images.

Figure 35 Default Pagination Buttons

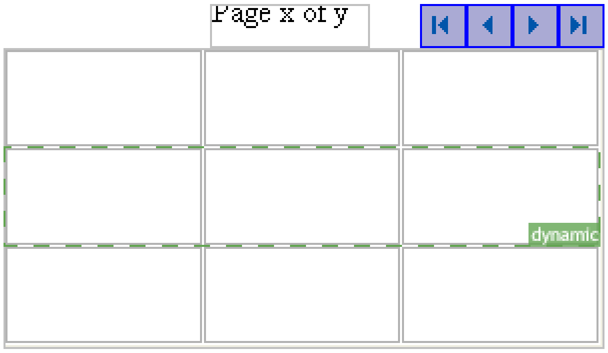


To add pagination to a row in a dynamic table

1. Right-click the dynamic row and then click **Enable Pagination**.

   The pagination label and buttons appear above the table.

Figure 36 Table With Dynamic Row and Pagination Enabled



2 If desired, move the label or any of the buttons to a different location in the Page Layout.

3 When you add pagination to a row, a property called **Paginate Count** is added to the **Table** property type. This property specifies how many rows the table accepts at a time. The default value is 10. If desired, increase or decrease the value.

To modify the pagination label

1 Select the label.

2 For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

3 The **Text** property is read only.

To modify a pagination button

1 Select the button.

2 If you want to replace the image file for the button, do the following:

A Import the new image file. See **"Importing Files" on page 46**.

B In the **Properties** tab, set the **Src** property to the imported file.

3 The **Border** property enables you to add a border to the image. The default value of 0 indicates the absence of a border. The value is measured in pixels.

4 The **HSpace** property enables you to add space to the left and right of the image. The value is measured in pixels.

5 The **Height** property enables you to override the default height of the image. The value is measured in pixels.

6 The **VSpace** property enables you to add space above and below the image. The value is measured in pixels.

7 The **Width** property enables you to override the default width of the image. The value is measured in pixels.

Adding Sorting to a Dynamic Table

You can specify that a column in a dynamic table will be sorted. The sort criteria include alphabetic, numeric, and date.

To add sorting to a dynamic table

1. In the row immediately above the dynamic row, click the cell in the column that you want to sort.
2. Set the value of the **Is Sortable** property to **true**.
3. Specify how the column should be sorted by selecting a value for the **Sort Type** property. If you want to consider case when sorting alphabetic values, select the **alpha** value; otherwise, select the **alphaIgnoreCase** value.

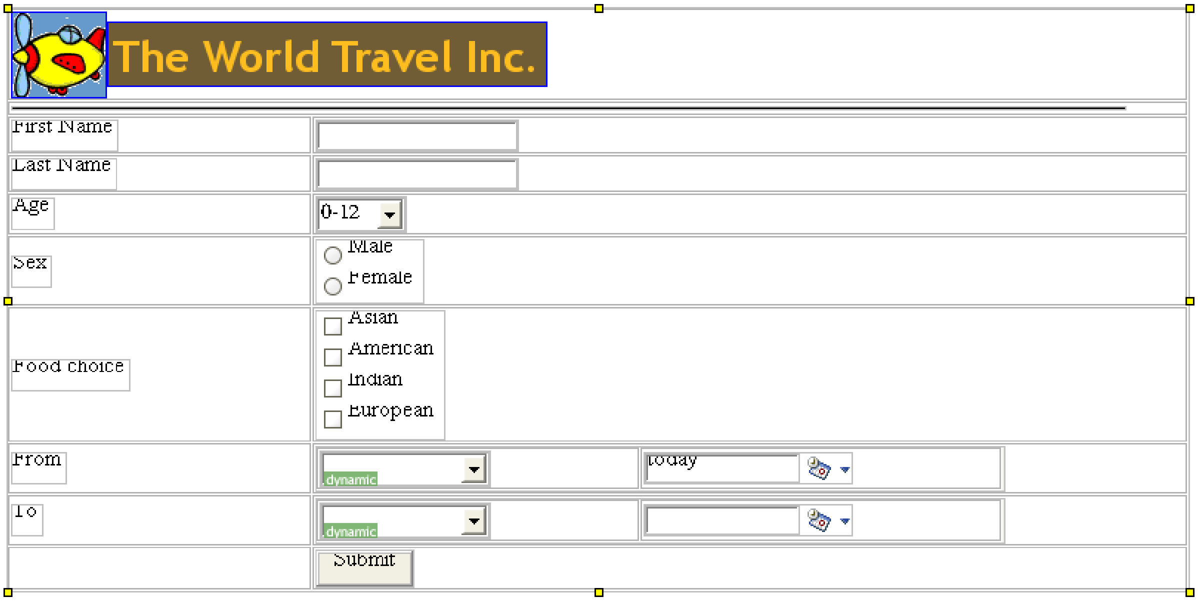
Using Tables in Composing Forms

In addition to using tables for tabular data, you can use them to guide layout.

Assume that you want a Page Layout to include a form for users to fill out. You could add a table, and then place various objects inside the table.

Figure 37 shows an airline reservation form that is based on a table.

Figure 37 Form Based on a Table



First Name		
Last Name		
Age	0-12 ▾	
Sex	○ Male ○ Female	
Food choice	□ Asian □ American □ Indian □ European	
From	dynamic ▾	today
To	dynamic ▾	
	Submit	

The first row contains two images. The two cells in the row have been merged into one.

The second row contains a horizontal line. The two cells in the row have been merged into one.

The third row contains a label (**First Name**) and a text input field.

The fourth row contains a label (**Last Name**) and a text input field.

The fifth row contains a label (**Age**) and a drop-down list.

The sixth row contains a label (**Sex**) and a radio group.

The seventh row contains a label (**Food choice**) and a check box group.

The eighth row contains a label (**From**) and a nested table, which in turn contains a drop-down list and a calendar. The drop-down list is dynamic.

The ninth row contains a label (**To**) and a nested table, which in turn contains a drop-down list and a calendar. The drop-down list is dynamic.

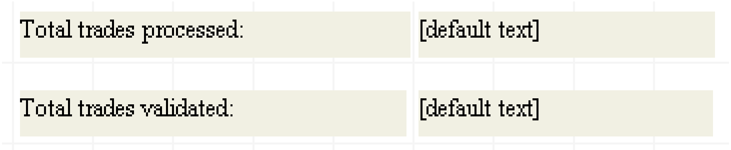
The tenth row contains a submit button.

5.1.3 Labels

The **Label** object creates a field on the design canvas to hold static text, or a placeholder for dynamic text. Use this object for labels and general information.

Figure 38 shows a group of four **Label** objects on the design canvas. The objects on the left are static, and the objects on the right are dynamic.

Figure 38 Label Examples



Total trades processed:	[default text]
Total trades validated:	[default text]

To create a label

1. From the **HTML Objects** palette, drag the **Label** icon onto the canvas.
2. You can enter text for the object now, or at a later time by using the **Text** property.

   For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.
3. The **Container** property enables you to specify the HTML tag in which the code will be located.

   For information about the **Class** property, see **"Applying Classes" on page 59**.
4. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **"Applying a Local Style" on page 57**.

   For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.
5. If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.
6. The **For** property enables you to associate the label with a form object. Enter the value of the form's object **ID** property.
7. The **Title** property enables you to enter a tooltip.

   For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

8 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

9 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.1.4 Images

The **Image** object holds a static graphic image. When you drag the icon onto the canvas, you are prompted to select the image.

Note: *Before you begin, you must import the image into the Project. For more information, see* **"Importing Files" on page 46***.*

To create an image

1 From the **HTML Objects** palette, drag the **Image** icon onto the canvas.

The **Enter value** dialog box appears.

2 Select the image and click **Open**.

The image appears on the canvas.

3 The **Align** property enables you to specify how the image should be aligned with the surrounding text.

4 The **Border** property enables you to add a border to the image. The default value of 0 indicates the absence of a border. The value is measured in pixels.

5 If you want the My Pictures image toolbar to appear when the mouse pointer is placed over the image, set the **Gallery Image** property to **yes**. This property is specific to Internet Explorer.

Note: *The* **Page** *property type includes a property called* **Image Toolbar***, which specifies the behavior of the My Pictures image toolbar for the overall Page Layout. Setting the* **Gallery Image** *property overrides the* **Image Toolbar** *setting.*

6 The **Height** property enables you to override the default height of the image. The value is measured in pixels.

7 The **HRef** property enables you to specify a URL that will be launched when the user clicks the image.

Note: *You cannot use both the* **HRef** *property and the* **Link** *property.*

8 The **HSpace** property enables you to add space to the left and right of the image. The value is measured in pixels.

9 The **Is Map** property indicates whether the image is a server-side image map.

10 The **Link** property enables you to specify a URL that will be launched when the user clicks the image.

11 The **Name** property enables you to assign a name that can be referenced in scripting.

12 The **Src** property enables you to change the image.

13 You can use the **Target** property to specify how the target location opens. This property is associated with the **HRef** and **Link** properties.

14 The **Use Map** property enables you to set the label name for a client-side image map.

15 The **VSpace** property enables you to add space above and below the image. The value is measured in pixels.

16 The **Width** property enables you to override the default width of the image. The value is measured in pixels.

For information about the **Class** property, see **“Applying Classes” on page 59**.

17 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **“Applying a Local Style” on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

18 The **Alt** property enables you to specify replacement text for the image.

19 The **Long Description** property enables you to specify a URL that provides additional information about the image (beyond that provided by the **Alt** property).

20 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

21 The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

22 The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code.

5.1.5 Image Maps

You create an image map by bounding an area of an image and attaching linking code to the area within the boundary. Users can then click the area to execute the code. You can define an image map to link to an external web site, or to link to another page in the Page Flow.

Note: *Before you begin, you must import the image into the Project. For more information, see* **“Importing Files” on page 46**.

To create an image map

1 From the **HTML Objects** palette, drag the **Image Map** icon onto the canvas.

The **Enter value** dialog box appears.

2 Select the image and click **Open**.

The image appears on the canvas.

3 To add a link to the image, do one of the following:

- In the Project Explorer tree, drag a Page Link and drop it onto the image. This link has predefined parameters and a target location.
- From the **HTML Objects** palette, drag the **Link** icon onto the image. This link requires that you specify link parameters and a target location.

When you drop a link onto the image, a box appears. This is the *link area*.

4 In the **Properties** tab, specify the target location of the link area (if necessary):

A In the left column, select the **HRef** property.

B In the right column, enter the target location (for example, **http://www.seebeyond.com**).

5 The **Shape** property indicates whether the area is a rectangle, circle, or polygon.

6 The **Coordinates** property indicates where the area is located. The format depends on the shape of the area.

- If the shape is a rectangle, then the format has four values: left-x, top-y, right-x, bottom-y.
- If the shape is a circle, then the format has three values: center-x, center-y, radius.
- If the shape is a polygon, then the format has the following values: x1, y1, x2, y2, ..., xN, yN.

7 If desired, create additional link areas in the image.

5.1.6 Horizontal Lines

The **Horizontal Line** object enables you to create an HTML horizontal rule.

To create a horizontal line

1 From the **HTML Objects** palette, drag the **Horizontal Line** icon onto the canvas.

2 The line is initially of a fixed length. To extend the line, place the pointer on the end that you want to change (right or left, up or down). The pointer changes to a resize arrow. Drag the line end to modify the length. The bidirectional arrow cursor is dual-purpose. You can add thickness to the line as well as adjust the length. Drag the line end carefully, taking care not to add thickness to the line.

3 To move the line, click the line’s mid-point (avoid the end points) and drag it to a new location.

4 If you do not want the line to include a three-dimensional shading effect, then set the **No Shade** property to **false**.

5 The **Size** property enables you to specify the height in pixels. The height cannot be greater than 100 pixels.

6 The **Width** property enables you to specify the width in pixels.

For information about the **Class** property, see **“Applying Classes” on page 59**.

7 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

8 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

9 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

10 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.1.7 Inline Frames

The **Inline Frame** object enables you to create a frame within a Page Layout.

"Tree Controls and Inline Frames" on page 150 describes a strategy for using the **Tree** object and the **Inline Frame** object together as the control mechanism for a Page Flow.

To create an inline frame

1 From the **HTML Objects** palette, drag the **Inline Frame** icon onto the canvas.

The frame appears as a blue square.

2 If desired, move or resize the frame.

3 The **Align** property enables you to specify how the frame should be aligned with the surrounding text.

4 If you want the frame to have a border, set the **Frame Border** property to **1**.

5 The **Height** property enables you to override the default height of the frame. The value is measured in pixels.

6 The **Margin Height** property specifies the number of blank pixels in the top and bottom margins of the frame.

7 The **Margin Width** property specifies the number of blank pixels in the left and right margins of the frame.

8 You can use the **Name** property to associate a **Link** object with the **Inline Frame** object. Set the value of the **Link** object's **Target** property to the value of the **Inline Frame** object's **Name** property.

9 The **Scrolling** property lets you control the behavior of scroll bars. The default value is **auto**, which means that scroll bars appear only when necessary. If you want scroll bars to always appear, set the value to **yes**. If you want scroll bars to never appear, set the value to **no**.

10 The **Src** property enables you to specify a URL whose contents will appear in the frame.

11 The **Width** property enables you to override the default width of the frame. The value is measured in pixels.

For information about the **Class** property, see **“Applying Classes” on page 59**.

12 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **“Applying a Local Style” on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

13 The **Long Description** property enables you to specify a URL that provides additional information about the frame.

14 The **Title** property enables you to enter a tooltip.

5.1.8 Divisions

The **Div** object enables you to create an area called a division. You can drop other objects into the division and format them with the same styles.

To create a division

1 From the **HTML Objects** palette, drag the **Div** icon onto the canvas.

2 If desired, move or resize the container.

3 Place other objects inside the container.

4 The **Align** property enables you to specify the horizontal alignment.

5 The **Auto Grow** property indicates whether the container resizes automatically when you add objects. The default value is **true**. If you want to disable this behavior, then set the value to **false**.

For information about the **Class** property, see **“Applying Classes” on page 59**.

6 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **“Applying a Local Style” on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

7 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

8 The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

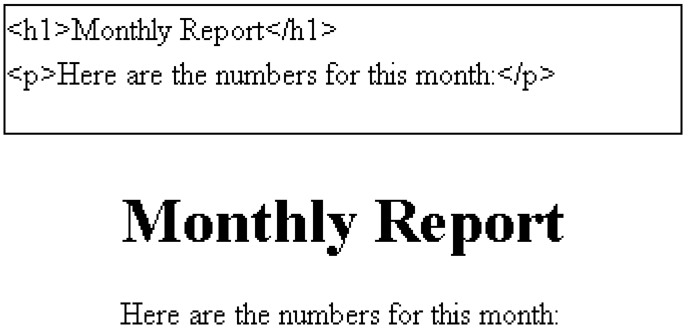
9 The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code.

5.1.9 HTML Containers

The **HTML** object enables you to create a container for HTML code.

The upper portion of Figure 39 shows an example of this object on the design canvas. The lower portion of Figure 39 shows an example of this object in Preview mode.

Figure 39 HTML Container (Design Canvas and Preview Mode)



```
<h1>Monthly Report</h1>
<p>Here are the numbers for this month:</p>
```

Monthly Report

Here are the numbers for this month:

To create an HTML container

1. From the **HTML Objects** palette, drag the **HTML** icon onto the canvas.
2. In the **Properties** tab, enter the HTML code:

   A. In the left column, select the **Text** property.

   B. Place your cursor over the existing value and click the **Command** button (...). The **Enter value for "Text"** dialog box appears.

   C. Enter the HTML code.

   D. Click **OK**.

3. The **Align** property enables you to specify the horizontal alignment.

   For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

4. The **Container** property enables you to specify the HTML tag in which the code will be located.

   For information about the **Class** property, see **"Applying Classes" on page 59**.

5. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **"Applying a Local Style" on page 57**.

   For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

6. The **Title** property enables you to enter a tooltip.
7. For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

8 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

9 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.2 Form Objects

The **Form Objects** palette contains pre-built combinations of Java classes and JSP code representing web interface entities. These objects enable users to communicate with Page Flows in the runtime environment. You can also access these objects from the **All Objects** palette.

Table 10 describes the form objects.

Table 10 Form Objects

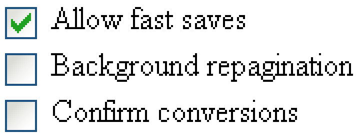
Object	Name	Purpose
☑	Checkbox Group	Enables you to create a group of check boxes. The user can select one or more check boxes in the group. For more information, see **"Check Box Groups" on page 90**.
▤	Drop-Down List	Enables you to create multiple user-selectable options. For more information, see **"Drop-Down Lists" on page 93**.
⬚	Hidden Input	Creates a hidden text field (a text field that is hidden from users at runtime), which you can use to pass session information to another page without being visible to users. For more information, see **"Hidden Input Elements" on page 99**.
🖼	Image Button	Creates an image "hotspot" that the user can click to perform an action, such as launching an application or jumping to another internal web page or an external web site. For more information, see **"Image Buttons" on page 102**.
***	Password	Creates a password input box with bullets that mask the password. For more information, see **"Password Boxes" on page 98**.
◉	Radio Group	Enables you to create a group of radio buttons. The user can select only one radio button in the group. For more information, see **"Radio Groups" on page 92**.

Object	Name	Purpose
⏻	Reset Button	Enables the user to cancel an operation or reset values to a default condition. For more information, see **“Reset Buttons” on page 100**.
↳	Submit Button	Enables the user to launch an operation or submit text to the application. For more information, see **“Submit Buttons” on page 101**.
▤	Text Area	Enables the user to enter text that will be displayed on the web page surrounded by a box. This object supports multiple lines of text. For more information, see **“Text Areas” on page 96**.
▭	Text Input Field	Enables the user to enter text that will be displayed on the web page surrounded by a box. This object supports one line of text. For more information, see **“Text Input Fields” on page 95**.
⬆	Upload	Enables the user to upload files and data to the web application. For more information, see **“Upload Objects” on page 97**.

5.2.1 Check Box Groups

The **Checkbox Group** object enables you to create a group of check boxes. The user can select one or more check boxes in the group. You specify the label and value for each check box.

Figure 40 Checkbox Group Example (Preview Mode)



☑ Allow fast saves
□ Background repagination
□ Confirm conversions

The **Checkbox Group** object contains the following property types: **Checkboxgroup**, **Checkboxitem**, **Checkbox**, and **Label**.

To create a check box group

1. From the **Form Objects** palette, drag the **Checkbox Group** object onto the canvas.
2. Right-click the object, and then click **Edit Options**.

The **Edit Options** dialog box appears.

Note: *You can also access this dialog box from the* **Edit Options** *property or by double-clicking the object.*

3 Select a **Label** field and type a label. Duplicate labels are not allowed.

The label will be displayed next to the check box at runtime.

4 In the **Value** field, type a value.

The value will be submitted when the user checks the box at runtime.

5 If you want this check box to be initially checked, select the check box in the **Checked** column.

6 To add additional check boxes to the group, click **Add** and repeat the previous steps.

7 Click **OK**.

8 By default, the check boxes are aligned vertically. The **Alignment** property enables you to align the check boxes horizontally.

For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

For information about the **Class** property, see **"Applying Classes" on page 59**.

9 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

10 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

11 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

12 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

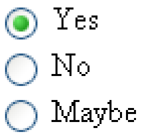
13 If you want to ensure that the user selects at least one check box in the group, set the **Validation Type** property to **isChecked**. The **Validation Msg** property specifies the error message that appears when the validation does not succeed.

Note: *You can drag a* **Checkbox Group** *object into a table cell and make the table row dynamic. At runtime, the object can be modified by external sources to generate additional cells containing check boxes.*

5.2.2 Radio Groups

The **Radio Group** object enables you to create a group of radio buttons. The user can select only one radio button in the group. You specify the label and value for each radio button.

Figure 41 Radio Group Example (Preview Mode)



● Yes
○ No
○ Maybe

The **Radio Group** object contains the following property types: **Radiogroup**, **Radioitem**, **Radio**, and **Label**.

To create a radio group

1 From the **Form Objects** palette, drag the **Radio Group** object onto the canvas.

2 Right-click the object, and then click **Edit Options**.

The **Edit Options** dialog box appears.

Note: *You can also access this dialog box from the* **Edit Options** *property or by double clicking the object.*

3 Select a **Label** field and type a label. Duplicate labels are not allowed.

The label will be displayed next to the button at runtime.

4 In the **Value** field, type a value.

This value will be submitted when the user clicks the button at runtime. This field cannot be empty. This object requires that a value be assigned in order to pass the true/false condition.

5 If you want this button to be initially selected, select the check box in the **Checked** column.

6 To create additional buttons in the group, click **Add** and repeat the previous steps.

7 Click **OK**.

8 By default, the radio buttons are aligned vertically. The **Alignment** property enables you to align the radio buttons horizontally.

For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

For information about the **Class** property, see **"Applying Classes" on page 59**.

9 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

10 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

11 The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

12 The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code.

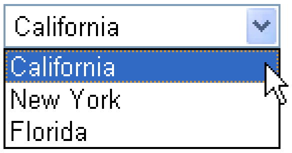
13 If you want to ensure that the user makes a selection, set the **Validation Type** property to **isChecked**. The **Validation Msg** property specifies the error message that appears when the validation does not succeed.

Note: *You can drag a* ***Radio Group*** *object into a table cell and make the table row dynamic. At runtime, the object can be modified by external sources to generate additional cells containing radio buttons.*

5.2.3 Drop-Down Lists

The **Drop-Down List** object enables you to create multiple user-selectable options. You specify the label and value for each option.

Figure 42 Drop-Down List (Preview Mode)



You can make a **Drop-Down List** object dynamic. Dynamic objects are extended or replicated using the Business Rule Designer.

The **Drop-Down List** object contains the following property types: **Select** and **Option**.

To create a drop-down list

1 From the **Form Objects** palette, drag the **Drop-Down List** object onto the canvas.

2 Right-click the object, and then click **Edit Options**.

The **Edit Options** dialog box appears.

Note: *You can also access this dialog box from the* **Edit Options** *property or by double-clicking the object.*

3 Click **Add**.

4 Select a **Label** field and type a label. Duplicate labels are not allowed.

The label will be displayed as an option to users at runtime.

5 In the **Value** field, type a value.

The value will be submitted when the user clicks the option at runtime.

6 If you want this option to be initially selected, select the check box in the **Selected** column.

7 To create additional options in the list, click **Add** and repeat the previous steps.

8 Click **OK**.

For information about the **Auto Size** property, see **“Automatically Resizing Page Layout Objects” on page 64**.

9 If you set the **Auto Submit** property to **true**, then the Page Layout is automatically submitted when the user selects an option at runtime.

10 If you want to make the object inactive, set the **Disabled** property to **true**.

11 If you want to allow the user to select more than one option, set the **Multiple** property to **true**.

12 The **Size** property enables you to change the number of options that are visible at a time. The default value is 1.

For information about the **Class** property, see **“Applying Classes” on page 59**.

13 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **“Applying a Local Style” on page 57**.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

14 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

15 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

16 The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

17 The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code.

18 If you want to ensure that the user makes a selection, set the **Validation Type** property to **isSelected**. The **Validation Msg** property specifies the error message that appears when the validation does not succeed.

19 To make the **Drop-Down List** object dynamic, do one of the following:

- Right-click the object, and then click **Make Dynamic**.
- Set the **Is Dynamic** property to **true**.

At runtime, a dynamic **Drop-Down List** object can be extended with additional labels and values generated by external sources at runtime. The **Edit Options** menu is disabled when you select **Make Dynamic**.

Note: *The values of dynamic objects do not appear when you click the* **Preview** *icon, because the values are assigned at runtime.*

5.2.4 Text Input Fields

The **Text Input Field** object enables the user to enter text that will be displayed on the web page surrounded by a box. This object supports one line of text. If you want the user to be able to enter more than one line of text, then create a **Text Area** object.

To create a text input field

1. From the **Form Objects** palette, drag the **Text Input Field** object onto the canvas.
2. If you want to make the text box inactive, set the **Disabled** property to **true**.
3. The **Max Length** property enables you to specify the maximum number of characters that users can enter.
4. If you want to prevent users from editing the text box, set the **Read Only** property to **true**.
5. The **Size** property enables you to specify the size of the input element, measured in number of characters.
6. The **Type** property specifies the type of form object. This property is read only.
7. The **Value** property enables you to enter default text.

   For information about the **Class** property, see **“Applying Classes” on page 59**.

8. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **“Applying a Local Style” on page 57**.

   For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

9. If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.
10. If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.
11. The **Title** property enables you to enter a tooltip.

    For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

12. The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

13 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

14 The **Validation Type** property enables you to validate the text that the user enters. For some of the functions, you must enter an argument in the **Validation Args** property. For all of the functions, the **Validation Msg** property enables you to specify the error message that appears when the text is invalid. If you do not want an empty text box to be validated, then set the **Validate Empty** property to **false**.

Table 11 describes the validation functions.

Table 11 Validation Functions

Function	Description
isMaximumLength	Ensures that the text does not exceed a maximum number of characters. Enter the number in the **Validation Args** property.
isMinimumLength	Ensures that the text is a minimum number of characters. Enter the number in the **Validation Args** property.
isRequired	Ensures that the text is not blank.
isValidInt	Ensures that the text is a valid whole number (positive or negative).
isValidRegex	Enables you to specify a custom validation function using a JavaScript regular expression. Enter the regular expression in the **Validation Args** property. For example, the following regular expression checks whether the text consists of three numbers, followed by a hyphen, followed by two numbers, followed by a hyphen, followed by four numbers: **\d{3}-\d{2}-\d{4}**

5.2.5 Text Areas

The **Text Area** object enables the user to enter text that will be displayed on the web page surrounded by a box. This object supports multiple lines of text. If you want the user to be able to enter only one line of text, then create a **Text Input Field** object.

You can use the style sheet functionality of the Page Layout Designer to force the text to remain on a single line, or to create a horizontal scrollbar:

```
white-space:nowrap;overflow:scroll;
```

To create a text area

1 From the **Form Objects** palette, drag the **Text Area** object onto the canvas.

2 The **Columns** property enables you to specify the width of the text area in columns.

3 If you want to make the text area inactive, set the **Disabled** property to **true**.

4 If you want to prevent users from modifying the text area, set the **Read Only** property to **true**.

5 The **Row** property enables you to specify the height of text area in rows.

6 The **Text** property enables you to enter default text.

For information about the **Class** property, see **"Applying Classes" on page 59**.

7 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

8 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

9 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

10 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

11 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

12 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

For information about the validation properties, see **"Text Input Fields" on page 95**.

5.2.6 Upload Objects

The **Upload** object enables the user to upload a file to the web server. The object consists of a text field and a browse button.

The user can specify the file in one of two ways:

- Enter the fully qualified name of the file in the text field
- Click the browse button to navigate the file system and select the file

When you add the **Upload** object to a Page Layout, you must set the Page Layout's **Enctype** property to **multipart/form-data**. The **Enctype** property is located in the **Form** property type.

When the user uploads a file, eVision Studio places the file in the following directory:

```
Logical_Host_install_dir\is\domains\domain_name\applications\j2ee-
apps\icanData\eVision\project_name\documents
```

To create an upload object

1 From the **Form Objects** palette, drag the **Upload** object onto the canvas.

2 The **Accept** property enables you to specify a comma-separated list of valid MIME types that can be uploaded (for example, **text/html,image/gif**).

3 If you want to make the object inactive, set the **Disabled** property to **true**.

4 The **Size** property enables you to specify the size of the input element, measured in number of characters.

5 The **Type** property specifies the type of form object. This property is read only.

For information about the **Class** property, see **"Applying Classes" on page 59**.

6 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

7 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

8 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

9 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

10 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

11 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.2.7 Password Boxes

The **Password** object enables the user to enter a password into a text field. The password is masked with bullets.

To create a password box

1 From the **Form Objects** palette, drag the **Password** object onto the canvas.

2 If you want to make the object inactive, set the **Disabled** property to **true**.

3 The **Max Length** property enables you to specify the maximum number of characters that users can enter.

4 The **Size** property enables you to specify the size of the input element, measured in number of characters.

5 The **Type** property specifies the type of form object. This property is read only.

For information about the **Class** property, see **"Applying Classes" on page 59**.

6 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

7 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

8 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

9 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

10 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

11 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

12 By default, the password is not validated. If you want to validate the password, set the **Validation Type** property to the appropriate function. For example, you can ensure that the password is numeric by setting this property to **isValidInt**.

The **Validation Msg** property specifies the error message that appears when the password is invalid. For example, if you are ensuring that the password is numeric, you could set this property to **Please enter a numeric password**.

13 If you do not want an empty text field to be validated, then set the **Validate Empty** property to **false**.

5.2.8 Hidden Input Elements

The **Hidden Input** object creates an invisible field in the Page Layout. You can use this object to pass session information to another page without being visible to users. The object appears on the design canvas, but not in Preview mode or at runtime.

To create a hidden element

1 From the **Form Objects** palette, drag the **Hidden Input** object onto the canvas.

For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

2 The **Type** property specifies the type of form object. This property is read only.

3 The **Value** property specifies the value that is submitted.

4 The **ID** property contains the unique HTML identifier for the object. This property is read only.

5.2.9 Reset Buttons

The **Reset Button** object enables the user to change the settings of other form objects on the same page to their original values.

For example, assume that the **Calendar** object initially has no value in the text box. The user clicks the drop-down arrow and selects the date 07/14/2005. If the user clicks the **Reset Button** object, the date is removed from the text box.

To create a reset button

1. From the **Form Objects** palette, drag the **Reset Button** object onto the canvas.

   For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

2. If you want to make the button inactive, set the **Disabled** property to **true**.

3. The **Type** property specifies the type of form object. This property is read only.

4. The **Value** property specifies the text that appears on the button. If desired, change the default value.

   For information about the **Class** property, see **"Applying Classes" on page 59**.

5. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **"Applying a Local Style" on page 57**.

   For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

6. If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

7. If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

8. The **Title** property enables you to enter a tooltip.

   For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

9. The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

10. The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.2.10 Submit Buttons

The **Submit Button** object enables the user to send form data on the Page Layout to the web application.

To create a submit button

1 From the **Form Objects** palette, drag the **Submit Button** object onto the canvas.

For information about the **Auto Size** property, see **"Automatically Resizing Page Layout Objects" on page 64**.

2 If you want to make the button inactive, set the **Disabled** property to **true**.

3 The **Type** property specifies the type of form object. This property is read only.

4 The **Value** property specifies the text that appears on the button. If desired, change the default value.

For information about the **Class** property, see **"Applying Classes" on page 59**.

5 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

6 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

7 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

8 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

9 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

10 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

Back-End System Scenario

The **Submit Button** object is often used for a scenario in which you want to send data entered by the user to a back-end system.

To help ensure that the data is successfully sent to the back-end system, create two Page Layouts.

- The first Page Layout contains the **Submit Button** object.
- The second Page Layout is a confirmation page. Add a message such as the following:

```
Your data has been submitted.
```

Below the message, add a **Close** button, which enables the user to close the browser window.

If the Page Layouts are located within a sub-process, see **"Close Buttons" on page 113** for information about the behavior of the **Close** button.

5.2.11 Image Buttons

The **Image Button** object enables you to create a graphical submit button.

"Image Buttons" on page 153 describes how the mouse coordinates of where the user clicked are exposed.

Note: *Before you begin, you must import the image into the Project. For more information, see* **"Importing Files" on page 46**.

To create an image button

1. From the **Form Objects** palette, drag the **Image Button** object onto the canvas.

   The **Enter value** dialog box appears.
2. Select the image and click **Open**.

   The image appears on the canvas.
3. If you want to make the button inactive, set the **Disabled** property to **true**.
4. The **Height** property enables you to override the default height of the image. The value is measured in pixels.
5. The **Mouse Over Src** property enables you to specify a image that appears when the user places the mouse pointer over the button.
6. The **Src** property enables you to change the image.
7. The **Type** property specifies the type of form object. This property is read only.
8. The **Value** property specifies the value that is submitted.
9. The **Width** property enables you to override the default width of the image. The value is measured in pixels.

   For information about the **Class** property, see **"Applying Classes" on page 59**.

10 The **ID** property contains the unique HTML identifier for the object. This property is read only.

11 For information about the **Style** property, see **"Applying a Local Style" on page 57**.

12 For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

13 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

14 The **Alt** property enables you to specify replacement text for the image.

15 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

16 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

17 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

18 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.3 Extension Objects

The **Extension Objects** palette contains objects that are neither HTML nor form objects. You can also access these objects from the **All Objects** palette.

Table 12 describes the extension objects.

Table 12 Extension Objects

Object	Name	Purpose
	Calendar	Enables the user to select a calendar date. The object consists of a text field and a drop-down arrow. For more information, see **“Calendars” on page 107**.
	Chart	Creates a two-dimensional chart. eVision Studio provides a variety of predefined chart types, including area charts, bar charts, line charts, and pie charts. For more information, see **Chapter 7** **“Creating Charts”**.
	Close	Enables the user to close the browser window. The Page Flow is automatically ended. For more information, see **“Close Buttons” on page 113**.
	Cookie	Enables an eVision Studio application to store information on a client computer, and to examine that information at a later time. For more information, see **“Cookie Objects” on page 116**.
	If	Enables you to conditionally display an area of a Page Layout. For more information, see **“If Objects” on page 105**.
	Logout	Enables the user to exit the eVision Studio web application. Clicking this button returns the user to the beginning of the Page Flow. For more information, see **“Logout Buttons” on page 115**.
	Progress Bar	Enables you to illustrate the percentage of completion of an action. For more information, see **“Progress Bars” on page 108**.

Table 12 Extension Objects

Object	Name	Purpose
	SVG	Enables you to configure portions of a graphic to take actions at runtime based on dynamic values and conditions. For more information, see **"SVG Objects" on page 117**.
	Switch	Enables you to conditionally display an area of a Page Layout. For more information, see **"Switch Objects" on page 106**.
	Tab Set	Enables you to create a group of tabbed pages. For more information, see **"Tab Sets" on page 109**.
	Tree	Enables you to create a tree control, which is useful for the navigation of hierarchical data. For more information, see **"Tree Controls" on page 111**.

5.3.1 If Objects

The **If** object enables you to conditionally display an area of a Page Layout.

After you drag the object onto the canvas, you add content. When the **Value** property is set to **true**, the content inside this object appears at runtime. When the **Value** property is set to **false**, the content does not appear at runtime. You can set this property statically in the Page Layout Designer, or dynamically in the Page Flow Designer.

The **If** object contains the following property types:

Table 13 If Property Types

Property Type	Description
If	The main properties.
Divif	Properties for the object's container.

For information about the **Class** property, see **"Applying Classes" on page 59**.

The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.

The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

The **Language** property enables you to specify the base language of the object’s content. Enter a valid language code.

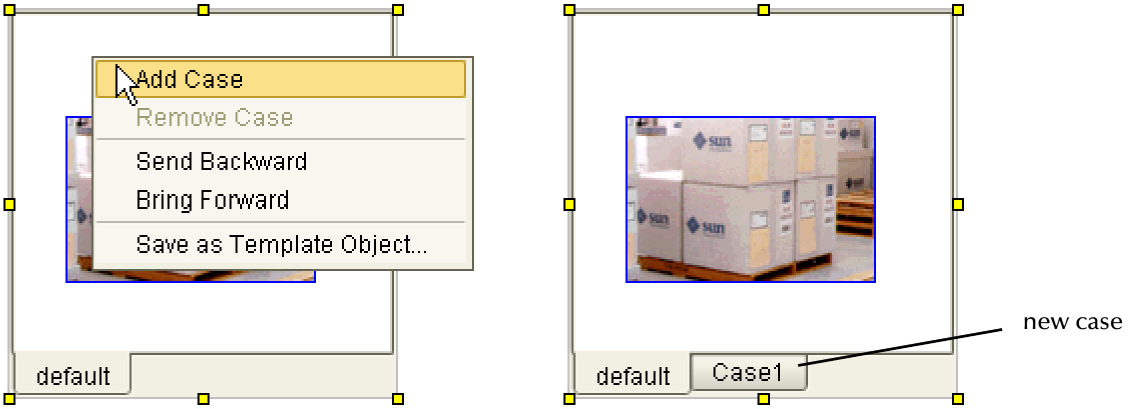
You can resize the **If** object horizontally or vertically by using the mouse.

5.3.2 Switch Objects

The **Switch** object enables you to conditionally display an area of a Page Layout. You specify two or more cases and add different content to each case.

To add a case, right-click the object and select **Add Case**.

Figure 43 Adding a Case to the Switch Object



The **Switch** object contains the following property types:

Table 14 Switch Property Types

Property Type	Description
Switch	Properties for the overall object.
Default	Properties for the default case.
Case	Properties for a non-default case.

The **Value** property of the **Switch** property type specifies the name of the case that appears at runtime. You can set this property statically in the Page Layout Designer, or dynamically in the Page Flow Designer.

The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.

For a non-default case, the **Case** property of the **Case** property type enables you to change the name of the case.

You can resize the **Switch** object horizontally or vertically by using the mouse.

To remove a case, right-click the case and select **Remove Case**. You cannot remove the default case.

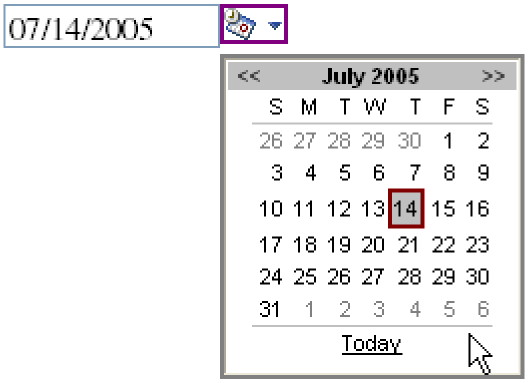
5.3.3 Calendars

The **Calendar** object enables the user to select a calendar date. The object consists of a text field and a drop-down arrow.

When the user clicks the drop-down arrow, the current month appears. The user can scroll back to previous months and scroll forward to succeeding months. When the user clicks a date, the date appears in the text box.

Figure 44 shows how this object appears in Preview mode.

Figure 44 Calendar Example (Preview Mode)



07/14/2005

S	M	T	W	T	F	S
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

The **Calendar** object contains the following property types:

Table 15 Calendar Property Types

Property Type	Description
Divcalendar	Properties for the object's container.
Calendar	The main properties.

To create a calendar

1. From the **Extension Objects** palette, drag the **Calendar** object onto the canvas.
2. In the **Properties** tab, select the **Calendar** property type.
3. If you want to make the calendar inactive, set the **Disabled** property to **true**.
4. The **Max Length** property enables you to specify the maximum number of characters that users can enter in the text field.
5. If you want to prevent users from modifying the text field, set the **Read Only** property to **true**.

6 The **Type** property is read only.

7 If you want the calendar to automatically display the current date, set the **Value** property to **today**.

For information about the **Class** property, see **"Applying Classes" on page 59**.

8 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

9 If you want to assign a keyboard shortcut to the object, enter the letter or number as the value of the **Access Key** property. The user invokes the shortcut by pressing a key (such as Alt or Ctrl) plus the character that you specify.

10 The **Alt** property enables you to specify replacement text for the image.

11 If you have multiple form objects on a page, you can use the **Tab Index** property to specify the order in which the form objects are selected when the user presses the Tab key.

12 The **Title** property enables you to enter a tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

13 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

14 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

15 By default, the selected date appears in MM/dd/yyyy format. The **Validation Args** property enables you to change the format to dd/MM/yyyy.

The **Validation Msg** property specifies the error message that appears when the user enters an invalid date. If desired, change the default error message.

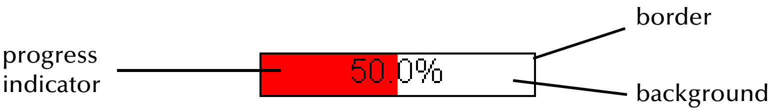
If you do not want an empty text field to be validated, then set the **Validate Empty** property to **false**.

The **Validation Type** property is read only.

5.3.4 Progress Bars

The **Progress Bar** object enables you to illustrate the percentage of completion of an action. The progress bar has a horizontal orientation. Figure 45 shows how this object appears in Preview mode.

Figure 45 Progress Bar Example (Preview Mode)



You specify the data for the progress bar in the Page Flow Designer, rather than the Page Layout Designer.

To create a progress bar

1. From the **Extension Objects** palette, drag the **Progress Bar** object onto the canvas.
2. This object has the following color-related properties. If desired, change the default color of one or more properties.
   - The **Background Color** property sets the color of the area in which the progress indicator expands.
   - The **Border Color** property sets the color of the outer border.
   - The **Progress Color** property sets the color of the progress indicator.
3. You do not need to modify the **Percent Complete** property in the Page Layout Designer. When you place the Page Layout in a Page Flow, create a business rule to specify the data that the progress bar will measure.

   For example, assume that Page Layout 1 contains a text box and Page Layout 2 contains a progress bar. Create a business rule between the Page Layouts. In the Business Rule Designer, map the **value** node of the text box to the **Percent Complete** node of the progress bar.

   Chapter 6 "Using the Page Flow Designer" describes how to use the Business Rule Designer.
4. The **ID** property contains the unique HTML identifier for the object. This property is read only.

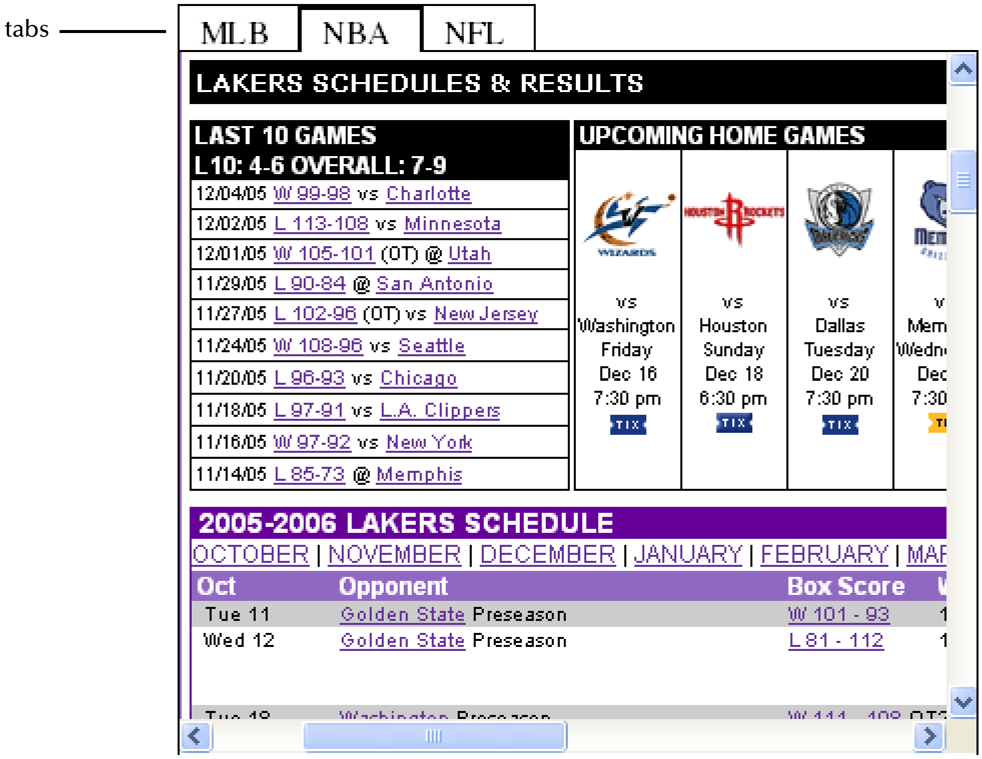
   For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.
5. Set the Page Layout in which the progress bar is located to automatically refresh after a brief period of time (for example, 2 seconds). **"Automatically Refreshing Page Layouts" on page 64** describes how to perform this task.

5.3.5 Tab Sets

The **Tab Set** object enables you to create a group of tabbed pages. The tab portion appears on the top.

You can place one or more objects on a tabbed page. For example, you could place an **Inline Frame** object on a tabbed page and then set the **Src** property of the frame to an external URL.

Figure 46 Tab Set Example (Preview Mode)



LAKERS SCHEDULES & RESULTS

LAST 10 GAMES L10: 4-6 OVERALL: 7-9
12/04/05 W 99-98 vs Charlotte
12/02/05 L 113-108 vs Minnesota
12/01/05 W 105-101 (OT) @ Utah
11/29/05 L 90-84 @ San Antonio
11/27/05 L 102-96 (OT) vs New Jersey
11/24/05 W 108-96 vs Seattle
11/20/05 L 96-93 vs Chicago
11/18/05 L 97-91 vs L.A. Clippers
11/16/05 W 97-92 vs New York
11/14/05 L 85-73 @ Memphis

UPCOMING HOME GAMES

vs Washington Friday Dec 16 7:30 pm TIX	vs Houston Sunday Dec 18 6:30 pm TIX	vs Dallas Tuesday Dec 20 7:30 pm TIX	v Mem Wedn Dec 7:30 TI

2005-2006 LAKERS SCHEDULE

OCTOBER | NOVEMBER | DECEMBER | JANUARY | FEBRUARY | MAR

Oct	Opponent	Box Score	W
Tue 11	Golden State Preseason	W 101 - 93	1
Wed 12	Golden State Preseason	L 81 - 112	1
Tue 18	Washington Preseason	W 111 - 108 OT	

To create a tab set

1 From the **Extension Objects** palette, drag the **Tab Set** icon onto the canvas.

2 Right-click the object and select **Edit Options**.

The **Edit Options** dialog box appears.

3 In the **Label** column, enter the tab name.

4 If you want this tabbed page to be initially displayed, select the check box in the **Default** column.

5 To create additional tabbed pages, click **Add** and repeat the previous steps.

6 Click **OK**.

7 The **Tabset** property type contains properties for the entire group of tabbed pages. If desired, change one or more properties:

A The **Active Background Color** property specifies the background color for the currently selected tab. The **Active Font Color** property specifies the color of the label for the currently selected tab.

B The **Border Color** property specifies the color of the outside border.

C The **Edit Options** property enables you to add or remove tabbed pages.

D The **Font Family** property specifies the type of font for the labels. The **Font Size** specifies the size of the labels.

E The **HAlign** property specifies the horizontal alignment of the tabs (left, right, or center).

F The **Hover Background Color** specifies the background color of a tab when the mouse pointer is placed over the tab. The **Hover Font Color** specifies the color of the label when the mouse pointer is placed over the tab.

G The **Inactive Background Color** property specifies the background color for all tabs that are not currently selected. The **Inactive Font Color** property specifies the color of the label for all tabs that are not currently selected.

H The **Margin Left** property specifies the number of blank pixels to the left of the tab. The **Margin Right** property specifies the number of blank pixels to the right of the tab.

I The **Show Borders** property specifies whether the tabs object has an outside border.

J The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

8 The **Tab** property type contains properties for an individual tabbed page. If desired, change one or more properties:

A The **Active Image Src** property enables you to specify an image file that will be used for the currently selected tab.

B The **Default** property specifies whether the tabbed page is initially displayed.

C The **Inactive Image Src** property enables you to specify an image file that will be used for all tabs that are not currently selected.

D The **Text** property specifies the tab name.

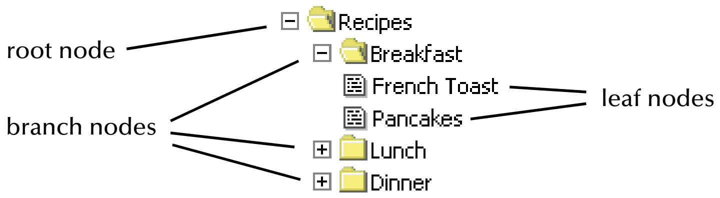
5.3.6 Tree Controls

The **Tree** object enables you to create a tree control, which is useful for the navigation of hierarchical data.

The tree always has one root node. The tree can contain any number of branch and leaf nodes.

Figure 47 shows an example of this object in Preview mode.

Figure 47 Tree Example (Preview Mode)



You associate each leaf node with a Page Link or a URL. When users click a leaf node, the corresponding information is displayed.

The number of nodes in a tree can be static or dynamic.

If you are using Internet Explorer and Windows XP Service Pack 2, then Internet Explorer may display the following message when you try to preview a Page Layout that contains a **Tree** object: **To help protect your security, Internet Explorer has restricted this file from showing active content that could access your computer.** This message appears because the preview functionality of the Page Layout Designer is based on a locally generated HTML file. The runtime behavior of the **Tree** object is not affected.

"Tree Controls and Inline Frames" on page 150 describes a strategy for using the **Tree** object and the **Inline Frame** object together as the control mechanism for a Page Flow.

To create a tree control

1. From the **Extension Objects** palette, drag the **Tree** icon onto the canvas.
2. Right-click the object and select **Edit Options**.

   The **Edit Options** dialog box appears.

Note: *You can also access this dialog box from the* **Edit Options** *property.*

3. To add a node to the tree, select the parent node and click **Add Node**.
4. To delete a node from the tree, right-click the node and select **Delete Node**. You cannot delete the root node.
5. Click **OK**.
6. For each node, you can edit the following properties:

   A The **LName** property is the internal name for the object. This name appears in the mapping area of the Page Flow Designer.

   B To make the node a repeating node, set the **Is Repeatable** property to true. When you configure the Page Flow in which the Page Layout appears, you map data into this node. The number of times that this node appears is determined at runtime. This property is not available for the root node.

   C The **Text** property specifies the node name that appears to the user.
7. For each node link, you can edit the following properties:

Note: *Typically, you set either the* **HRef** *property or the* **Link** *property. If you set both properties, then the* **HRef** *property takes precedence.*

   A The **HRef** property specifies the URL that is associated with the node.

   B The **Link** property specifies the Page Link that is associated with the node.

   C The **Target** property controls how the information is displayed when the user clicks the node. If you want to display the information in an **Inline Frame** object, enter the **Name** value of the **Inline Frame** object. Another approach is to select one of the predefined values.

D For information about the **Class** property, see **"Applying Classes" on page 59**.

E The **ID** property contains the unique HTML identifier for the object. This property is read only.

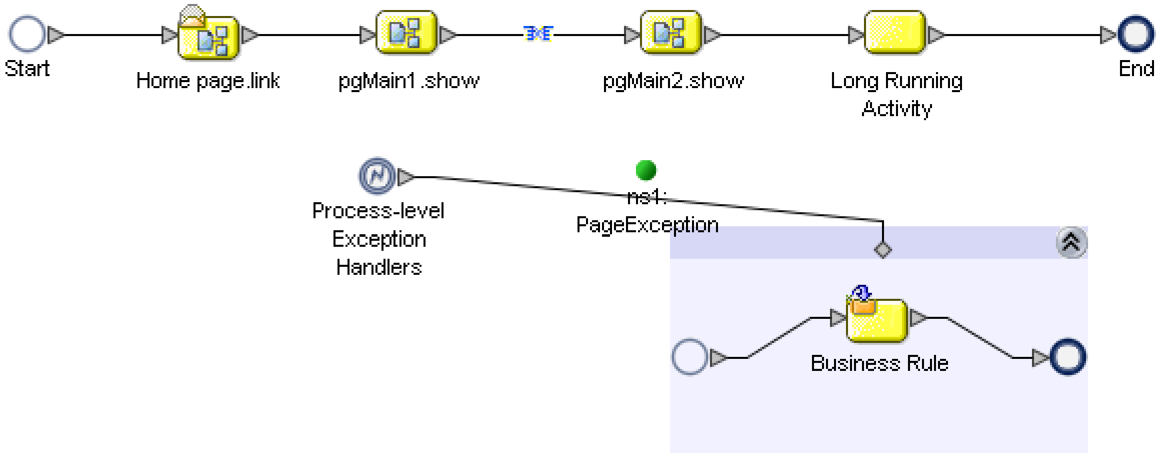
5.3.7 Close Buttons

When a user presses the **Close** button, eVision Studio sends an exception for pending requests to the Page Flow, ends the session, and closes the browser.

If you want the Page Flow to continue after the user presses the **Close** button, then you must catch the exception.

Figure 48 shows a Page Flow in which the second Page Layout contains a **Close** button. The second Page Layout is followed by a long-running activity. The Page Flow includes a **Catch Named Exception** element, which catches the exception sent by the second Page Layout and performs appropriate business logic.

Figure 48 Catching Named Exception in Page Flow with Close Button



For more information about exception handling, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.

Note: *If a session timeout occurs, eVision Studio sends a page exception and invalidates the session. By default, the session timeout is 60 minutes.* **"Linking Each Service to a Web Connector" on page 158** *describes how to change the default value.*

The **Close** object contains the following property types:

Table 16 Close Property Types

Property Type	Description
Img	Properties that are specific to the button image.
Close	All other properties.

To create a close button

1. From the **Extension Objects** palette, drag the **Close** icon onto the canvas.
2. In the **Properties** tab, select the **Img** property type.
3. The **Border** property enables you to add a border to the image. The default value of 0 indicates the absence of a border. The value is measured in pixels.
4. If you want the My Pictures image toolbar to appear when the mouse pointer is placed over the **Close** button, set the **Gallery Image** property to **yes**. This property is specific to Internet Explorer.

Note: *The* **Page** *property type includes a property called* **Image Toolbar***, which specifies the behavior of the My Pictures image toolbar for the overall Page Layout. Setting the* **Gallery Image** *property overrides the* **Image Toolbar** *setting.*

5. The **Height** property enables you to override the default height of the image. The value is measured in pixels.
6. If you want to change the default image, do the following:

   A. Import the image that you want to use into your Project. For more information, see **“Importing Files” on page 46**.

   B. In the **Properties** tab, click the **Src** property.

   C. Place your cursor over the existing value and click the **Command** button (...). The **Enter value for “Src”** dialog box appears.

   D. Navigate to the image that you want to use.

   E. Click **Open**.
7. The **Width** property enables you to override the default width of the image. The value is measured in pixels.

   For information about the **Class** property, see **“Applying Classes” on page 59**.
8. The **ID** property contains the unique HTML identifier for the object. This property is read only.

   For information about the **Style** property, see **“Applying a Local Style” on page 57**.

   For information about the **Z-Index** property, see **“Configuring the Z-Direction of Objects” on page 49**.
9. The **Alt** property enables you to specify replacement text for the image.
10. The **Long Description** property enables you to specify a URL that provides additional information about the image (beyond that provided by the **Alt** property).
11. The **Title** property enables you to change the tooltip.

    For information about the JavaScript properties, see **“Event Handler Properties” on page 60**.
12. The **Direction** property enables you to specify the base direction of the object’s content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

13 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.3.8 Logout Buttons

When a user presses the **Logout** button, eVision Studio sends an exception for pending requests to the Page Flow, invalidates the session, and returns the user to the beginning of the Page Flow.

If you want the Page Flow to continue after the user presses the **Logout** button, then you must catch the exception. For information about exception handling, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.

Note: *If a session timeout occurs, eVision Studio sends a page exception and invalidates the session. By default, the session timeout is 60 minutes.* **"Linking Each Service to a Web Connector" on page 158** *describes how to change the default value.*

The **Logout** object contains the following property types:

Table 17 Logout Property Types

Property Type	Description
Img	Properties that are specific to the button image.
Logout	All other properties.

To create a logout button

1 From the **Extension Objects** palette, drag the **Logout** icon onto the canvas.

2 In the **Properties** tab, select the **Img** property type.

3 The **Border** property enables you to add a border to the image. The default value of 0 indicates the absence of a border. The value is measured in pixels.

4 If you want the My Pictures image toolbar to appear when the mouse pointer is placed over the **Logout** button, set the **Gallery Image** property to **yes**. This property is specific to Internet Explorer.

Note: *The* **Page** *property type includes a property called* **Image Toolbar***, which specifies the behavior of the My Pictures image toolbar for the overall Page Layout. Setting the* **Gallery Image** *property overrides the* **Image Toolbar** *setting.*

5 The **Height** property enables you to override the default height of the image. The value is measured in pixels.

6 If you want to change the default image, do the following:

A Import the image that you want to use into your Project. For more information, see **"Importing Files" on page 46**.

B In the **Properties** tab, click the **Src** property.

C Place your cursor over the existing value and click the **Command** button (...). The **Enter value for "Src"** dialog box appears.

D Navigate to the image that you want to use.

E Click **Open**.

7 The **Width** property enables you to override the default width of the image. The value is measured in pixels.

For information about the **Class** property, see **"Applying Classes" on page 59**.

8 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.

9 The **Alt** property enables you to specify replacement text for the image.

10 The **Long Description** property enables you to specify a URL that provides additional information about the image (beyond that provided by the **Alt** property).

11 The **Title** property enables you to change the tooltip.

For information about the JavaScript properties, see **"Event Handler Properties" on page 60**.

12 The **Direction** property enables you to specify the base direction of the object's content. Use the value **ltr** for left-to-right, or the value **rtl** for right-to-left.

13 The **Language** property enables you to specify the base language of the object's content. Enter a valid language code.

5.3.9 Cookie Objects

The **Cookie** object enables an eVision Studio application to store information on a client computer, and to receive that information at a later time. For example, the application might want to store a unique ID for the client. The information is stored in a text file.

RFC 2109, "HTTP State Management Mechanism," is the specification for cookies.

To create a cookie object

1 From the **Extension Objects** palette, drag the **Cookie** icon onto the canvas.

2 The **Name** property specifies the name of the cookie. This property is required.

3 The **Value** property specifies the value of the cookie. This property is required.

4 The **Comment** property enables you to document the intended use of the cookie.

5 The **Domain** property enables you to specify the domain for which the cookie is valid.

6 The **Max Age** property enables you to specify how long the cookie remains in existence. The value is expressed in seconds. By default, the cookie is deleted when the web browser exits.

7 The **Path** property enables you to specify the path on the web server where the web browser returns the cookie.

8 If the **Secure** property is set to **true**, then the web browser sends the cookie to the web server only over a secure protocol. If the **Secure** property is set to **false**, then the web browser sends the cookie to the web server over any protocol.

9 The **Version** property specifies the version of the state management specification that the cookie complies with.

10 The **ID** property contains the unique HTML identifier for the object. This property is read only.

5.3.10 SVG Objects

The **SVG** object enables you to configure portions of a graphic to take actions at runtime based on dynamic values and conditions.

For example, you could include a world map that shows daily sales totals for various countries, and configure the map such that a visual cue indicates when the sales volume for a country is less than a specified amount.

The image must be in the Scalable Vector Graphics (SVG) format. The SVG specification provides an XML-based language for describing two-dimensional graphics. For detailed information, go to http://www.w3.org/Graphics/SVG/.

"SVG Objects" on page 155 describes how to use this object in the Page Flow Designer.

Note: *Before you begin, you must import the image into the Project. For more information, see* **"Importing Files" on page 46**.

To create an SVG object

1 From the **Extension Objects** palette, drag the **SVG** icon onto the canvas.

The **Enter value** dialog box appears.

2 Select the image and click **Open**.

3 The **Src** property enables you to change the image.

4 For information about the **Class** property, see **"Applying Classes" on page 59**.

5 The **ID** property contains the unique HTML identifier for the object. This property is read only.

For information about the **Style** property, see **"Applying a Local Style" on page 57**.

5.4 Custom Objects

Chapter 8 "Creating Custom Objects" on page 183 describes how to create custom objects.

Chapter 6

Using the Page Flow Designer

You use the Page Flow Designer to create an eVision Studio web application, which consists of Page Layouts and specialized design elements that are linked together in a Page Flow.

What's in This Chapter

6.1 Page Flow Designer Overview

A *Page Flow is a structured series of web pages that comprise a web-enabled business process.*

A web-enabled business process can be an internal service, or it can be exposed as an external application over the web. A Page Flow can involve a variety of participants, and may include internal and external computer systems. When you create a Page Flow, you are creating a graphical representation of what will become a fully functional and deployable web application.

6.1.1 Adding a Page Flow to a Project

When you add a Page Flow to a Project, the Page Flow Designer opens a design canvas that enables you to add Page Layouts and other design elements and then connect them together in a sequence.

Before you can start building a Page Flow, you must add the Page Flow to your Project.

The Page Flow Designer allows you to cut, copy, and paste Page Flows.

To add a Page Flow to a Project

1. In the Project Explorer of Enterprise Designer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page Flow**.
3. If you want to change the default name of the Page Flow, click the Page Flow name in the Project Explorer and then enter the new name.

6.1.2 Adding Components to a Page Flow

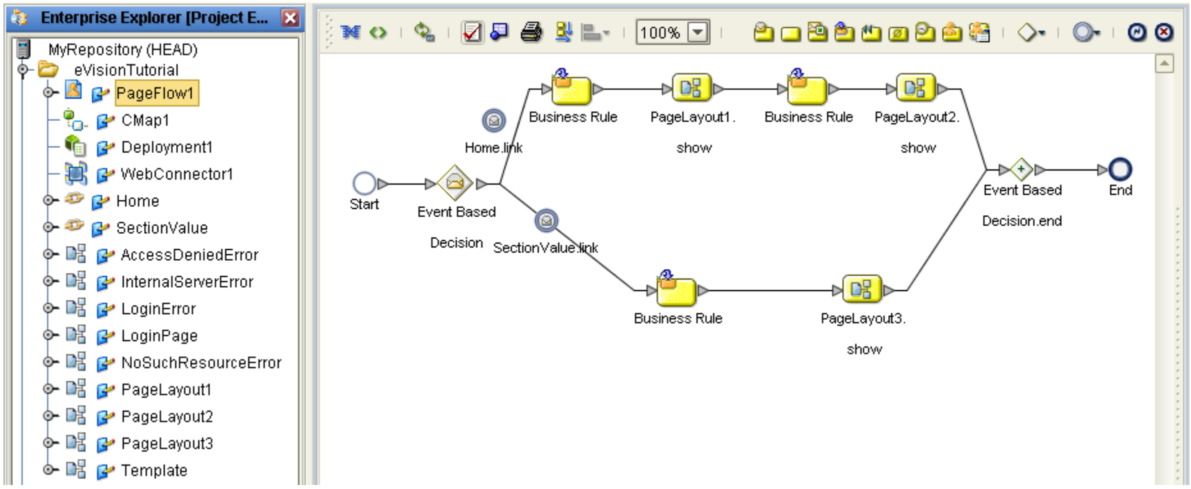
After you add a Page Flow to a Project, you drag and drop Page Flow elements and web service operations onto the canvas. Web service operations include:

- The **show** operation of a Page Layout
- The **link** operation of a Page Link

You then link the components together.

The Page Flow Designer provides the tools to lay out and connect Page Flow elements and other specialized elements. Figure 49 shows an example of the Page Flow Designer interface.

Figure 49 Page Flow Example in the Page Flow Designer



When you start a new web application project, the **Start** and **End** icons automatically appear on the blank Page Flow Designer canvas. The Page Flow can have only one starting point. The Page Flow can have multiple end points.

To add components to a Page Flow

1. Drag the desired Page Flow elements and web service operations onto the Page Flow Designer canvas.
2. Create links between the elements.
3. On the **File** menu, click **Save All**.

   This action validates the connectivity of the Page Flow, generates the code to run it, and saves the Page Flow in the Repository.

6.1.3 Page Flow Designer Toolbar

Figure 50 shows the Page Flow Designer toolbar.

Figure 50 Page Flow Designer Toolbar

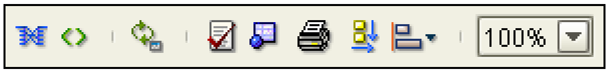


Table 18 describes each tool.

Table 18 Page Flow Designer Tools

Tool	Name	Description
	Display Business Rule Designer	Opens a window in the lower portion of the Page Flow Designer that enables you to configure relationships between Input and Output attributes.
	Show Page Flow Code	Opens a window in the lower portion of the Page Flow Designer that displays the BPEL4WS code generated by the Page Flow Designer.
	Synchronize Graphical Model and Page Flow Code	Synchronizes the Page Flow on the canvas with the underlying code generation.
	Validate Page Flow Model	Checks for and reports on problems in the Page Flow code.
	Show Property Sheet	Enables you to set alert, logging, message event, and timer event properties for the Page Flow.
	Print	Prints the Page Flow screen image. You can control the scale of the printed image.
	Do Auto Layout	Automatically spaces and organizes components on the design canvas. For more information, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.

Tool	Name	Description
▤▾	Align or Distribute	Aligns two or more selected objects.
100% ▾	Zoom	Increases or decreases the Page Flow image on the screen.

6.1.4 Generating Page Flow Reports

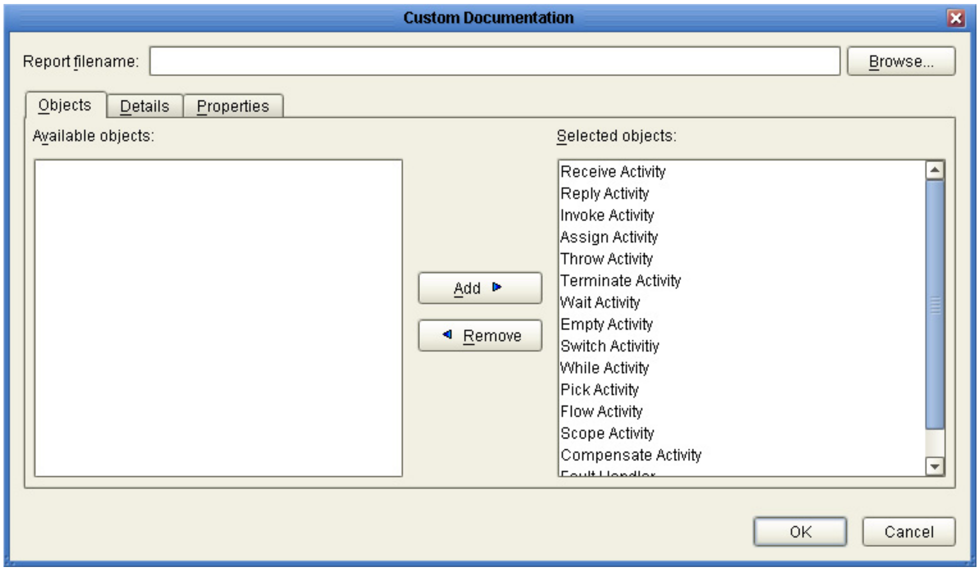
After creating a Page Flow, you can generate a report about the Page Flow.

To generate a Page Flow report

1. Right-click the Page Flow.
2. On the shortcut menu, click **Generate Report**.

   The **Custom Documentation** dialog box appears.

Figure 51 Custom Documentation Dialog Box



Receive Activity
Reply Activity
Invoke Activity
Assign Activity
Throw Activity
Terminate Activity
Wait Activity
Empty Activity
Switch Activitiy
While Activity
Pick Activity
Flow Activity
Scope Activity
Compensate Activity
Fault Handler

3. Enter a file name for the report.
4. Specify which objects you want to appear in the report, and the level of detail for the objects.
5. Click **OK**.

   The report appears in your default web browser.

6.2 Adding Elements to a Page Flow

The Page Flow Designer provides elements that enable you to customize a Page Flow. You drag pages from the Project Explorer and and drop them onto the design canvas. In addition to pages, Page Flows can contain various types of elements, including basic elements, branching elements, and intermediate events.

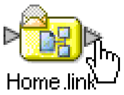
6.2.1 Linking Page Flow Components

In a Page Flow, you create links between components. You can change the style of the links.

To link Page Flow components

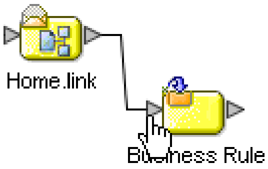
1. Move your cursor over the right connector of a Page Flow component until the cursor changes from the arrow pointer to the hand pointer.

Figure 52 Starting a Link



2. Press the mouse button.
3. Drag the cursor to the left connector of the second Page Flow component.
4. Release the mouse button.

Figure 53 Finishing a Link



To change the link style

1. Right-click the Page Flow
2. On the shortcut menu, click **Toggle Link Style**.

6.2.2 Basic Elements

You can add several types of basic elements to a Page Flow. In addition, the Start Node and End Node elements are automatically added to a Page Flow.

Table 19 Basic Elements

Icon	Name	Description
○	Start Node	The **Start Node** element indicates the start of the Page Flow. The Start Node is added automatically when you create a Page Flow. A Start Node can connect to a **Page Link** or an **Event Based Decision** element.
	Link	**Links** define the connectivity of the Page Flow by connecting page and sub-process elements together. The Page Flow Designer ensures that the Page Flow is correctly linked by rejecting invalid links. Links can also accept business rules. A link with a business rule is marked with a blue icon.
○	End Node	The **End Node** element indicates the completed state of a Page Flow. The End Node is added automatically when you create a Page Flow.
	Receive	The **Receive** element can connect to a **Page Link** or can connect to a **Start Node** via a **Page Link**. It is used to indicate the invocation of the Page Flow. The Receive element represents the actual method by which a Page Flow is initiated (for example, a user types a URL into the browser and a servlet initiates the Page Flow).
	Activity	An **Activity** is a step in the Page Flow in which the Page Flow Engine invokes a web service or an eGate Integrator component. Depending on the configuration of the component, a response may or may not be required. One example is a synchronous extraction process from a database to return the credit status of a trading partner.
	Reply	The **Reply** element enables a Page Flow to respond to the external system or user that originally invoked the Page Flow. The original Receive at the beginning of the Page Flow is paired with the Reply at the end. When a message must be sent back to the caller of the Page Flow, the Reply uses information that correlates the message in the calling system. A Reply is the last step in a Page Flow that acts as a web service or sub-process. A Reply correlates the outbound message back to the calling process. For example, it can reply to an external system as a web service.
	Business Rule	The **Business Rule** element sets data values, including task assignments. This element is used when pages have multiple data mappings between the invocation of human tasks or automated systems.
	Compensate	The **Compensate** element is used to invoke compensation on an inner scope that has already completed normally. This construct can be invoked only from within a fault handler or another compensation handler. For more information, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.

Table 19 Basic Elements

Icon	Name	Description
	Empty	The **Empty** element allows data to pass through without changes.
	Wait	The **Wait** element suspends processing for a specified period of time or until a deadline is reached.

6.2.3 Branching Elements

Branching elements enable you to specify the logical flow of information. eVision Studio provides the following types of branching elements: Decision, Event Based Decision, and Flow.

Table 20 Branching Elements

Icon	Name	Description
	Decision	The **Decision** element enables one of several possible paths to execute, based on expression logic. This element is used to create complex expressions that determine the path of the Page Flow. It also contains the expression and connection names. The Decision element enables you to define expressions that are evaluated to determine the routing of the Page Flow. You build expressions using the mapping interface and Page Flow attributes.
	Event Based Decision	The **Event Based Decision** element enables one of several possible paths to execute, based on which link the user selected. For more information, see **"Event Based Decision" on page 128**.
	Flow	The **Flow** element specifies that one or more pages and/or processes are to flow concurrently. **Note:** If you place more than one eVision Studio artifact in the Flow element, then the Page Flow will not function correctly. The term *artifact* refers to a Page Layout or the **isUserInRole** operation.

To add a branching element to the Page Flow Designer canvas

1. On the Page Flow Designer toolbar, click the expansion arrow on the **Branching Activities** icon.
2. Click the branching element that you want to use and drag it to the Page Flow Designer canvas.

6.2.4 Intermediate Events

Intermediate events are elements that can interrupt a Page Flow. Some intermediate events handle exceptions that may occur at runtime or compensate for exceptions.

Table 21 Intermediate Events

Icon	Name	Description
	Compensation Handler	The **Compensation Handler** is used when something in a Page Flow fails and requires a rollback based on upstream activities. On an automatic basis in the Page Flow, upstream steps in the Page Flow are notified that the failure has occurred and certain transactions need to be reversed, sometimes in a sequential order. The Compensation Handler enables you to design the process and circumstances in which the compensation takes place. For more information, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.
	Catch Named Exception	Each automated system (backend system) or web service can publish its possible error codes (for instance, fault 15 is "bad data"). These codes can be mapped to exception handlers. Each exception handler is connected to the scope that surrounds one or more steps in a Page Flow. The components within that scope throw the exceptions when errors occur, and the exception handler automatically initiates the appropriate process to handle the problem. For more information, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.
	Catch All Exceptions	The **Catch All Exceptions** handler is configured to handle all exceptions. For more information, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.
	Message Event	The **Message Event** is similar to a Receive Activity, but it occurs only in the middle of a process. For more information, see **"Event Based Decision" on page 128**.
	Timer Event	The **Timer Event** imposes a time-out condition on pages, groups of pages, or a Page Flow as a whole to ensure that processes complete within a specified time-frame. Conditions also enable the creation of the process that takes place after a time-out condition takes place. For more information, see **"Event Based Decision" on page 128**.
	Throw	The **Throw** handler throws exceptions.
	Terminate Process	The **Terminate Process** handler ends the Page Flow.

To add an intermediate event to the Page Flow Designer canvas

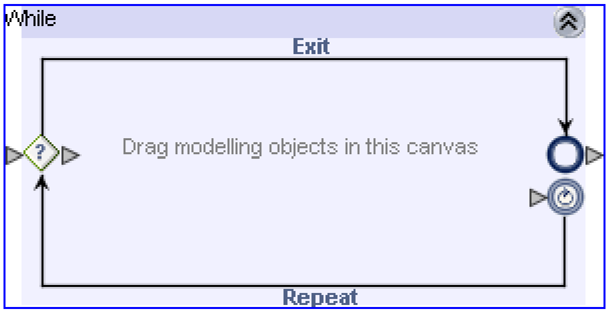
1. On the Page Flow Designer toolbar, click the expansion arrow on the **Intermediate Events** icon.
2. Click the Intermediate event that you want to use and drag it to the Page Flow Designer canvas.

6.2.5 While Element

The **While** element enables you to encapsulate all or part of a Page Flow within a looping process. This element creates and maintains a looping process within a Page Flow. A loop continues a process until an event takes place that signals that the Page Flow is to continue.

After you drag this element onto the design canvas, double click the icon. The element expands.

Figure 54 While Element - Expanded

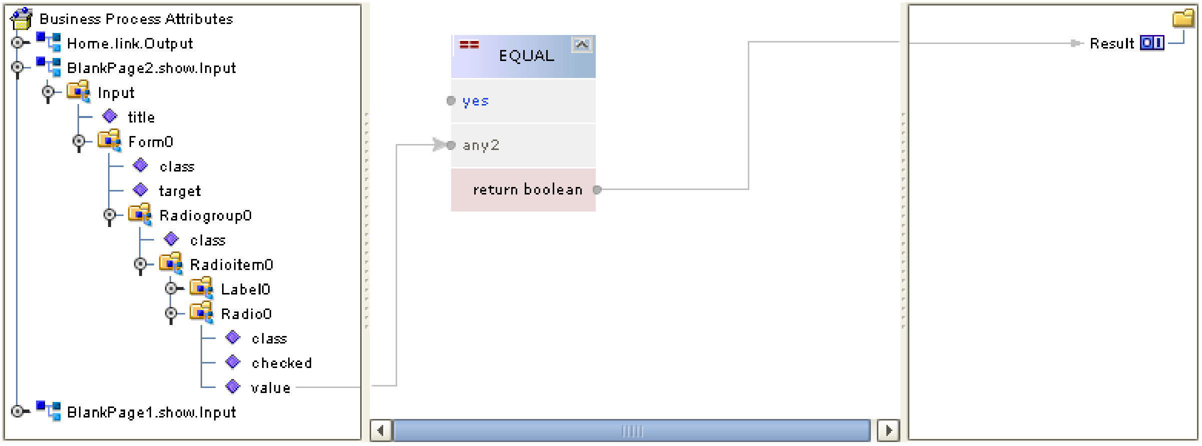


You can now drag objects into the While area by using either of the following approaches:

- Click an element in the Page Flow Designer toolbar and drag the element into the While area.
- Select one or more objects on the Page Flow Designer canvas. Press Control-X to cut the objects. Click inside the While area. Press Control-V to paste the objects them from the clipboard into the While area. Any links that connect the objects are included.

To specify the Boolean expression that controls whether the while loop is executed, click the diamond with a question mark inside it. Then configure the expression in the Business Rule Designer.

Figure 55 Boolean Expression for the While Loop Example

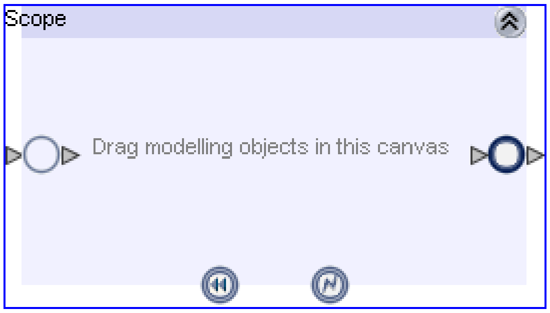


6.2.6 Scope Element

The **Scope** element enables you to assign a compensation handler and fault handlers to a portion of a Page Flow. For example, you can use this element in a Page Flow that updates back-end systems that might need to be compensated.

After you drag this element onto the design canvas, double click the icon. The element expands.

Figure 56 Scope Element - Expanded



You can now drag objects into the Scope area by using either of the following approaches:

- Click an element in the Page Flow Designer toolbar and drag the element into the Scope area.
- Select one or more objects on the Page Flow Designer canvas. Press Control-X to cut the objects. Click inside the Scope area. Press Control-V to paste the objects them from the clipboard into the Scope area. Any links that connect the objects are included.

The lower boundary of the Scope area contains two icons. You can drag a **Compensation Handler** element onto the left icon. You can drag a **Catch Named Exception** element and/or a **Catch All Exceptions** element onto the right icon.

6.2.7 Event Based Decision

The **Event Based Decision** element allows one of several possible paths to execute within a Page Flow, based on an event associated with each path.

Typically, you use this element when Page Layouts contain links to other Page Layouts that will appear in the same browser window. For example, a banking-related web application might contain a home page, a credit page, and a mortgage page, all of which contain links to each other.

As long as the user clicks a link on each page, the **Event Based Decision** element displays the appropriate Page Layout. When the user clicks another type of component, such as a **Submit** button, control passes out of the **Event Based Decision** element. Thus, this element enable you to support a nonlinear flow.

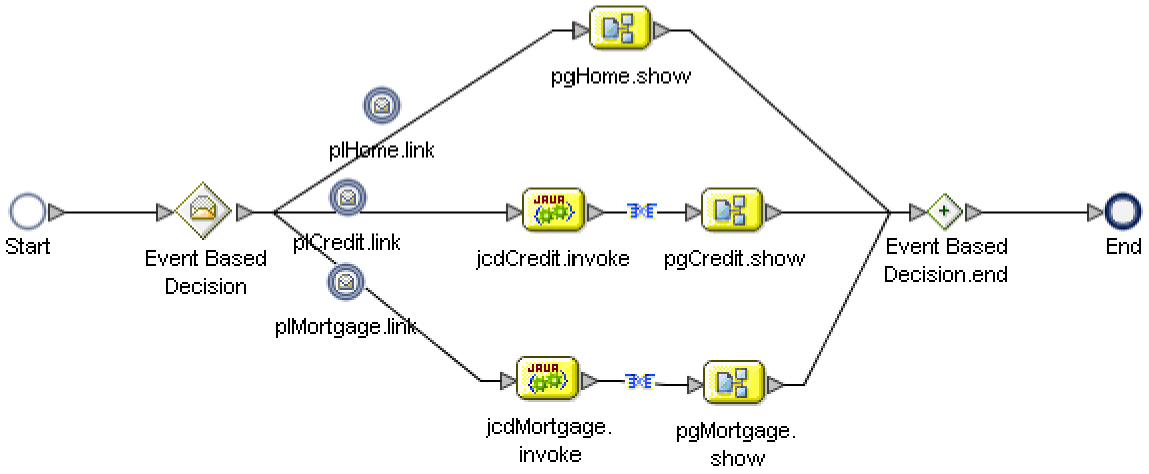
There are two categories of events:

- The **Message Event** represents an incoming message.
- The **Timer Event** represents a duration period or a deadline.

At least one path must be associated with a **Message Event**.

Figure 57 shows an **Event Based Decision** element that has three paths. Each path is associated with a **Message Event**. Each **Message Event** represents the **link** operation of a Page Link.

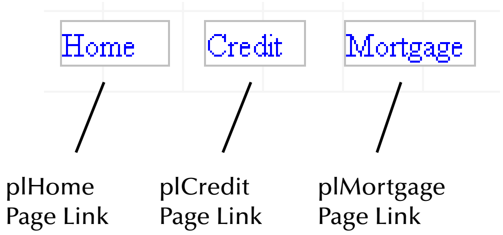
Figure 57 Event Based Decision Example



The **plHome** Page Link is designated as the home page. Therefore, control passes to this Page Link first.

Figure 58 shows the Page Links in the **pgHome** Page Layout. Clicking the **plHome** Page Link displays the same page. Clicking the **plCredit** Page Link displays the credit page (the **pgCredit** Page Layout). Clicking the **plMortgage** Page Link displays the mortgage page (the **pgMortgage** Page Layout).

Figure 58 Page Links in Home Page Example



In the example, each Page Layout contains a **Logout** button. When a user clicks this button, control passes out of the **Event Based Decision** element and the Page Flow exits.

An **Event Based Decision** element cannot point directly to another Page Flow. However, you can call another Page Flow from an activity within one of the branches.

A Page Flow can contain more than one **Event Based Decision** element. For information about how the element maintains conversational state in this situation, see **Appendix B “Conversational State”**.

Creating an Event Based Decision

This section describes the basics of creating an Event Based Decision.

To create an Event Based Decision

1 Drag the **Event Based Decision** element from the Page Flow Designer toolbar onto the design canvas.

Two icons appear: **Event Based Decision** and **Event Based Decision.end**.

2 If you want to rename the icons, click the name of the first icon and enter a new name.

3 Create the two or more paths that can be executed.

4 Connect the **Event Based Decision** icon to the first element of each path. By default, a **Message Event** appears on the link.

5 Connect the last element of each path to the **Event Based Decision.end** icon.

6 Drag the **link** operation of a Page Link onto the links that have a **Message Event**. The name of the event changes based on the name of the Page Link.

7 To change a **Message Event** event to a **Timer Event**, drag the **Timer Event** from the Page Flow Designer toolbar and place it on top of the **Message Event**. You must then configure the **Timer Event**, as described in **“Configuring a Timer Event” on page 130**.

Configuring a Timer Event

When you place a **Timer Event** on a link, you must configure the duration or deadline of the event. The duration or deadline can be static (that is, set at design time) or dynamic (that is, set at runtime).

To configure a Timer Event

1. Select the **Timer Event**.
2. On the Page Flow Designer toolbar, click the **Show Property Sheet** icon.

   The property sheet appears in the right portion of the design canvas.
3. Click the **Timeout** property and then click the ellipsis (**...**).

   The **Timeout** dialog box appears.
4. Choose the appropriate option from the **Alarm Type** drop-down list.

Table 22 Alarm Type Options

Option	Description
Static Duration	The event will take place after a specified amount of time.
Static Deadline	The event will take place at a specified date and time.
Dynamic Duration	The event will take place after a specified amount of time, which is determined at runtime.
Dynamic Deadline	The event will take place at a specified date and time, which is determined at runtime.

   The fields in the **Timeout** dialog box change depending on which option is selected.

   A. If you chose the **Static Duration** option, specify the duration.

   B. If you chose the **Static Deadline** option, specify the date and time.

   C. If you chose the **Dynamic Duration** option, select the attribute that has the value for the duration.

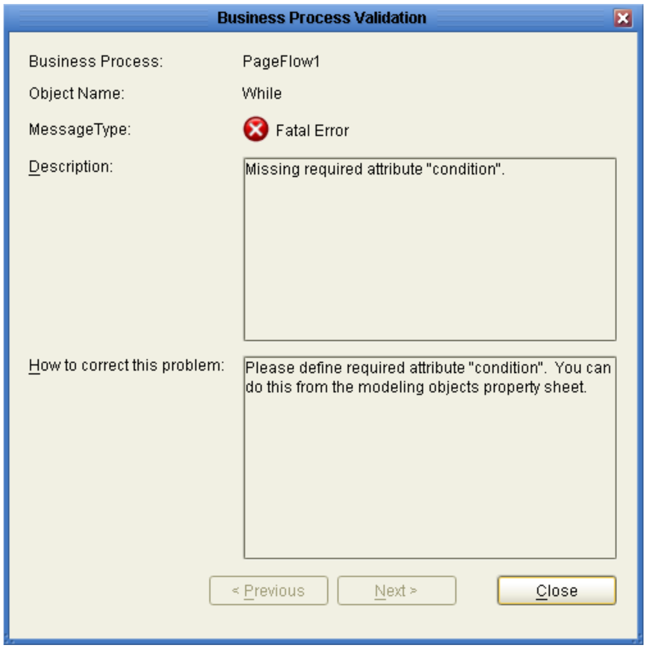
   D. If you chose the **Dynamic Deadline** option, select the attribute that has the value for the deadline.
5. Click **OK**.
6. On the Page Flow Designer toolbar, click the **Hide Property Sheet** icon.

6.2.8 Validating a Page Flow

You can check to see whether a Page Flow has any problems, such as pages that are not connected or an incorrect number of output links from a page.

Figure 59 shows an example of the error that appears when the Boolean condition is not configured for a **While** element.

Figure 59 Business Process Validation Dialog Box



Business Process: PageFlow1

Object Name: While

MessageType: Fatal Error

Description: Missing required attribute "condition".

How to correct this problem: Please define required attribute "condition". You can do this from the modeling objects property sheet.

To validate a Page Flow

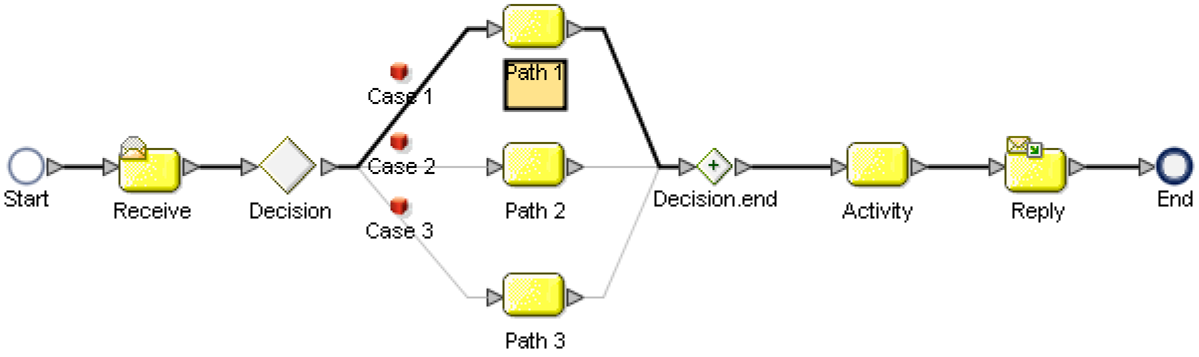
- On the Page Flow Designer **toolbar, click Validate Page Flow Model**.

  The **Business Process Validation** dialog box displays information about any errors or warnings.

6.2.9 Highlighting Upstream and Downstream Paths

If you create a Page Flow with many paths, you can select a single element to see the connected elements upstream and downstream.

Figure 60 Upstream and Downstream Paths



6.3 Configuring Page Flow Designer Elements

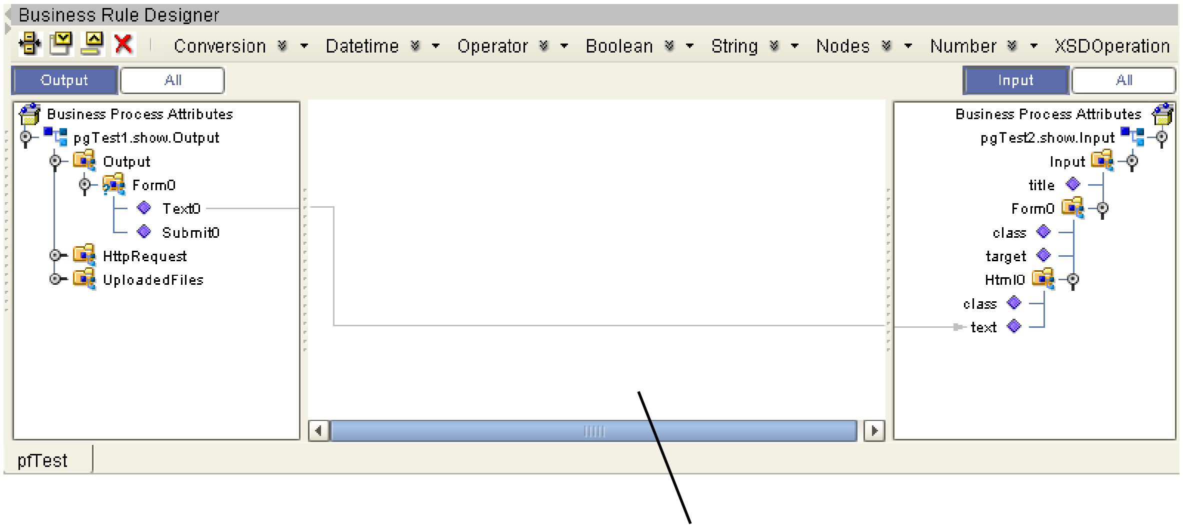
Some elements in the Page Flow Designer have configurable options. This section describes the elements and explains how to configure the options.

6.3.1 Using the Business Rule Designer

The Business Rule Designer enables you to configure relationships between input and output attributes. Some attributes are automatically configured for each sub-process when you drag and drop a component on the Page Flow Designer.

The area where you map attributes in the Business Rule Designer is called the *Mapper*.

Figure 61 Mapper in Business Rule Designer



Mapper

To display the Business Rule Designer, do either of the following:

- On the Page Flow Designer toolbar, click the **Display Business Rule Designer** icon
- Double click an inline Business Rule or an Assign element

6.3.2 Adding an Inline Business Rule

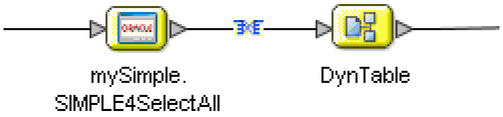
You can add a Business Rule to a link that connects two Page Flow components.

To add an inline Business Rule

1. In the Page Flow, right-click a link between two Page Flow components.
2. On the shortcut menu, click **Add Business Rule**.

   The Business Rule icon appears on the link.

Figure 62 Business Rule Icon

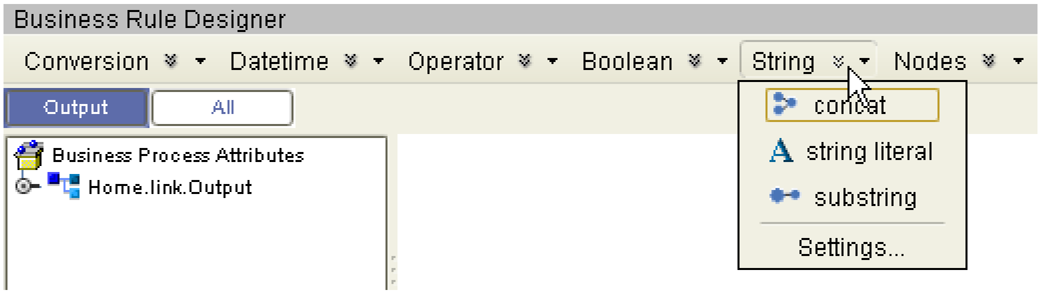


6.3.3 Using the Method Palette

Use the Method Palette in the Business Rule Designer to configure data passed between input and output nodes. You can drag and drop a method from the Method Palette to the Business Rule Designer and then configure the method.

To add methods to or remove methods from the Method Palette, click the vertical chevrons on the Business Rule Designer toolbar and then click **Settings**.

Figure 63 Business Rule Designer Toolbar



Appendix A describes each method in the Method Palette.

6.4 Page Flow Properties

Each Page Flow has a set of properties that you can edit. eVision Studio uses the properties to automatically create the appropriate Page Flow attributes and input/output structures, for use in the Business Rule Designer.

To edit Page Flow properties

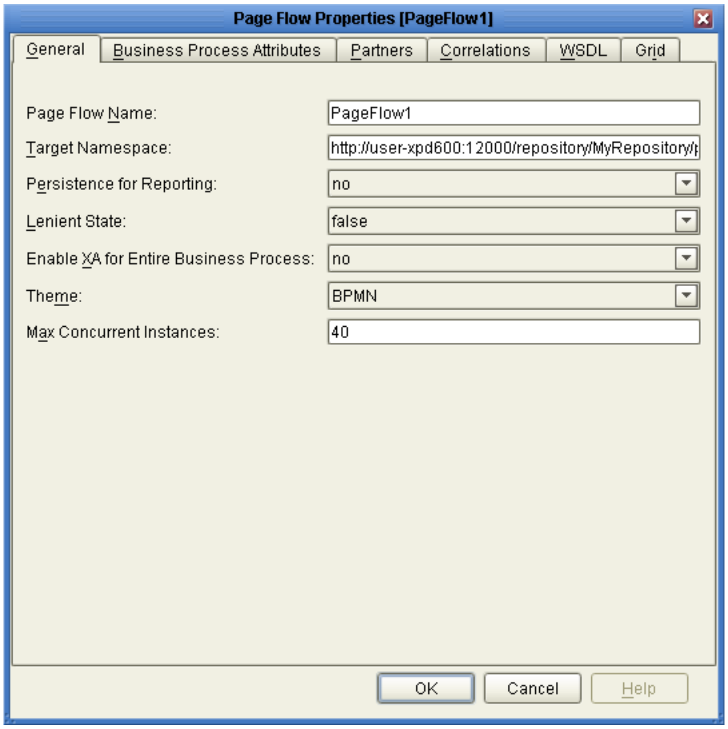
1. In the Project Explorer, right-click the Page Flow.
2. On the shortcut menu, click **Properties**.

   The **Page Flow Properties** dialog box appears.
3. Select one or more tabs and edit properties.
4. Click **OK**.

6.4.1 General Tab

The **General** tab enables you to edit general Page Flow properties.

Figure 64 Page Flow Properties: General Tab



Page Flow Name: PageFlow1
Target Namespace: http://user-xpd600:12000/repository/MyRepository/
Persistence for Reporting: no
Lenient State: false
Enable XA for Entire Business Process: no
Theme: BPMN
Max Concurrent Instances: 40

Table 23 describes the properties.

Table 23 General Tab Properties

Property	Description
Page Flow Name	The name of the Page Flow.
Target Namespace	The address of the Page Flow.
Persistence for Reporting	For information about this property, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.
Lenient State	This property applies only to Page Flows that are imported from release 5.0.0, or third-party BPEL code that does not check for the presence of data before doing a copy. If you incorrectly receive the error message **No pointer for xpath: xpath**, then set the value for this property to **true**.
Enable XA for Entire Business Process	For information about this property, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.
Theme	The look and feel of the Page Flow Designer. The default theme is **BPMN**.
Max Concurrent Instances	For information about this property, see the *Sun SeeBeyond eInsight Business Process Manager User's Guide*.

6.4.2 Page Flow Attributes Tab

Page Flow Attributes are data values used by a Page Flow. These attributes make it possible to share data between activities in a Page Flow, as well as move data to and from the components that implement those activities. Complex structures such as Object Type Definitions (OTDs) and Collaborations are represented automatically in the Project Explorer and are available for use in a Page Flow.

Some examples of Page Flow Attributes are:

- Customer names
- Addresses
- Order quantities
- Item descriptions

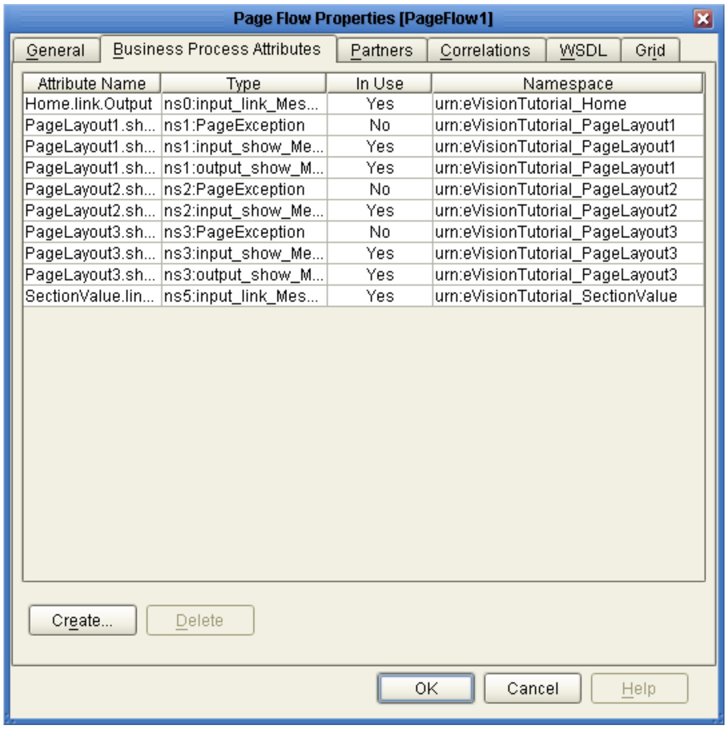
Page Flow Attributes are used to pass values between the Page Flow and external sources. You can assign Page Flow Attributes to specific activities. For example, the customer name is passed to an order process from the originating source. The customer name may be used by several of the activities in the Page Flow and is included in the Page Flow output.

eVision Studio can pass all or part of a complex structure, or it can even assemble a composite input to a component or web service from multiple Page Flow attributes.

To create a Page Flow attribute

1. In the Project Explorer, right-click the Page Flow.
2. On the shortcut menu, click **Properties**.
3. Select the **Page Flow Attributes** tab.

Figure 65 Page Flow Properties: Attributes Tab

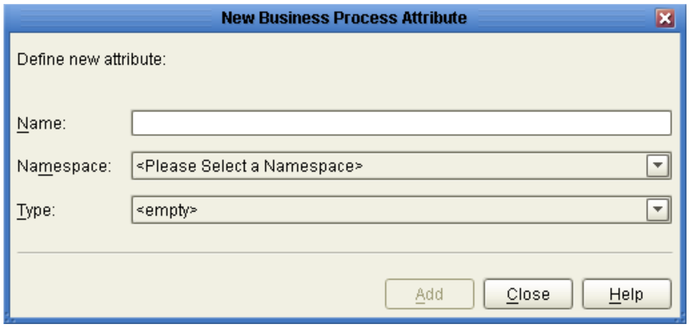


Attribute Name	Type	In Use	Namespace
Home.link.Output	ns0:input_link_Mes...	Yes	urn:eVisionTutorial_Home
PageLayout1.sh...	ns1:PageException	No	urn:eVisionTutorial_PageLayout1
PageLayout1.sh...	ns1:input_show_Me...	Yes	urn:eVisionTutorial_PageLayout1
PageLayout1.sh...	ns1:output_show_M...	Yes	urn:eVisionTutorial_PageLayout1
PageLayout2.sh...	ns2:PageException	No	urn:eVisionTutorial_PageLayout2
PageLayout2.sh...	ns2:input_show_Me...	Yes	urn:eVisionTutorial_PageLayout2
PageLayout3.sh...	ns3:PageException	No	urn:eVisionTutorial_PageLayout3
PageLayout3.sh...	ns3:input_show_Me...	Yes	urn:eVisionTutorial_PageLayout3
PageLayout3.sh...	ns3:output_show_M...	Yes	urn:eVisionTutorial_PageLayout3
SectionValue.lin...	ns5:input_link_Mes...	Yes	urn:eVisionTutorial_SectionValue

4 Click **Create**.

The **New Page Flow Attribute** dialog box appears.

Figure 66 New Page Flow Attribute Dialog Box



Define new attribute:

Name:

Namespace: <Please Select a Namespace>

Type: <empty>

5 In the **Name** field, enter a name for the attribute.

6 In the **Namespace** field, select an existing namespace.

7 In the **Type** field, select one of the available types.

8 Click **Add**, and then click **Close**.

9 Click **OK**.

To edit a Page Flow attribute

1 In the **Page Flow Properties** dialog box, select the **Page Flow Attributes** tab.

2. Select the attribute.
3. To rename the attribute, double click the attribute text and type a new name. Some attributes cannot be renamed.
4. Click **OK**.

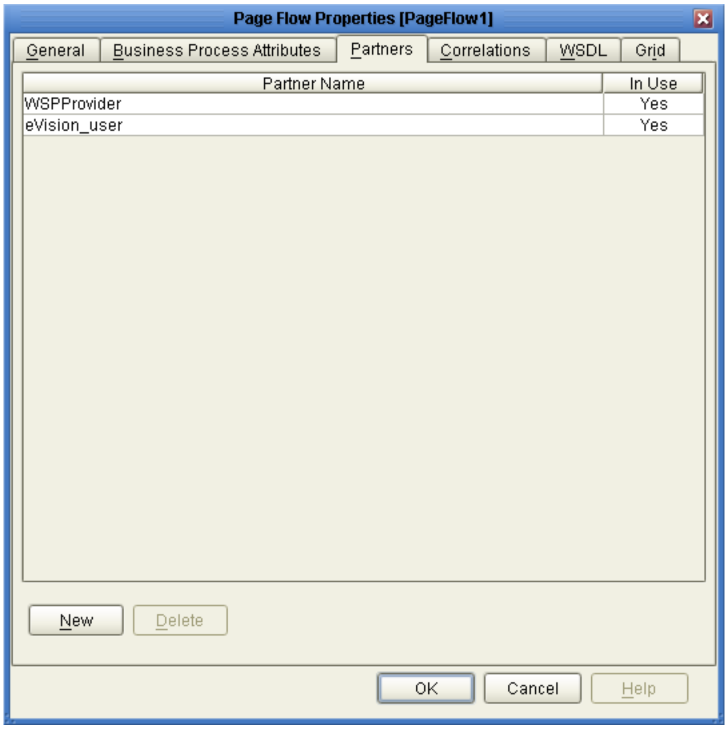
To delete a Page Flow attribute

1. In the **Page Flow Properties** dialog box, select the **Page Flow Attributes** tab.
2. Select the attribute and click **Delete**. Some attributes cannot be deleted.
3. Click **OK**.

6.4.3 Partners Tab

A partner is an abstracted identification for an external system that will appear in the binding box in the Connectivity Map Editor. Multiple activities can use the same external system; therefore, multiple Activities may have the same Partner. By default, eVision Studio assigns this identification to speed up and automate the model development.

Figure 67 Page Flow Properties: Partners Tab



Partner Name	In Use
WSPProvider	Yes
eVision_user	Yes

When creating a Page Flow that will be used as a sub-process, you must create a partner and associate it with the Receive element or Receive/Reply pair. See **"Invoking Another Page Flow or a Business Process" on page 166**.

The **Partners** tab enables you to delete a partner that is not in use.

6.4.4 Correlations Tab

eVision Studio enables you to match existing Page Flow instances to messages that are arriving into a Page Flow.

To correlate messages

1. Create the correlation keys.
2. Create the correlation sets.
3. Bind correlation sets to Page Flow elements.

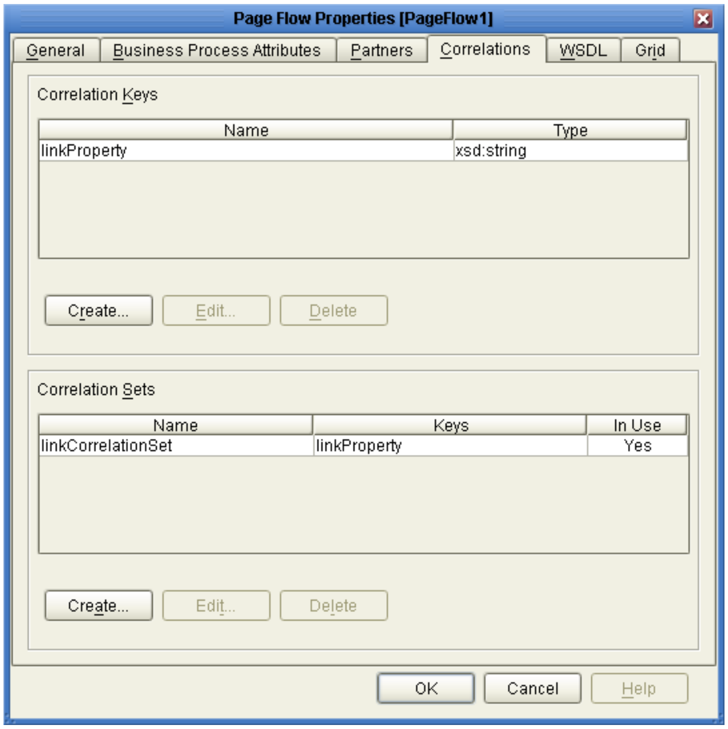
Creating Correlation Keys

A *correlation key* is a value that you can assign to a Page Flow, such as a Purchase Order number. The correlation key provides a way to associate and route information about specific Page Flow instances. For asynchronous message exchange between components, you must implement correlation of the instance identification. An example of when you use asynchronous message exchanges is when you create a Receive activity in the middle of a Page Flow.

To create a correlation key

1. In the Project Explorer, right-click the Page Flow.
2. On the shortcut menu, click **Properties**.
3. Select the **Correlations** tab.

Figure 68 Page Flow Properties: Correlations Tab



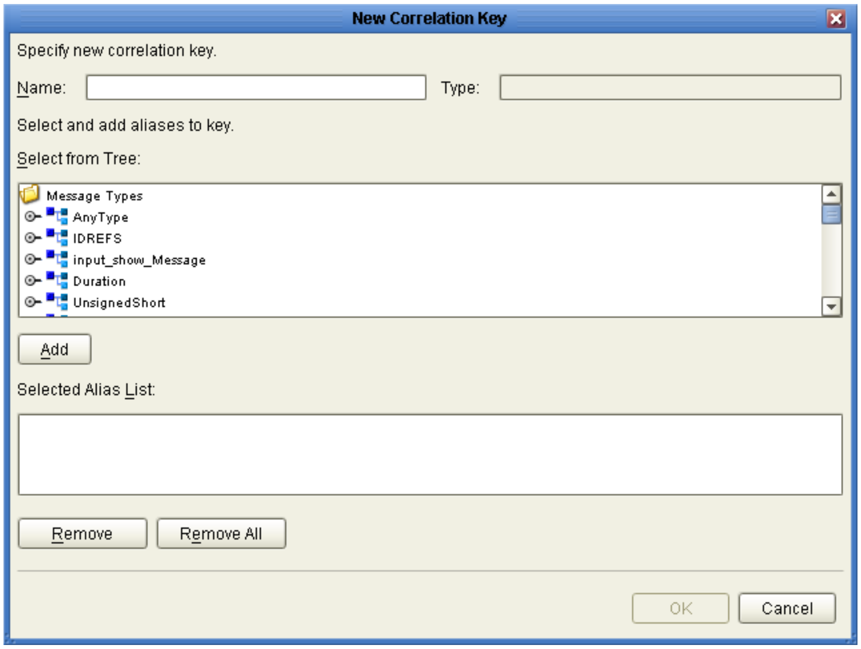
Name	Type
linkProperty	xsd:string

Name	Keys	In Use
linkCorrelationSet	linkProperty	Yes

4 In the **Correlation Keys** section, click **Create**.

The **New Correlation Key** dialog box appears.

Figure 69 New Correlation Key Dialog Box



New Correlation Key

Specify new correlation key.

Name:

Type:

Select and add aliases to key.

Select from Tree:

5 Enter a **Name** (alias) for the correlation key.

6 Select a **Message Type** from the list to alias. Select one or more correlation keys that comprise a unique identifier for a step in a Page Flow.

7 To save the new alias to the **Selected Alias List** box, click **Add**.

8 Click **OK**.

Adding Correlation Sets

Correlation sets are groups of properties shared by all messages in the group. A correlation set matches messages and conversations with a Page Flow instance. For example, you might want to assign a Purchase Order number and an invoice number to a transaction, so that all information about the purchase and payment are associated.

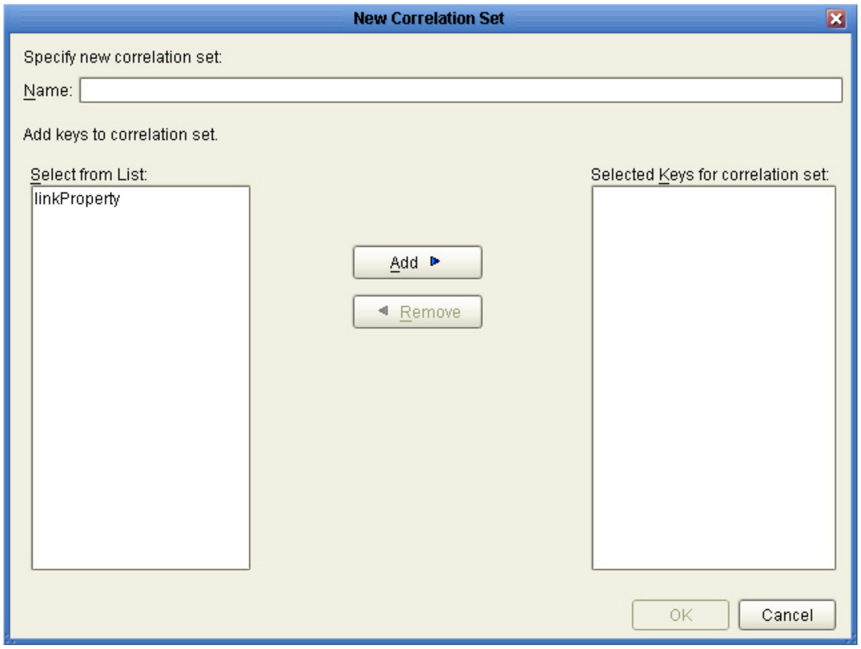
To add a correlation set

1 Select the **Correlations** tab.

2 In the **Correlation Sets** section, click **Create**.

The **New Correlation Set** dialog box appears.

Figure 70 New Correlation Set Dialog Box



New Correlation Set

Specify new correlation set:

Name:

Add keys to correlation set.

Select from List:

linkProperty

Selected Keys for correlation set:

3 In the **Name** field, enter a name for the correlation set.

4 To add to the correlation set, select correlation keys from the list.

5 To move your selections to the correlation set, click the arrow button.

6 Click **OK**.

Binding Correlation Sets to Page Flow Elements

When you use one or more correlation sets in a Page Flow, the values must first be initialized. If you decide to initialize the set within an element, then you must either use both Page Flow attributes or identify which Page Flow attribute to use.

To bind a correlation set to a Page Flow element

1 Select a Page Flow element.

2 On the Page Flow Designer toolbar, click the **Show Property Sheet** icon.

The properties window appears in the right portion of the design canvas.

3 Locate the **Use Correlations** property and click the **no** field.

4 In the **no** field, click the **Command** button (**...**).

The **Use Correlations** dialog box appears.

5 Click **Add**.

The **Assign Correlation Set** dialog box appears.

6 In the left pane, select the correlation set that you want to add to the Page Flow element.

7 Click the arrow button to move it to **Selected Correlation Set(s)** area.

8. Click **OK**.
9. In the **Use Correlations** dialog box, click **OK**.

6.4.5 WSDL Tab

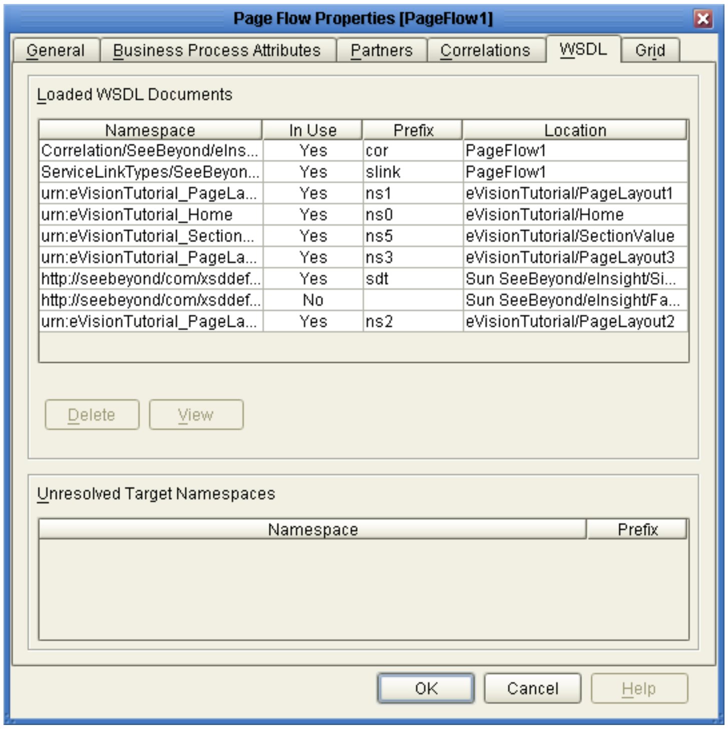
Web Services Description Language (WSDL) is an XML-based language used to describe business services. WSDL provides a way for individuals and other businesses to electronically access those services.

In the Page Flow Designer, WSDL files are used to invoke and operate web services on the Internet and to access and invoke remote applications and databases.

To view a WSDL file

1. In the Project Explorer, right-click the Page Flow.
2. On the shortcut menu, click **Properties**.
3. Select the **WSDL** tab.

Figure 71 Page Flow Properties: WSDL Tab



Loaded WSDL Documents

Namespace	In Use	Prefix	Location
Correlation/SeeBeyond/eIns...	Yes	cor	PageFlow1
ServiceLinkTypes/SeeBeyon...	Yes	slink	PageFlow1
urn:eVisionTutorial_PageLa...	Yes	ns1	eVisionTutorial/PageLayout1
urn:eVisionTutorial_Home	Yes	ns0	eVisionTutorial/Home
urn:eVisionTutorial_Section...	Yes	ns5	eVisionTutorial/SectionValue
urn:eVisionTutorial_PageLa...	Yes	ns3	eVisionTutorial/PageLayout3
http://seebeyond/com/xsddef...	Yes	sdt	Sun SeeBeyond/eInsight/Si...
http://seebeyond/com/xsddef...	No		Sun SeeBeyond/eInsight/Fa...
urn:eVisionTutorial_PageLa...	Yes	ns2	eVisionTutorial/PageLayout2

Unresolved Target Namespaces

Namespace	Prefix

4. Select a WSDL file and click **View**.

   The **WSDL Viewer** dialog box appears.

Note: *You cannot edit the WSDL code.*

6.4.6 Grid Tab

The **Grid** tab enables you to control various properties of the background grid.

To edit grid properties

1. In the Project Explorer, right-click the Page Flow.
2. On the shortcut menu, click **Properties**.
3. Select the **Grid** tab.
4. To change the width (in pixels) of each box within the grid, select a new value from the **Grid Width** spin box.
5. To change the height (in pixels) of each box within the grid, select a new value from the **Grid Height** spin box.
6. To change the color of the gridlines, do the following:
   A. Click the color that appears to the right of the **Grid Color** label. The **Choose a Color** dialog box appears.
   B. Specify the desired color.
   C. Click **OK**.
7. To change the thickness of the gridlines, select the desired setting from the **Grid Thickness** drop-down list.
8. To change the style of the gridlines, select the desired setting from the **Grid Style** drop-down list.
9. To set all of the properties to their default values, click **Reset to Defaults**.
10. Click **OK**.

6.5 Configuring Page Layout Objects

This section describes how to configure the following Page Layout objects in the Page Flow Designer:

- Dynamic tables
- Tree controls and inline frames
- Upload objects
- Image buttons
- SVG objects

6.5.1 Dynamic Tables

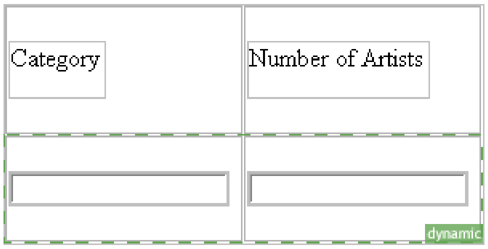
You can configure dynamic tables from the Business Rule Designer.

Mapping Data into a Dynamic Table

After you create a dynamic table in the Page Layout Designer, you map data into the table in the Page Flow Designer.

Figure 72 shows a dynamic table that will be used as an example. The first row is static. It contains two **Label** components that display column headings. The second row is dynamic. It contains two **Text Input Field** objects.

Figure 72 Dynamic Table Example

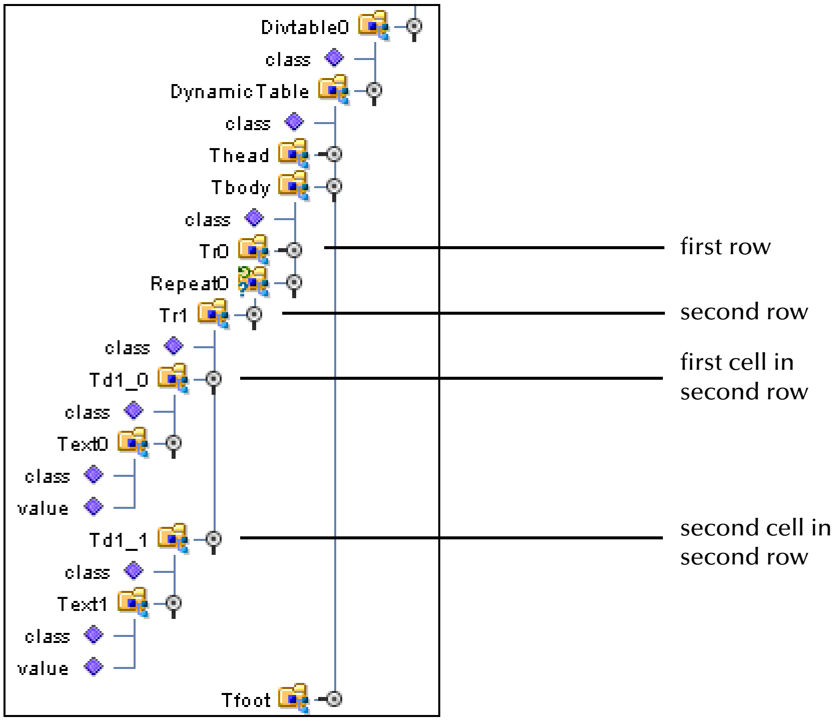


Category	Number of Artists

Figure 73 shows how the dynamic table appears in the Business Rule Designer.

The logical name of this table is **DynamicTable**. The **Tr0** node represents the first row. The **Tr1** node represents the second row. Because the second row is dynamic, it appears within a repeating node (called **Repeat0**). Repeating nodes are highlighted with a green, circular arrow.

Figure 73 Dynamic Table Example in the Business Rule Designer



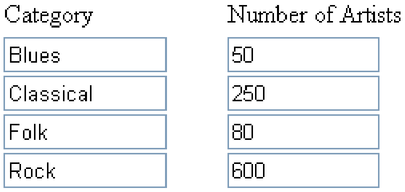
The **Td1_0** node represents the first cell in the second row. The **Text Input Area** object is called **Text0**.

The **Td1_1** node represents the second cell in the second row. The **Text Input Area** object is called **Text1**.

To populate the dynamic row, you map data from an input source (such as an Oracle database) into the appropriate nodes. In the dynamic table example, the data will be mapped into the **value** node of each **Text Input Area** object.

At runtime, the repeating node reads data from the input source until no data is left.

Figure 74 Dynamic Table Example at Runtime



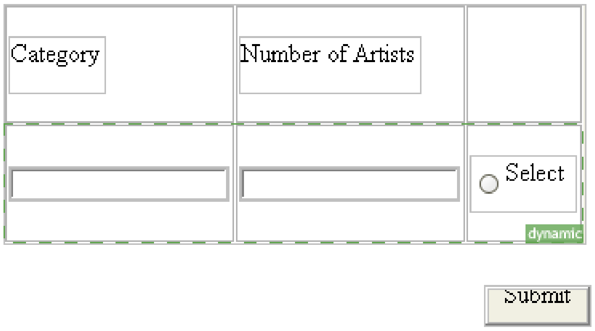
Category	Number of Artists
Blues	50
Classical	250
Folk	80
Rock	600

Using Predicates in a Dynamic Table

You can select one row from a set of dynamically generated rows by creating a predicate.

Figure 75 shows a modified version of the dynamic table in Figure 72. The table now contains three columns. The third cell in the dynamic row contains a **Radio Group** object. A **Submit Button** object has been added below the table.

Figure 75 Dynamic Table with Radio Group Object



At runtime, the user chooses one of the dynamically generated rows by selecting the radio button in the third cell. The user then clicks the Submit button.

Typically, you do something with the row that the user selected. For example, you could display additional information about the user's selection in a new page. To select the row (rather than all of the dynamically generated rows), you can create a *predicate* in the Business Rule Designer.

The predicate represents a condition. If the condition is met, then the mappings underneath the predicate take place. In the dynamic table example, if the value of the radio button is equal to **true**, then the radio button for this particular row is selected.

Once you create a predicate, the Business Rule Designer displays the predicate version of the repeating node immediately below the original version. You then perform the appropriate mapping from the predicate version.

Note: *While a radio button submits only a single value, a checkbox can submit multiple values. The value for the radio is always in the first position.*

The following is an example of a complex predicate:

Assign the output TxtRPSid when it is the same as a selected Radio Button Value.

```
Output/Form0/Divtable2/Table/Tbody/Repeat[ ( ../Repeat[1]/Tr0/Td0_0/
Radiogroup2/value = Tr0/Td0_1/TxtRPSid ) ]
```

You can edit or delete an existing predicate.

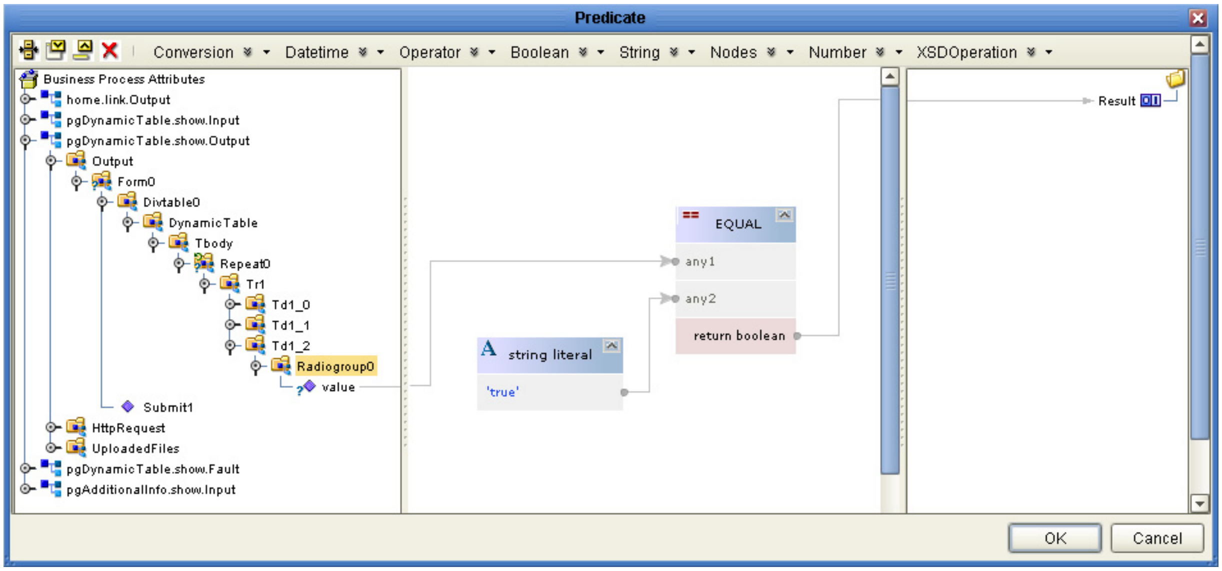
To create a predicate in a dynamic table

1. In the Business Rule Designer, right-click the repeating node that contains the dynamic row.
2. On the shortcut menu, click **New Predicate**.

The **Predicate** window appears.

3 Create the condition. The condition in Figure 76 states that the value of the radio button is equal to **true**.

Figure 76 Predicate Window



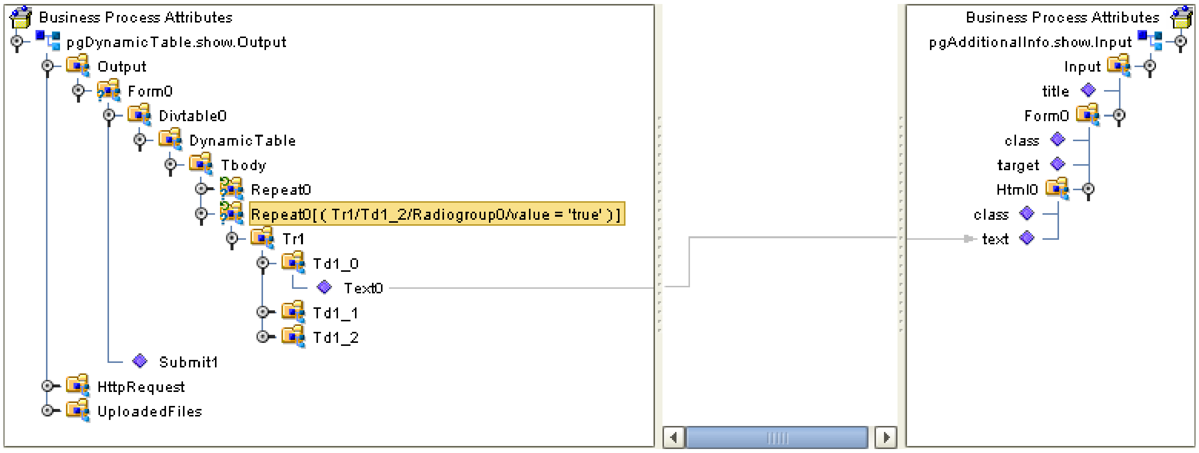
4 Click **OK**.

The Business Rule Designer displays the predicate version of the repeating node immediately below the original version. The syntax of the predicate appears in square brackets.

5 You can now map data from the predicate version of the repeating node.

Figure 77 shows a repeating node that contains a **Radio Group** object. The data is mapped to an **HTML** object in a new page.

Figure 77 Mapping from Predicate Version of Repeating Node (Radio Group)


Business Process Attributes
pgDynamicTable.show.Output
Output
Form0
Divtable0
DynamicTable
Tbody
Repeat0
Repeat0[(Tr1/Td1_2/Radiogroup0/value = 'true')]
Tr1
Td1_0
Text0
Td1_1
Td1_2
Submit1
HttpRequest
UploadedFiles
Business Process Attributes
pgAdditionalInfo.show.Input
Input
title
Form0
class
target
Html0
class
text

For a repeating node that contains a **Checkbox Group** object, the Boolean method **exist** is the default when you perform a straight mapping and the check box has the **checked**, **value**, and **class** properties. The output of the check box is the **value** value, not **true** or **false**.

To edit a predicate

1. In the Business Rule Designer, right-click the predicate.
2. On the shortcut menu, click **Edit Predicate**.
3. Click **Yes**.

   The **Predicate** window appears.
4. Make your changes.
5. Click **OK**.

To delete a predicate

1. In the Business Rule Designer, right-click the predicate.
2. On the shortcut menu, click **Delete Predicate**.
3. Click **Yes**.

   The predicate is deleted.

Setting the Reset Destination Option

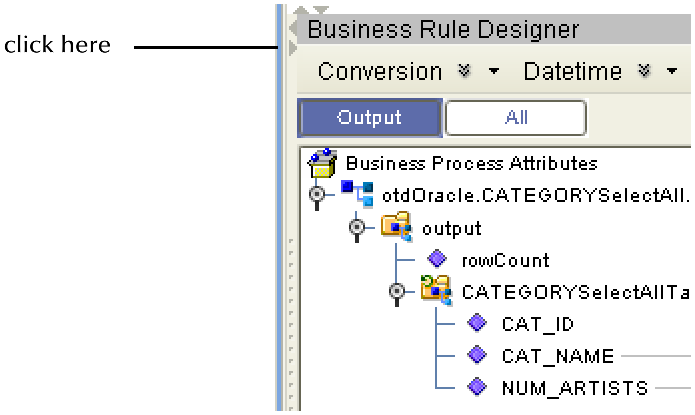
When you use predicates in a dynamic table, you might create a Page Flow in which the user can return to the Page Layout that contains the dynamic table. In this situation, you can “flush” the choices that the user made when the user previously accessed the Page Layout.

The following procedure describes how to set the appropriate option. Setting this option resets the values of the mapped nodes (that is, the destination) in the right side of the Business Rule Designer.

To set the Reset destination option

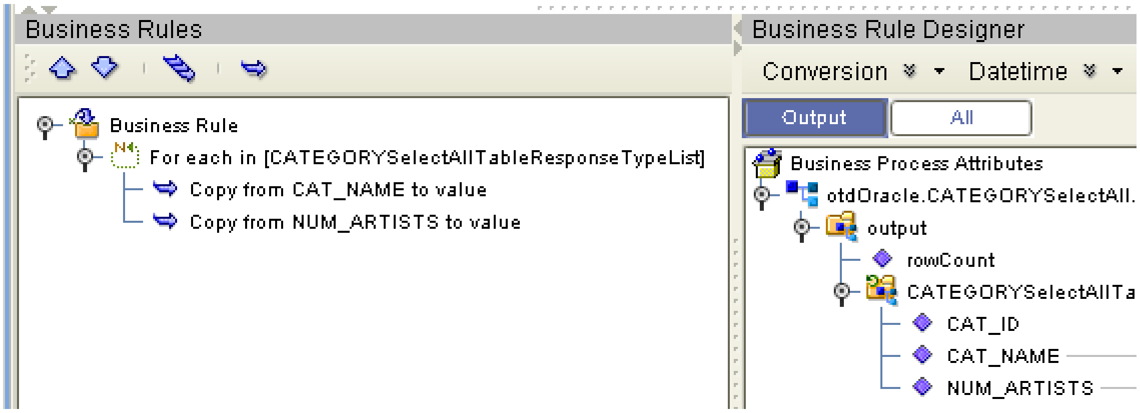
1. In the upper left corner of the Business Rule Designer, click the right-facing triangle.

Figure 78 Displaying the Business Rules Window



The **Business Rules** window appears. This window lists the names of each mapping in the Business Rule Designer.

Figure 79 Business Rules Window


Business Rules
Business Rule
For each in [CATEGORYSelectAllTableResponseTypeList]
Copy from CAT_NAME to value
Copy from NUM_ARTISTS to value
Business Rule Designer
Conversion
Datetime
Output
All
Business Process Attributes
otdOracle.CATEGORYSelectAll.
output
rowCount
CATEGORYSelectAllTa
CAT_ID
CAT_NAME
NUM_ARTISTS

2 Right-click the first business rule and select the **Reset destination** option. When this option is turned on, a check box appears next to the menu item.

3 Click the left-facing triangle.

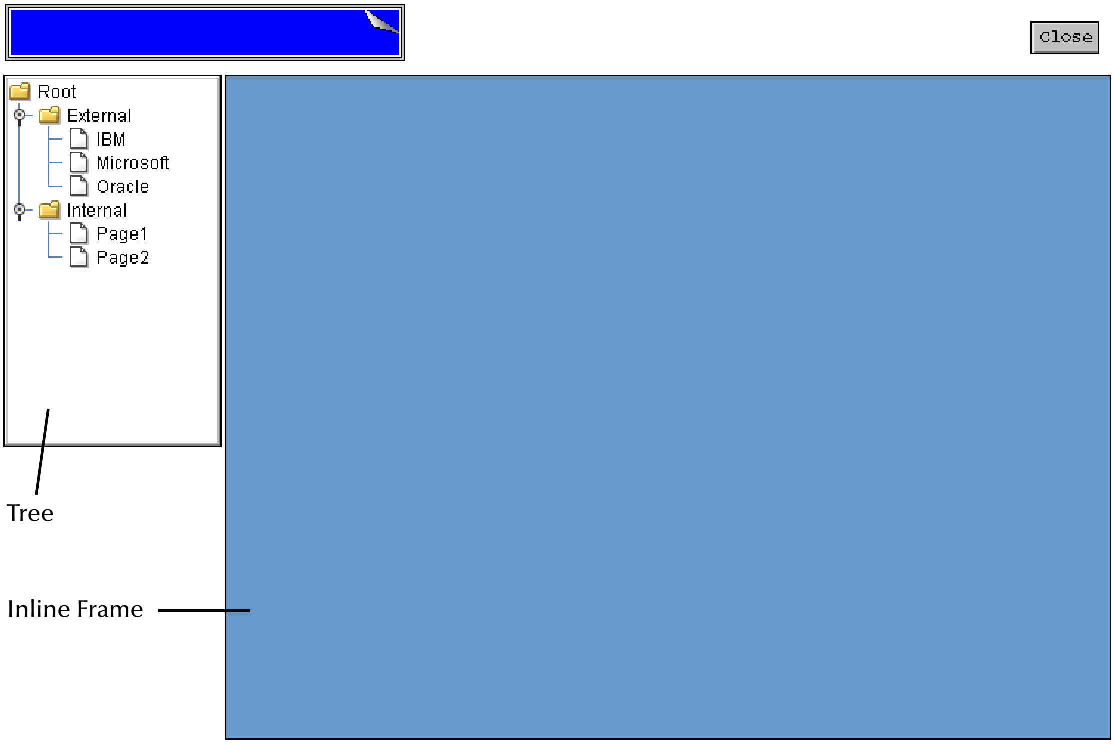
The **Business Rules** window closes.

6.5.2 Tree Controls and Inline Frames

This section describes a strategy for using the **Tree** object and the **Inline Frame** object as the control mechanism for a Page Flow.

In the home page, place a **Tree** object in the left portion and an **Inline Frame** object immediately to the right.

Figure 80 Home Page with Tree and Inline Frame Objects



In this example, the **Tree** object contains a branch node for external links and a branch node for internal links.

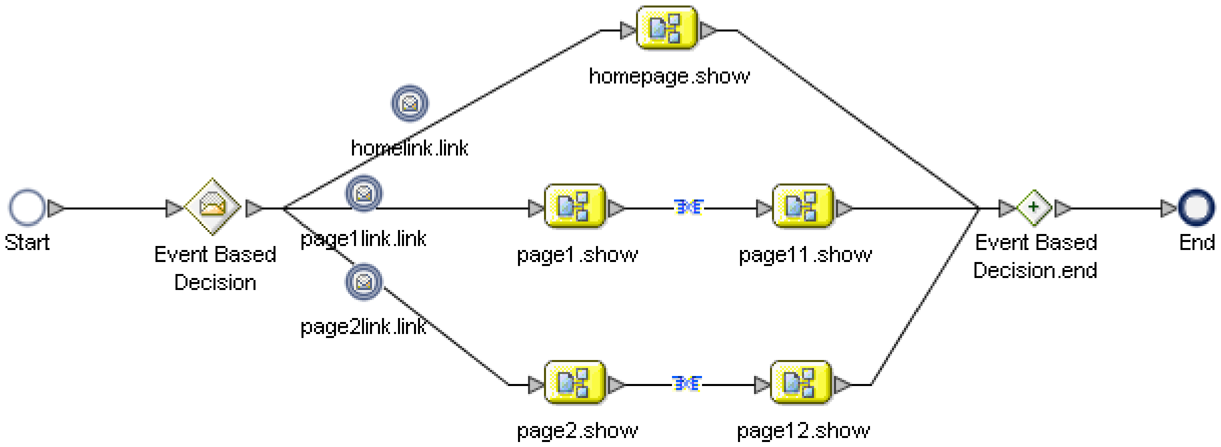
In the external branch node, the **Src** property of each leaf node is set to a company web site (IBM, Microsoft, or Oracle).

In the internal branch node, the **Link** property of each leaf node is configured as follows:

- The **Link** property of the **Page1** leaf node is set to a Page Link called **page1link**.
- The **Link** property of the **Page2** leaf node is set to a Page Link called **page2link**.

Figure 81 shows the corresponding Page Flow.

Figure 81 Page Flow for Tree and Inline Frame Strategy



At runtime, the **Event Based Decision** element passes control to the **homepage** Page Layout.

When the user clicks the IBM, Microsoft, or Oracle leaf node in the tree, the company’s web site is displayed within the **Inline Frame** object.

When the user clicks the **Page1** leaf node in the tree, the **page1** Page Layout is displayed within the **Inline Frame** object. When the user clicks the **Page2** leaf node in the tree, the **page2** Page Layout is displayed within the **Inline Frame** object.

When the user click the **Close** button on the home page, control passes out of the **Event Based Decision** element and the Page Flow ends.

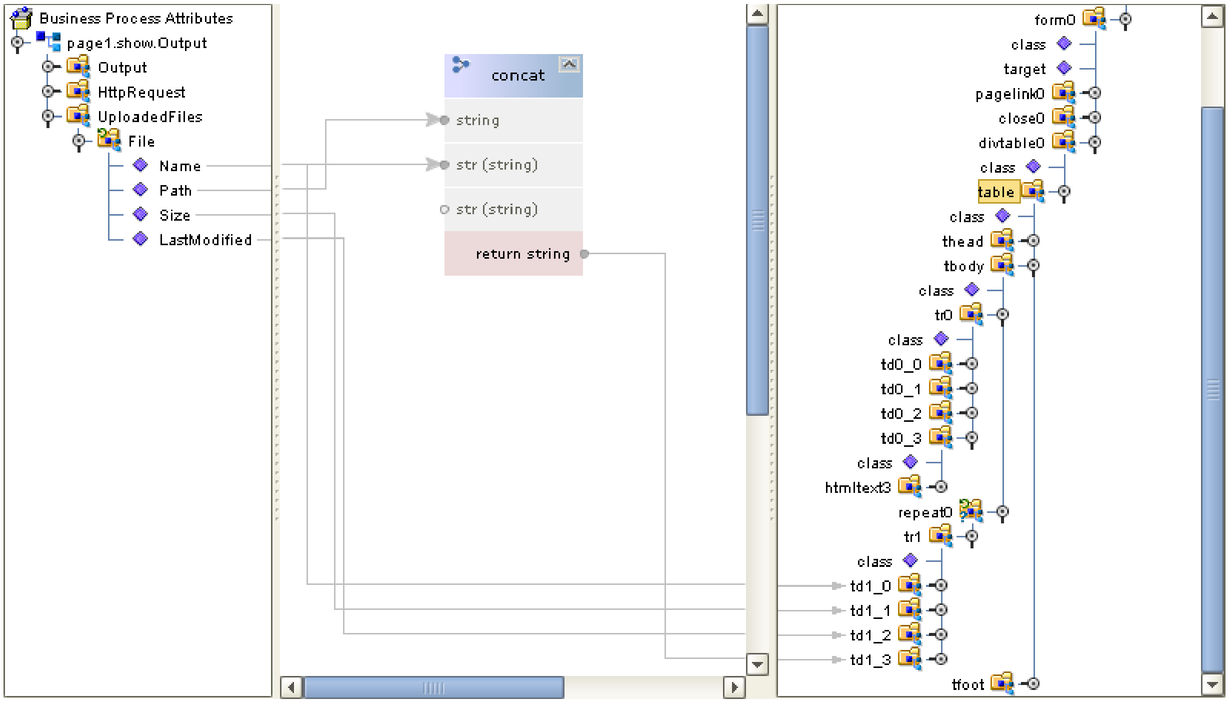
6.5.3 Upload Objects

When you create an **Upload** object in the Page Layout Designer, the Page Flow Designer exposes the following fields for the file that the user uploads: **Name**, **Path**, **Size**, and **LastModified**.

Figure 79 shows an example of mapping the output from the **Upload** object to a table. The table contains two rows and four columns. The output fields are mapped as follows:

- The **Name** field is mapped to the Name column of the table, which contains a **Label** object.
- The **Size** field is mapped to the Size column of the table, which contains a **Label** object.
- The **LastModified** field is mapped to the Modified column of the table, which contains a **Label** object.
- The **Path** and **Name** fields are concatenated, and the result is mapped to the Download column of the table, which contains a **Link** object. This mapping uses the **concat** method.

Figure 82 Mapping the Output from the Upload Object



6.5.4 Image Buttons

When you create an image button in the Page Layout Designer, the Page Flow Designer exposes the mouse coordinates of where the user clicked. The field **lname.x** represents the x coordinate, and the field **lname.y** represents the y coordinate.

Assume that:

- A Page Layout called **page1** contains two image buttons.
- The **lname** property of the first image button is set to **image1**.
- The **lname** property of the second image button is set to **image2**.

If you create a **Decision** element that follows the Page Layout, you can use the mouse coordinate fields to determine which image button the user clicked. You then perform the appropriate business logic.

Figure 83 shows the Page Flow for this example.

Figure 83 Page Flow for Image Button Example

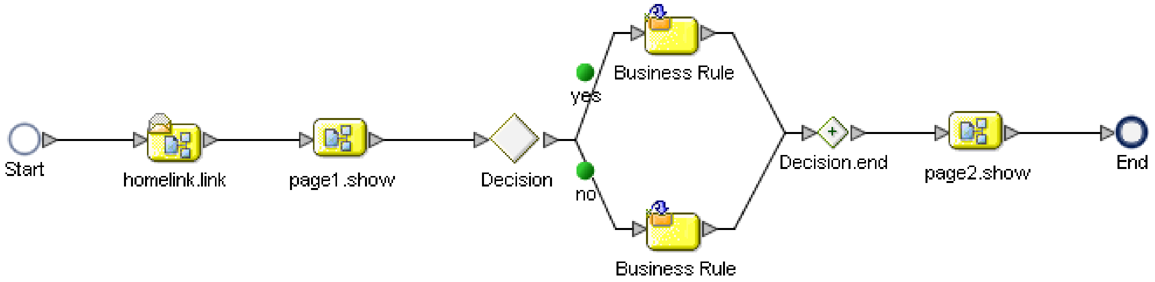
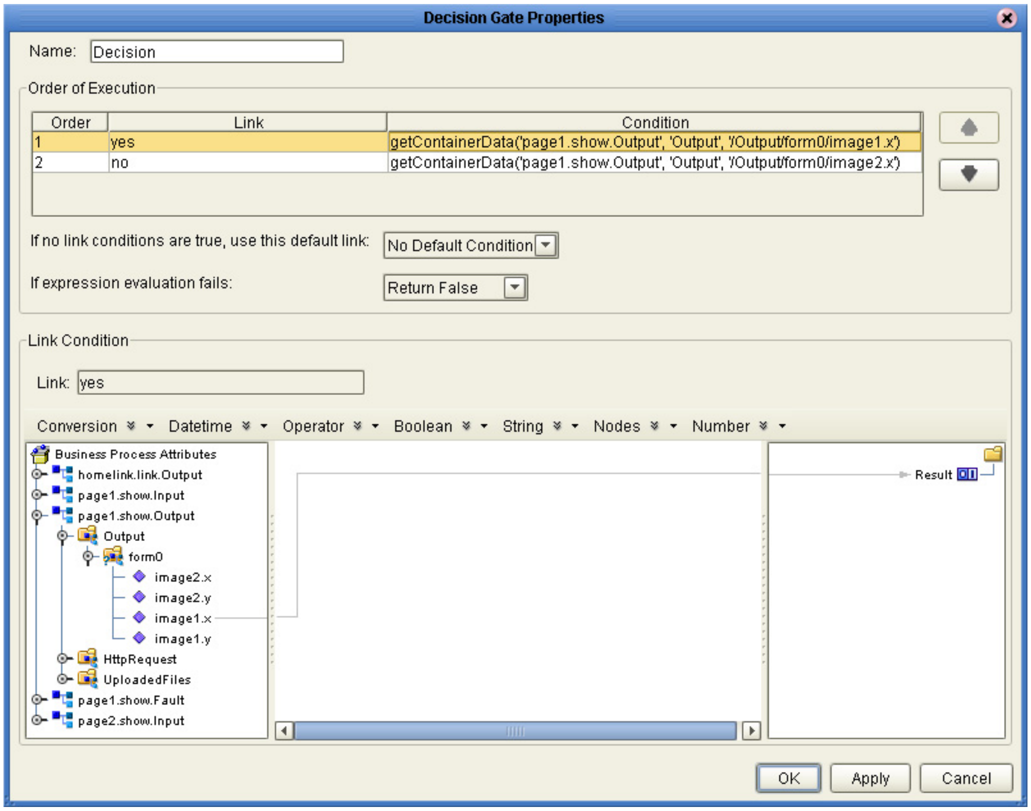


Figure 84 shows the **Decision Gate Properties** window for this example.

Figure 84 Decision Gate Properties Window for Image Button Example



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	yes	getContainerData('page1.show.Output', 'Output', '/Output/form0/image1.x')
2	no	getContainerData('page1.show.Output', 'Output', '/Output/form0/image2.x')

If no link conditions are true, use this default link: No Default Condition

If expression evaluation fails: Return False

Link Condition

Link: yes

The condition for Case 1 checks whether the x coordinate of the **image1** image button contains a value. If the x coordinate contains a value, then the button was clicked and the condition evalues to TRUE.

The condition for Case 2 checks whether the x coordinate of the **image2** image button contains a value. If the x coordinate contains a value, then the button was clicked and the condition evalues to TRUE.

For either condition, you can check the y coordinate instead.

6.5.5 SVG Objects

The Page Flow Designer exposes the structure of any Scalable Vector Graphics (SVG) objects in a Page Layout. You can map data into the elements of an SVG object to dynamically control the display characteristics of that image.

Assume that you want to enable runtime users to control whether a United States map displays states with the highest sales, states with the lowest sales, or both types of states.

Create a Page Layout called **pgCheck**. Add a **Checkbox Group** object. Set the labels to:

- Show States with Highest Sales
- Show States with Lowest Sales
- Show Both Highest and Lowest

Set the values for the labels to:

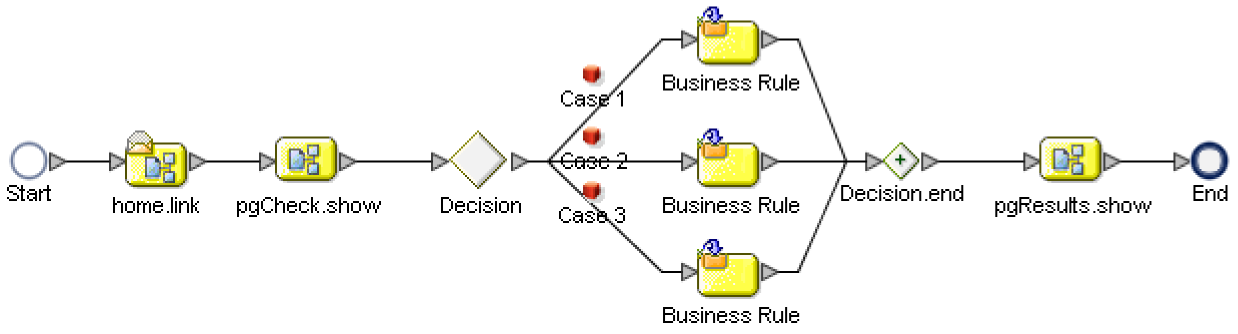
- HSALES
- LSALES
- HLSALES

Add a **Submit Button** object to the right of the checkbox group. Add an **Inline Frame** object below the checkbox group. In the **Properties** tab, click the **Form** property type. Set the **Target** property to the **LName** value of the **Inline Frame** object.

Create another Page Layout called **pgResults**. Import an SVG file that contains a map of the United States. Add an SVG object and specify the file that you imported.

Create a Page Flow as shown in Figure 85.

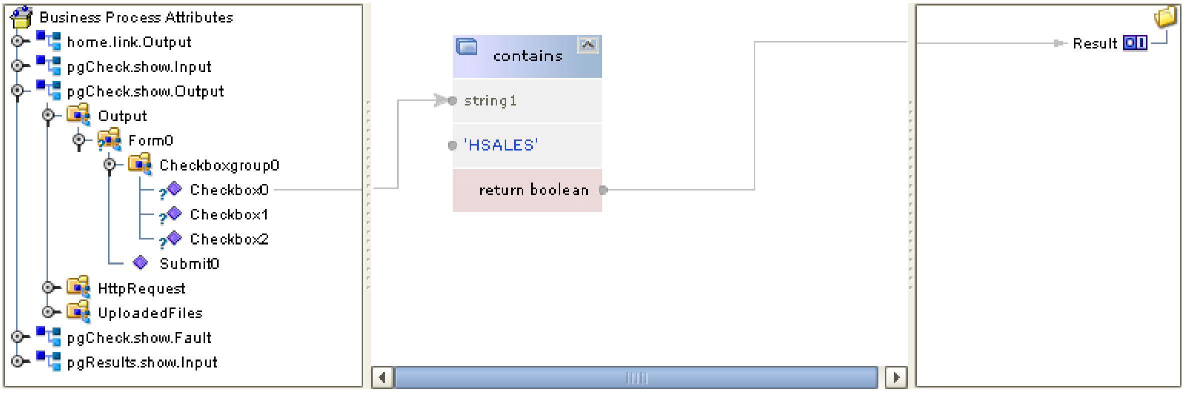
Figure 85 Page Flow for SVG Object Example



The purpose of the **Decision** element is to determine which check box the user selected and to then map a color into the appropriate state in the SVG object.

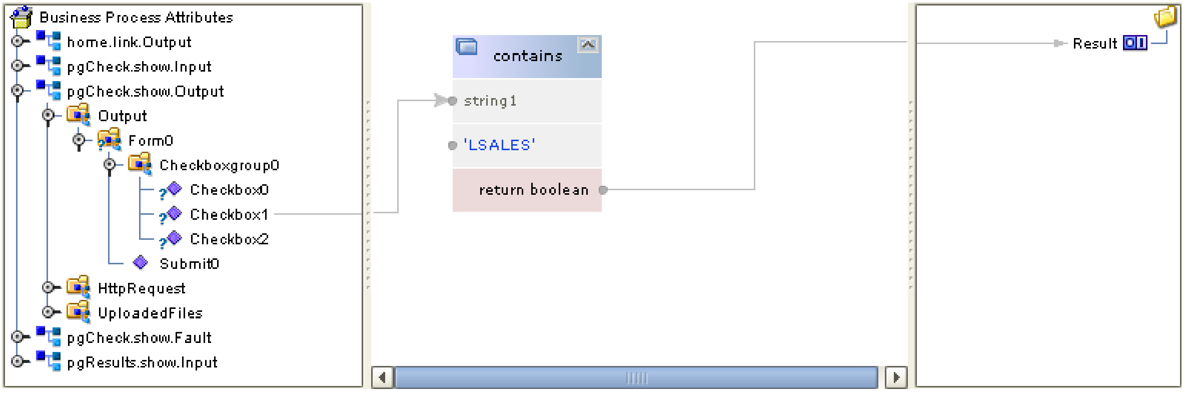
Configure the first case of the **Decision** element. The **HSALES** string literal matches the value of the first check box.

Figure 86 First Case of Decision Element



Configure the second case of the **Decision** element. The **LSALES** string literal matches the value of the first check box.

Figure 87 Second Case of Decision Element

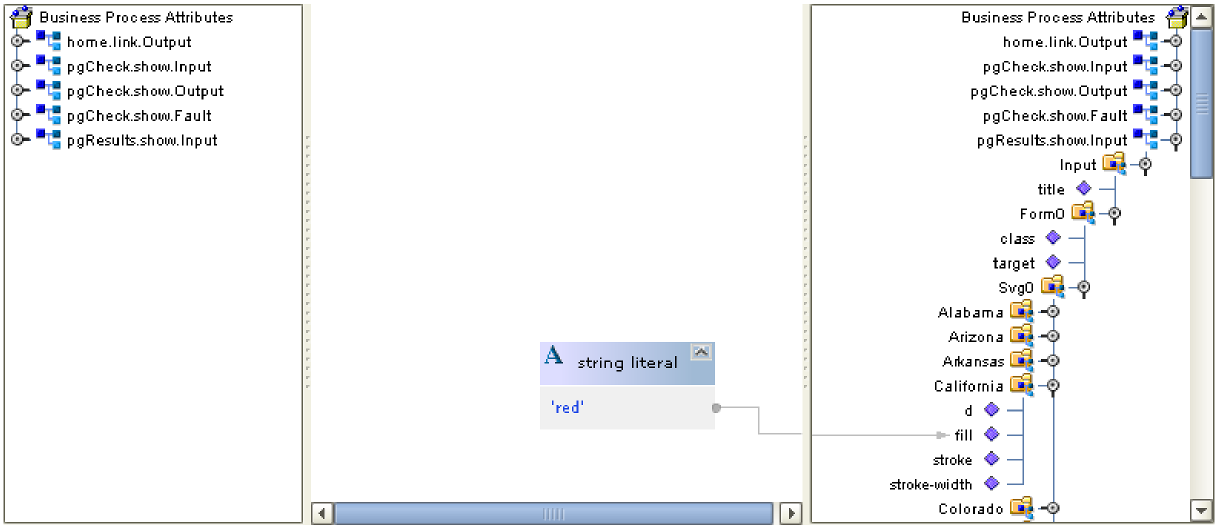


The **Decision Gate Properties** dialog box contains two drop-down lists:

- Set the default link to the third case.
- Indicate that the decision gate should return false if the expression evaluation fails.

Now configure the **Business Rule** elements. The default color of the states is green. In the element for the first case, map the string literal **red** into the color for California.

Figure 88 Business Rule Element for First Case



In the element for the second case, map the string literal **yellow** into the color for Texas.

In the element for the third case, perform both mappings.

6.6 Page Flows in Connectivity Maps

When you create the Connectivity Map for a Project that contains one or more Page Flows, you add each Page Flow in the Project to a Service. You then link each Service to a Web Connector.

6.6.1 Adding Each Page Flow to a Service

For each Page Flow in the Project, create a Service and then drag the Page Flow from the Project Explorer into the Service. The red gears in the Service change to a Page Flow symbol.

Figure 89 Change in Appearance of Service

Before Page Flow
is added
Service1
After Page Flow
is added
Service1

As a shortcut, you can drag the Page Flow from the Project Explorer onto a blank area of the design canvas. This action automatically creates a Service. The Service name is based on the Page Flow name.

6.6.2 Linking Each Service to a Web Connector

The Web Connector is a logical representation of the web container in which an eVision Studio web application runs.

In the following procedure, the binding box contains two types of services:

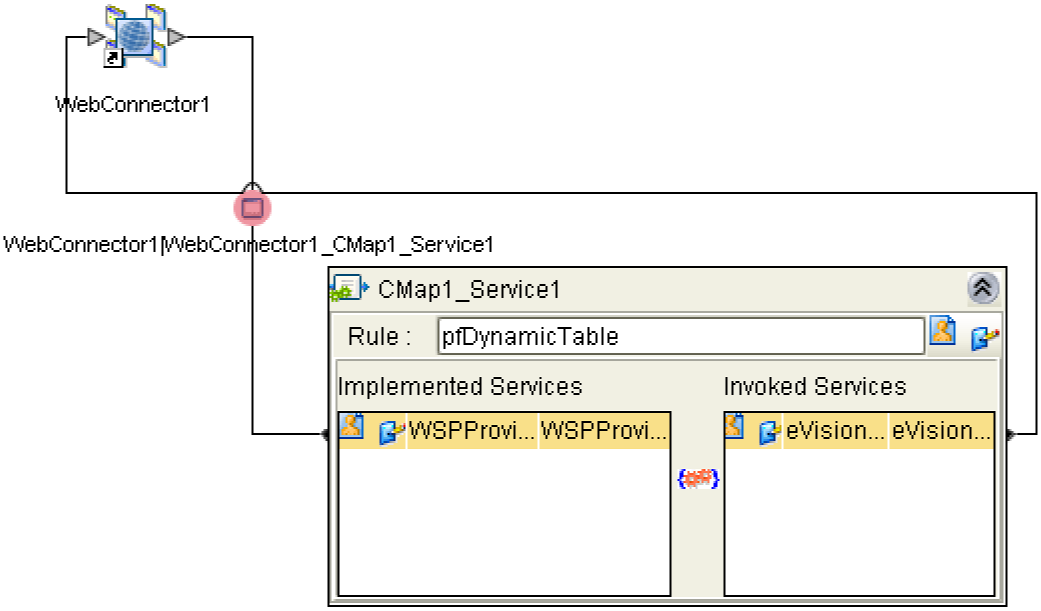
- Implemented Services represent the web services that are implemented and thus served by the Page Flow. The **WSPProvider** service is the entry point (the home page) of the web application.
- Invoked Services represent the web services that are called by the Page Flow.

To link each Service to a Web Connector

1. Drag a Web Connector from the Connectivity Map Editor toolbar onto the canvas.
2. Double-click each Service that contains a Page Flow.

   A binding box appears.
3. Link the **WSPProvider** Implemented Service to the Web Connector. Link the **eVision_user** Invoked Service to the Web Connector.

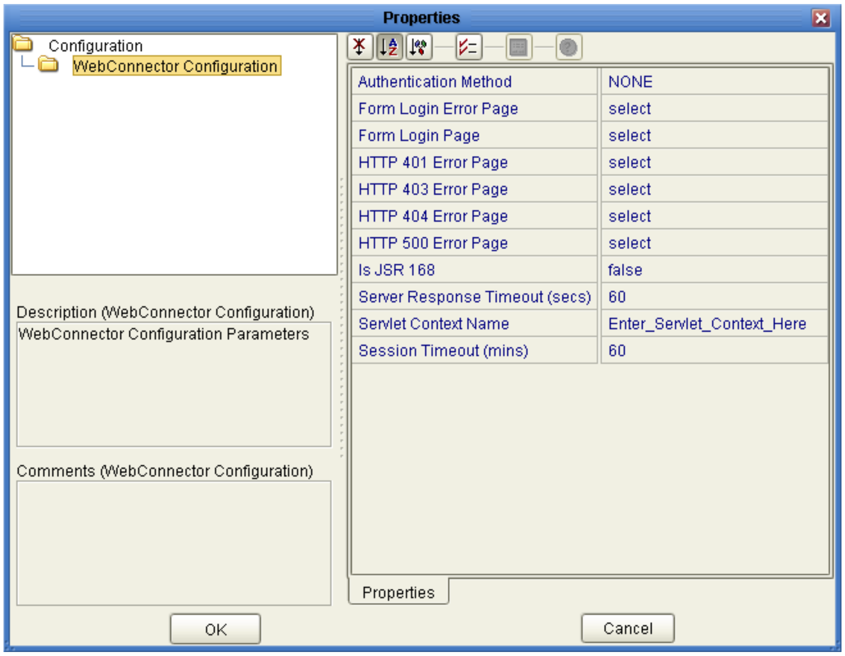
Figure 90 Service Binding Box



4 The web application consists of a presentation component and a back-end component. To configure properties for the presentation component of the application, double-click the connection icon between the **WSPProvider** Implemented Service and the Web Connector.

The **Properties** dialog box appears.

Figure 91 Web Connector Configuration Properties



Description (WebConnector Configuration)
WebConnector Configuration Parameters

Comments (WebConnector Configuration)

Authentication Method	NONE
Form Login Error Page	select
Form Login Page	select
HTTP 401 Error Page	select
HTTP 403 Error Page	select
HTTP 404 Error Page	select
HTTP 500 Error Page	select
Is JSR 168	false
Server Response Timeout (secs)	60
Servlet Context Name	Enter_Servlet_Context_Here
Session Timeout (mins)	60

5 If desired, change the values of one or more properties.

Chapter 10 **“Authentication, Authorization, and Error Handling”** describes the following properties: **Authentication Method**, **Form Login Error Page**, **Form Login Page**, **HTTP 401 Error Page**, **HTTP 403 Error Page**, **HTTP 404 Error Page**, and **HTTP 500 Error Page**.

Chapter 9 “Generating Portlets” on page 195 describes the **Is JSR 168** property.

The **Server Reponse Timeout** property is expressed in a whole number of seconds.

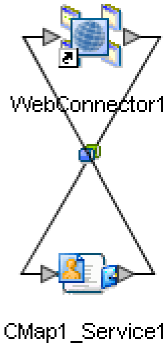
“Application URL” on page 161 describes the **Servlet Context Name** property.

The **Session Timeout** property is expressed in a whole number of minutes.

6 Click **OK**.

7 When you minimize the binding box, the connectors appear crossed. This behavior is normal.

Figure 92 Linked Service and Web Connector



6.7 Deploying Page Flows

The general steps for deploying a Project are:

1. Create an Environment
2. Create a Deployment Profile
3. Build an application file
4. Deploy the application file

For information about these general steps, see the *Sun SeeBeyond eGate Integrator User's Guide* and the *Sun SeeBeyond eGate Integrator System Administration Guide*.

The following information is specific to deploying Projects that contain Page Flows.

6.7.1 Application URL

To access an eVision Studio web application, users enter the *application URL* in the address field of their browser.

The application URL has the form:

```
http://hostname:portnumber/servletcontext
```

For example:

```
http://smith.company.com:18001/Project1Deployment1
```

Hostname

The hostname portion of the application URL represents the computer where the application server is running. Be sure to specify the fully qualified hostname. For example, use **smith.company.com** rather than **smith**.

Port Number

The port number is the HTTP port of the application server. If you are using the Sun SeeBeyond Integration Server on Windows platforms, then you can determine this value by starting the Domain Manager and checking the value of the HTTP row.

Servlet Context

The default value of the servlet context is the Project name concatenated with the Deployment Profile name. If you want to change the servlet context to a more user-friendly value, then perform the following procedure *before* building the application file.

To change the servlet context

1. In the Connectivity Map Editor, open the Service binding box and double-click the connection icon between the **WSPProvider** Implemented Service and the Web Connector.

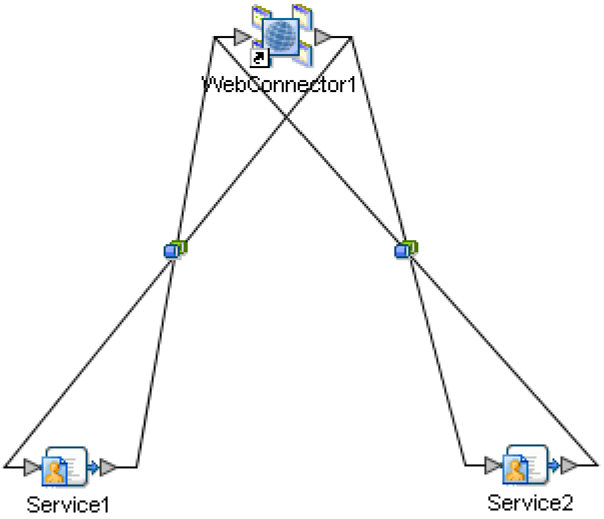
2 Change the value of the **Servlet Context Name** property.

3 Click **OK**.

Multiple Web Applications

eVision Studio support multiple web applications in the same deployment.

Figure 93 Connectivity Map with Multiple Web Applications



You must configure each application to have a different URL. To do this, open the connection icons and ensure that the **Servlet Context Name** properties have different values (for example, **WebApp1** and **WebApp2**).

If you do not perform this step, then the generated web application will not deploy, because multiple **.war** files will conflict with the same servlet context.

6.7.2 Manually Passing in Parameters

You can add one or more parameters to an application URL. The Page Flow can be designed to make a decision based on the value of a parameter. Because the value may differ among invocations, the same Page Flow may be invoked with different URLs.

Note: *In release 5.0.x, you had to include the following text in the URL when passing in parameters:* **input_link_Message_u002F_Input_u002F_**. *This approach is supported for backward compatibility.*

Home Page

If you want to pass one or more parameters into the home page, then add the parameters to the end of the application URL.

The following example passes one parameter into the home page.

```
http://myhostname:18001/Project1Deployment1?username=bill
```

The following example passes three parameters into the home page. The URL appears on multiple lines for readability.

```
http://myhostname:18001/Project1Deployment1?
model=accord&make=honda&year=2005
```

Non-Home Page

If you want to pass one or more parameters into a Page Layout other than the home page, then you must add the following text before the first parameter. In the port type, *LinkName* refers to the Page Link that points to the Page Layout.

```
PortType={urn:ProjectName_LinkName}link&Operation=link&
```

The following example passes one parameter into a Page Layout other than the home page. The URL appears on multiple lines for readability.

```
http://myhostname:18001/Project1Deployment1?
PortType={urn:Project1_link3}link&Operation=link&username=bill
```

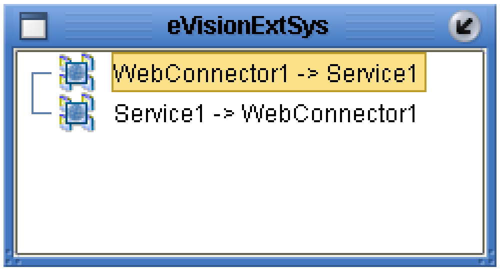
6.7.3 eVision External System

When you create the Environment, you must include an eVision External System. The eVision External System is a logical representation of the web container in which an eVision Studio web application runs. The eVision External System is where the presentation component of the application will execute.

When you create the Deployment Profile, some of the components in the left panel of the Deployment Profile Editor will contain a Web Connector. You can click the **Automap** icon, or you can manually drag these components into the eVision External System.

Figure 94 shows Web Connector components that have been deployed to an eVision External System.

Figure 94 Web Connector Components in eVision External System



WebConnector1 -> Service1

Service1 -> WebConnector1

The eVision External System contains properties that are used in authentication. For more information, see Chapter 10 **"Authentication, Authorization, and Error Handling"**.

6.7.4 SSL Support

You can configure an eVision Studio web application to use the Secure Sockets Layer (SSL), which is designed to protect communication between clients and servers over the Internet. The protocol portion of the application URL will be **https** instead of **http**.

For detailed instructions, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

6.8 Monitoring eVision Studio Projects

Enterprise Manager enables you to monitor deployed Projects.

To monitor eVision Studio Projects, you must first install the eVision Enterprise Manager Plug-In. For instructions, see **Chapter 3 "Installing eVision Studio"**.

The following procedures describe how to monitor Web Connectors and Web Connector links.

To monitor Web Connectors

1. In the Explorer panel of Enterprise Manager, select the Connectivity Map.
2. In the Details panel of Enterprise Manager, select the Web Connector in the Connectivity Map. The **Status** tab displays basic information about the Web Connector.
   - The **HostAndPort** row displays the computer name and administrative port on which the component is running.
   - The **System** row indicates whether the component is located in the 4.5.*x* tree or the 5.1.*x* tree.
   - The **Component** row displays the hierarchy of the component in the Explorer panel.
   - The **State** row indicates the current status of the component.
   - The **ComponentType** row indicates that the component is a Web Connector.
3. To prevent new sessions from being initiated, click **Stop**. To allow new sessions once again, click **Start**.
4. The **Summary** tab displays the components that are located at the same hierarchical level in the Explorer panel.

   For information about the **Logging** and **Alerts** tabs, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

To monitor Web Connector links

1. In the Explorer panel of Enterprise Manager, select the Connectivity Map.
2. In the Details panel of Enterprise Manager, select the connection icon between the Service and the Web Connector.
   - The **Component** row displays the hierarchy of the component in the Explorer panel.
   - The **State** row indicates the current status of the component.
   - The **Operation** row indicates the type of web service operation.
   - The **Destination** row indicates the name of the Service.
   - The **ComponentType** row indicates that the component is a Web Connector link.

- The **PortType** row indicates the name of the port type that contains the web service operation.

For information about the **Logging** and **Alerts** tabs, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

6.9 Invoking Another Page Flow or a Business Process

From within a Page Flow, you can invoke another Page Flow or an eInsight Business Process. Thus, you can create reusable Page Flows and Business Processes.

The *Sun SeeBeyond eInsight Business Process Manager User's Guide* describes how to incorporate sub-processes into business models. You can use the same approach to invoke another Page Flow or a Business Process.

Chapter 7

Creating Charts

The Chart object is available from the **Extension Objects** palette in the Page Layout Designer.

What's in This Chapter

7.1 Charts Overview

eVision Studio provides a variety of predefined chart types, including area charts, bar charts, line charts, and pie charts.

The process of creating a chart is divided into two phases:

1. Adding the chart to a Page Layout in the Page Layout Designer. For instructions, see **"Adding a Chart to a Page Layout" on page 167**.
2. Mapping data into the chart in the Page Flow Designer. For instructions, see **"Mapping Data into the Chart" on page 177**.

7.2 Adding a Chart to a Page Layout

The Page Layout Designer enables you to add one or more charts to a Page Layout.

To add a chart to a Page Layout

1. From the **Extension Objects** palette, drag the **Chart** object onto the canvas.

   A bar chart with boilerplate information appears.

2. If you want to change the chart type, then do the following in the **Properties** tab:

   A. In the left column, click the **Type** property.

   B. In the right column, select the chart type from the drop-down menu. **"Chart Types" on page 168** describes the available chart types.

3 Specify the behavior of each data set. See **"Data Sets" on page 174**.

4 Set additional properties as needed. See **"Additional Properties" on page 176**.

5 To preview the chart, click the **Preview** icon on the Page Layout Designer toolbar. Because you have not mapped data into the chart, you see only boilerplate information. The actual data appears at runtime.

7.2.1 Chart Types

The available chart types are divided into the following categories: area, bar, line, pie, scatter plot, waterfall, and xy step area.

Area

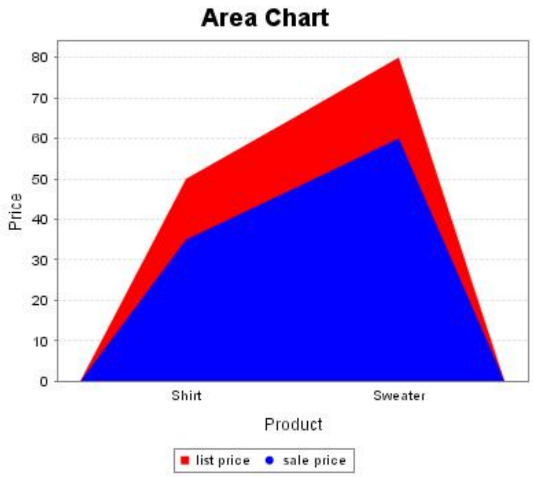
An area chart fills in the portion of the chart between the category axis and the lines that connect the data points. eVision Studio provides the following area chart types:

- **areaChart**
- **stackedAreaChart**
- **xyAreaChart**

Figure 95 shows an example of the **areaChart** type. The source data for the example is:

PRODUCT_NAME	LIST_PRICE	SALE_PRICE
Shirt	50	35
Sweater	80	50

Figure 95 areaChart Example



The **xyAreaChart** type displays (x,y) pairs of data. Rather than presenting a category axis and a value axis, this chart type presents two value axes.

Bar

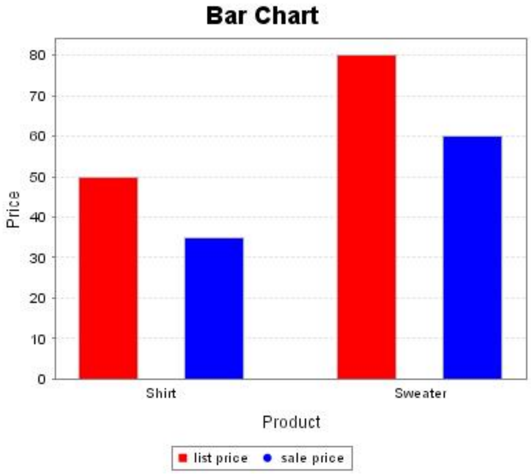
A bar chart displays the data points as vertical rectangles. eVision Studio provides the following bar chart types:

- **barChart**
- **barChart3D**
- **stackedBarChart**
- **stackedBarChart3D**

Figure 96 shows an example of the **barChart** type. The source data for the example is:

PRODUCT_NAME	LIST_PRICE	SALE_PRICE
Shirt	50	35
Sweater	80	50

Figure 96 barChart Example



The **barChart3D** and **stackedBarChart3D** types provide a three-dimensional visual effect.

The **stackedBarChart** and **stackedBarChart3D** types use a single bar to display the values for each category.

Line

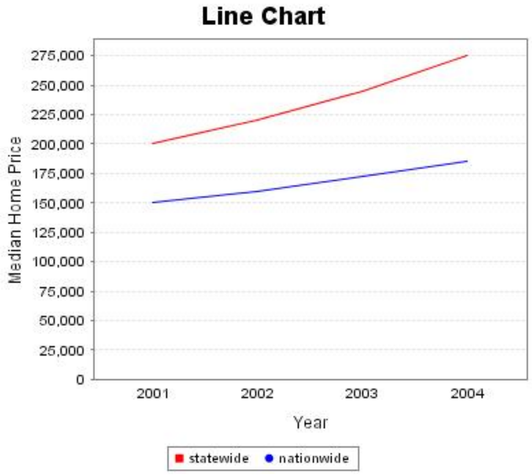
A line chart connects the data points with lines. eVision Studio provides the following line chart types:

- **lineChart**
- **xyLineChart**

Figure 97 shows an example of the **lineChart** type. The source data for the example is:

YEAR	STATE	NATION
2001	200000	150000
2002	220000	160000
2003	245000	172000
2004	275000	185000

Figure 97 lineChart Example



The **xyLineChart** type displays (x,y) pairs of data. Rather than presenting a category axis and a value axis, this chart type presents two value axes.

Pie

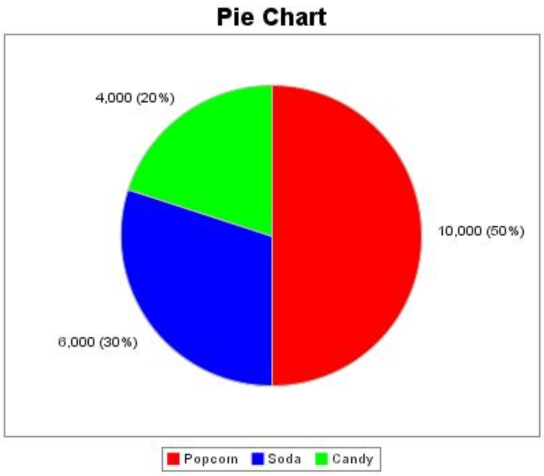
A pie chart displays the data as a circle that has been divided into two or more wedge-shaped segments. eVision Studio provides the following pie chart types:

- **pieChart**
- **pieChart3D**

Figure 98 shows an example of the **pieChart** type. In the chart, each segment is labeled with the raw number and the percentage. The source data for the example is:

PRODUCT_NAME	SALES
Popcorn	10000
Soda	6000
Candy	4000

Figure 98 pieChart Example



The **pieChart3D** type provides a three-dimensional visual effect.

Scatter Plot

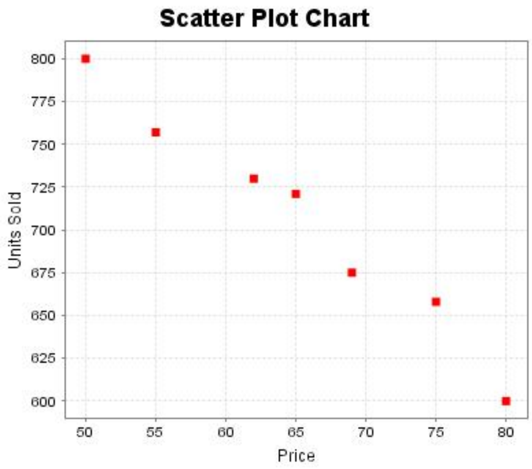
A scatter plot chart illustrates the correlation between (x,y) pairs of data. Rather than presenting a category axis and a value axis, this chart type presents two value axes. Each data point is represented by a dot. eVision Studio provides the following scatter plot type:

- **scatterPlotChart**

Figure 99 shows an example of the **scatterPlotChart** type. In this example, the data has a high negative correlation. As the price goes up, the number of units sold goes down. The source data for the example is:

PERIOD	UNITS_SOLD
50	800
55	757
62	730
65	721
69	675
75	658
80	600

Figure 99 scatterPlotChart Example



Waterfall

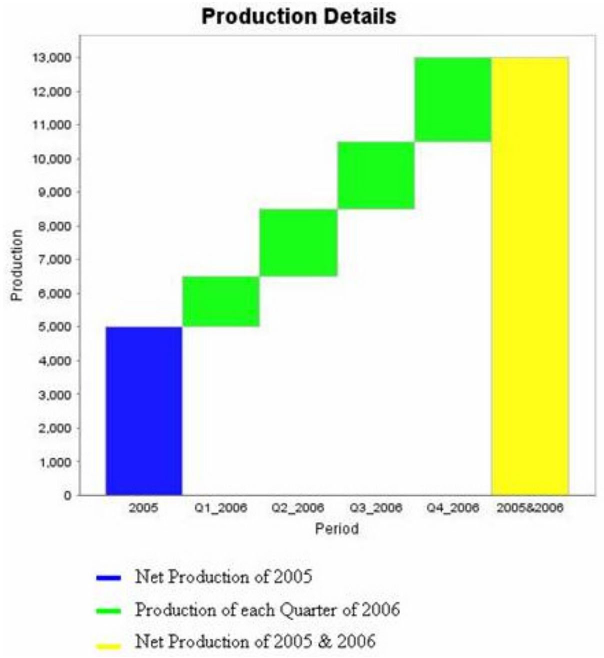
A waterfall chart is a variation of the bar chart type. The waterfall chart displays the bars as floating vertical rectangles. eVision Studio provides the following waterfall chart type:

- **waterfallChart**

Figure 100 shows an example of the **waterfallChart** type. The source data for the example is:

PERIOD	UNITS PRODUCED
2005	5000
Q1_2006	1500
Q2_2006	2000
Q3_2006	2000
Q4_2006	2500
2005 & 2006	13000

Figure 100 waterfallChart Example



XY Step Area

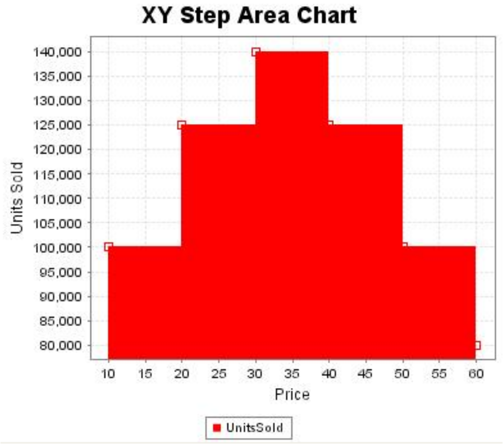
An xy step area chart illustrates the correlation between (x,y) pairs of data. Rather than presenting a category axis and a value axis, this chart type presents two value axes. Each data point is represented by a vertical rectangle. The rectangles are joined in a way that makes them resemble a series of steps. eVision Studio provides the following xy step area chart type:

- **xyStepAreaChart**

Figure 101 shows an example of the **xyStepAreaChart** type. The source data for the example is:

PRICE	UNITS_SOLD
10	100000
20	125000
30	140000
40	125000
50	100000
60	80000

Figure 101 xyStepAreaChart Example

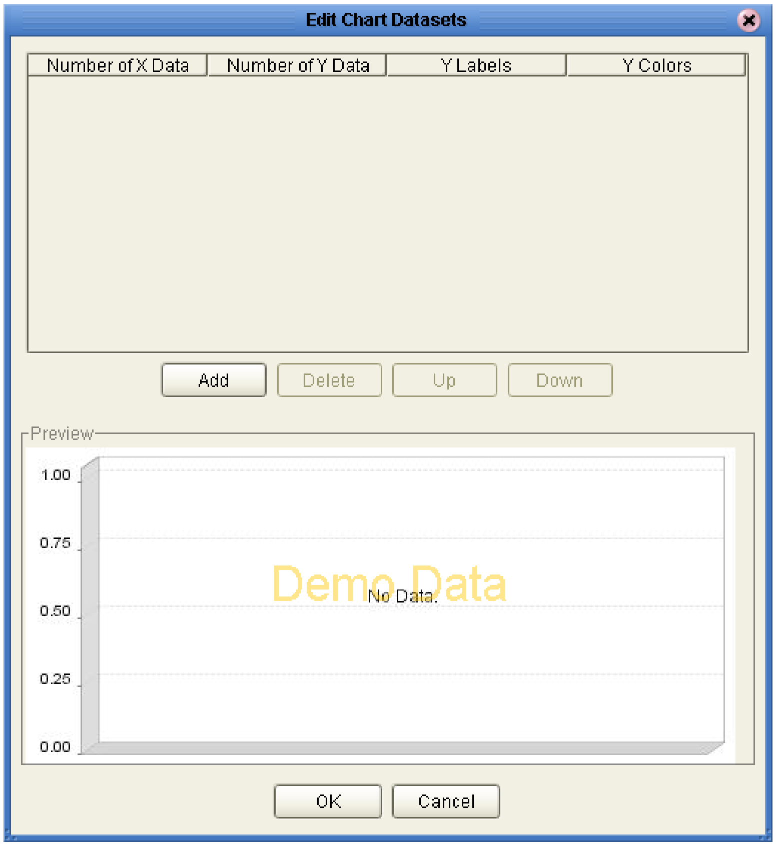


7.2.2 Data Sets

A data set is a group of related data points. Each chart has one or more data sets.

The **Datasets** property enables you to specify the characteristics of each data set. When you click the right column, the **Edit Chart Datasets** dialog box appears.

Figure 102 Edit Chart Datasets Dialog Box



For the chart types that are based on categories, you typically need to specify only one data set. For the chart types that are based on xy series, you might need to specify more than one data set. Table 24 lists the number of data sets for each chart type.

Table 24 Chart Types and Number of Data Sets

Chart Type	Based On	Number of Data Sets
areaChart	Categories	one
stackedAreaChart	Categories	one
xyAreaChart	XY Series	one or more
barChart	Categories	one
barChart3D	Categories	one
stackedBarChart	Categories	one
stackedBarChart3D	Categories	one
lineChart	Categories	one
xyLineChart	XY Series	one or more
pieChart	Categories	one
pieChart3D	Categories	one
scatterPlotChart	XY Series	one or more
waterfallChart	Categories	one
xyStepAreaChart	XY Series	one or more

The lower portion of the **Edit Chart Datasets** dialog box shows a demonstration of how the chart will appear at runtime.

You map data into the data set elements in the Page Flow Designer. See **"Mapping Data into the Chart" on page 177**.

Guidelines for Category Series Charts

The **Number of X Data** column specifies the number of input sources to be used for the categories. For example, if you were creating the bar chart in **Figure 96 on page 169**, you would set this column to 1.

Note: *Setting the column to 1 does not mean that only one category appears at runtime. The number of categories depends on the values in the input source.*

The **Number of Y Data** column specifies the number of values for each category. For example, if you were creating the bar chart in **Figure 96 on page 169**, you would set this column to 2.

If you are creating a pie chart, then you must set the **Number of Y Data** column to 1.

If you are creating a stacked bar chart, then setting both the **Number of X Data** column and **Number of Y Data** column to 1 enables multiple values to appear for each bar.

The **Y Labels** column enables you to specify the names in the chart legend. Separate the values with a comma. For example:

```
list price,sale price
```

The **Y Colors** column enables you to specify the colors of the value shapes. Separate the values with a comma. For example:

```
blue,green
```

Guidelines for XY Series Charts

The **Number of X Data** column specifies the number of input sources to be used for the x portion of the (x,y) pairs. For example, if you were creating the scatter plot chart in **Figure 99 on page 172**, you would set this column to 1.

The **Number of Y Data** column specifies the number of input sources to be used for the y portion of the (x,y) pairs. For example, if you were creating the scatter plot chart in **Figure 99 on page 172**, you would set this column to 1.

The **Y Labels** column enables you to specify the names in the chart legend. For example:

```
Price,Units Sold
```

If the number of (x,y) pairs in each series is different, then you must specify more than one data set.

7.2.3 Additional Properties

In addition to the **Datasets** and **Type** properties, the Chart object includes the following properties. Some of these properties appear only for certain chart types.

- The **Background Color** property specifies the color that appears in the background of the chart.
- The **Grid Line Color** property specifies the color of grid lines (if included).
- You can change the height of a chart by using the mouse, or by changing the value of the **Height** property.
- The **Is Clickable** property is used in the procedure for enabling clickability in a chart. See **"Enabling Clickability in a Chart" on page 179**.
- The **Item Label Format** property defines the pattern and format for item labels. For detailed information, see the documentation for the **DecimalFormat** class at http://java.sun.com/j2se/1.4.2/docs/api/java/text/DecimalFormat.html.
- The **Legend** property indicates the location of the chart legend: north, east, south, or west. You can also remove the chart legend.

Note: *If you want a waterfall chart to include a legend, then set the value of the* **Legend** *property to none and use the* **X-Axis Label** *property to add the legend.*

- The **Orientation** property indicates whether the chart orientation is vertical or horizontal. The default setting is vertical.

Note: *The pie chart types do not use the* **Orientation** *property.*

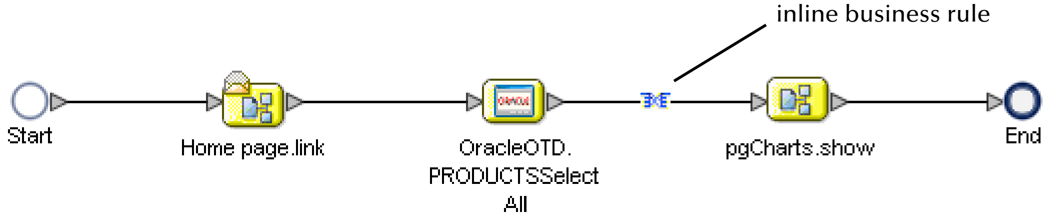
- The **Plot Background Color** property specifies the color that appears in the data portion of the chart.
- The **Show Grid Line** property indicates whether grid lines appear for the value axis.
- The **Show Item Label** property indicates whether each value item (such as a bar) has a label that displays the value.
- The **Transparency** property enables you to display overlapping data. Enter a value from **0.0** (transparent) through **1.0** (opaque).
- You can change the width of a chart by using the mouse, or by changing the value of the **Width** property.
- The **X Label Positions** property enables you to change the rotation of the category axis labels.
- The **X-Axis Label** property enables you to specify a label for the category axis.
- The **Y-Axis Label** property enables you to specify a label for the value axis.
- The following properties enable you to control the bar colors in a waterfall chart: **First Bar Color**, **Last Bar Color**, **Negative Bar Color**, and **Positive Bar Color**.
- The **ID** property contains the unique HTML identifier for the object. This property is read only.
- For information about the **Style** property, see **"Applying a Local Style" on page 57**.
- For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.
- The **Title** property enables you to add a title to the top of the chart. In addition, the value appears as a tooltip.

7.3 Mapping Data into the Chart

After adding the chart to a Page Layout, you specify what data will appear in the chart. You perform this step in the Page Flow Designer.

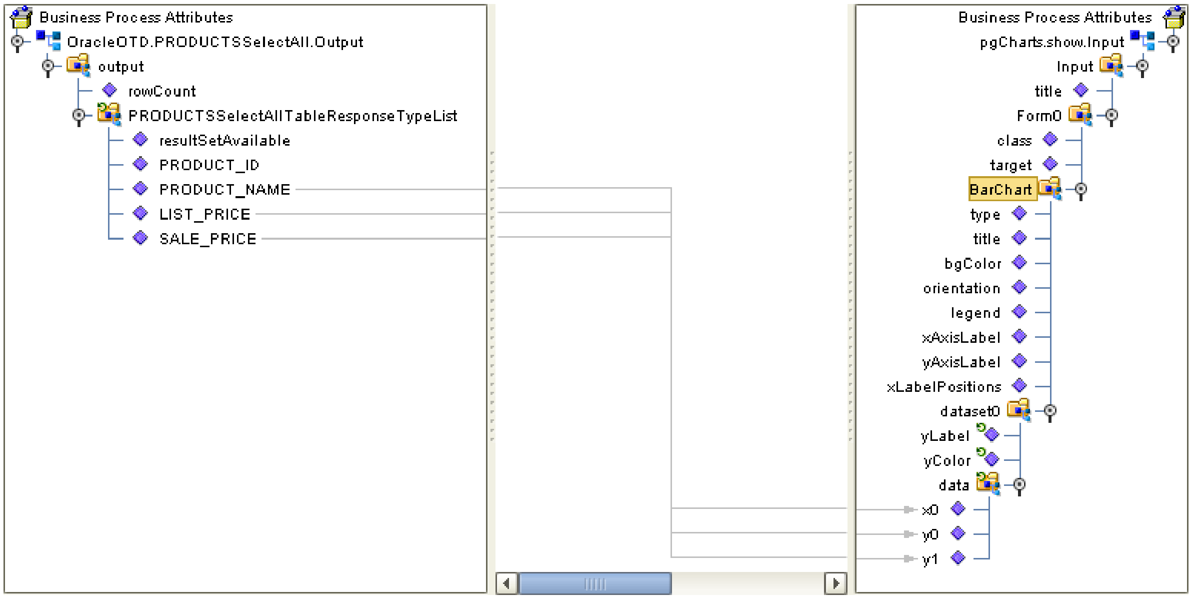
The example in this section maps data from an Oracle database into the chart. Figure 103 shows the Page Flow.

Figure 103 Page Flow for Mapping Example



The mapping occurs in the inline business rule between the **SelectAll** operation of the Oracle OTD and the **show** operation of the Page Layout. Figure 104 shows the Business Rule Designer view of the inline business rule.

Figure 104 Inline Business Rule Mapping



The Output pane (on the left) represents the output from the **SelectAll** operation of the Oracle OTD. The OTD was created for a table called **PRODUCTS**. The table contains the following columns: **PRODUCT_ID**, **PRODUCT_NAME**, **LIST_PRICE**, and **SALE_PRICE**. Notice that the **PRODUCTSSelectAllTableResponseTypeList** node is a repeating node.

The Input pane (on the right) represents the input to the Page Layout. Each chart in the Page Layout is identified by its **LName** property. Figure 104 shows one chart named **BarChart**.

The data set in each chart is identified by **datasetn**. The numeric portion starts from 0.

The x and y nodes under each data set correspond to the values in the **Edit Chart Datasets** dialog box. The numeric portion starts from 0. For example, if you enter 2 in the **Number of X Data** column and 4 in the **Number of Y Data** column, then the following nodes appear:

- x0
- x1
- y0
- y1
- y2
- y3

In the example, the **PRODUCT_NAME** column represents the categories that will appear in the bar chart. Therefore, this column is mapped to the **x0** node.

The **LIST_PRICE** and **SALE_PRICE** columns represent the values that will appear for each category. Therefore, these columns are mapped to the **y0** and **y1** nodes, respectively.

Figure 96 on page 169 shows how the chart appears when the Project is deployed. The chart has two categories: Shirt and Sweater. Each category has two values: list price and sale price.

7.3.1 Stacked Bar Chart Mapping

The mapping behavior of stacked bar charts requires special considerations.

When you specify the data set in the Page Layout Designer, you must set the **Number of X Data** column to 1 and the **Number of Y Data** column to 1.

When you perform the mapping in the Business Rule Designer, do the following:

- Map the category to the **x0** node.
- Map the quantity to the **y0** node.
- Map the series (that is, a subcategory) to the **yLabel** node.

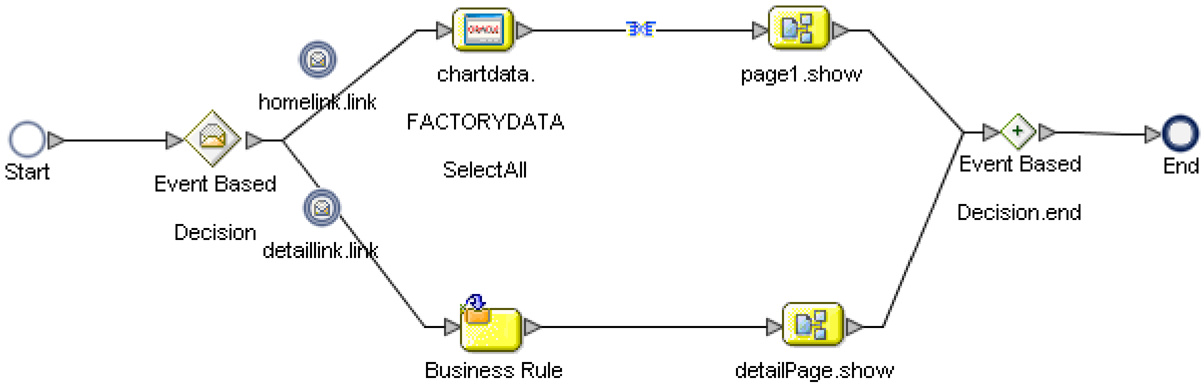
7.4 Enabling Clickability in a Chart

You can configure a chart so that data points within the chart are clickable. At runtime, when the user clicks a data point, an additional page is displayed. Typically, the additional page contains more detail about the area that the user clicked.

The example in this section maps a stacked bar chart that contains factory data to fields in a detail page.

Figure 105 shows the Page Flow for the example. The **page1** Page Layout contains the chart. When the user clicks a data point in the chart, the **Event Based Decision** element passes control to the **detaillink** Page Link. The mapping takes place in the **Business Rule** element. The **detailPage** Page Layout contains the fields.

Figure 105 Page Flow for Clickability Example



7.4.1 Clickability Parameters

Each chart exposes a set of clickability parameters. The parameters depend on the chart type. The following parameters are common to all chart types:

- type
- orientation
- legend
- bg Color
- xAxisLabel
- yAxisLabel
- title

Note: *All charts that contain x and y parameters can only accept numeric values.*

The following parameters are specific to the **areaChart**, **stackedAreaChart**, **barChart**, **barChart3D**, **stackedBarChart**, **stackedBarChart3D**, **lineChart**, and **waterfallChart** chart types:

- series
- category
- value
- X Label Positions

The following parameters are specific to the **xyAreaChart**, **xyLineChart**, **scatterPlotChart**, and **xyStepAreaChart** chart types:

- series
- xvalue
- yvalue
- X Label Positions

The following parameters are specific to the **pieChart** and **pieChart3D** chart types:

- section
- value

7.4.2 Configuring the Project Components

The Project components that you must configure are the Page Layout, Page Link, and Page Flow.

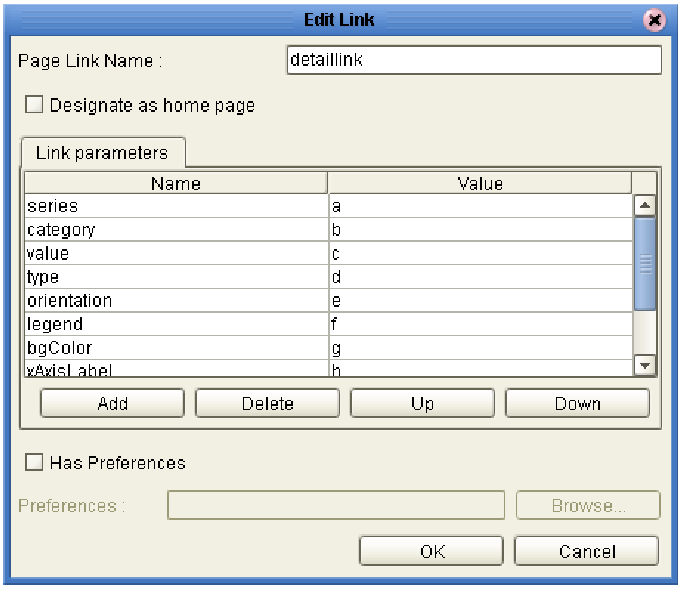
To configure the Project components

1. In the Page Layout Designer, set the **Is Clickable** property of the chart object to **true**.
2. If you want to create a pop-up window, set the **On Click** property of the **Area** property type to the **window.open()** method. For example:

```
window.open(this.href,'reassignWindow1','titlebar=no,toolbar=no,me
nubar=no,location=no,scrollbars=no,resizable=yes,status=no,width=6
00,height=700');return false;
```

3. Create a Page Link. Add the parameters that you want to expose in the Business Rule Designer. You can add custom parameters.

Figure 106 Page Link for Clickability Example



Page Link Name : detaillink

☐ Designate as home page

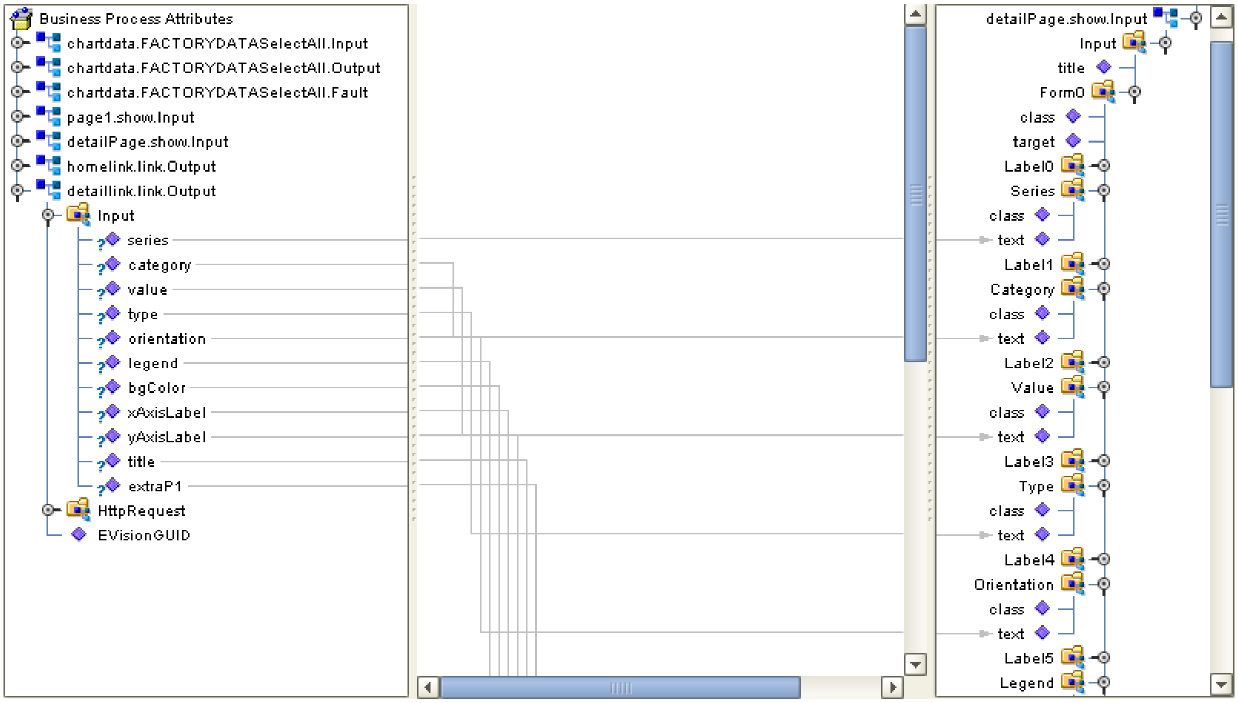
Name	Value
series	a
category	b
value	c
type	d
orientation	e
legend	f
bgColor	g
xAxisLabel	h

☐ Has Preferences

Important: *The parameters that the charts expose are case sensitive. For example, you must enter* **series** *rather than* **Series***.*

4. Create a Page Flow. In the Business Rule Designer, map parameters from the chart outputs to objects in the detail page.

Figure 107 Business Rule Mapping for Clickability Example



In the left side of Figure 107, the **detaillink.link.Output** node contains the predefined parameters for the **stackedBarChart** chart type, as well as a custom parameter called **extraP1**.

In the right side of Figure 107, each parameter is mapped to an HTML object in the **detailPage** Page Layout. The **LName** properties of the HTML objects are named based on the type of parameter that the objects will display at runtime: **Series**, **Category**, **Value**, and so on.

Chapter 8

Creating Custom Objects

eVision Studio allows you to create custom objects for use in Page Layouts.

What's in This Chapter

8.1 Custom Objects Overview

To create a custom object, you can either create the custom object directly in Enterprise Designer, or first create the files described in Table 25 (The XML file is required; the other files are optional) and then create a zip file for them.

Table 25 Custom Object Files

File	Description	Required
XML	Defines the basic characteristics of the object.	Yes
XSD Input	Defines one or more elements that can receive input.	No
XSD Output	Defines one or more elements that can produce output.	No
JSP	Contains runtime code for the object.	No
Icon	Contains the image that appears for the custom object in the object palette and on the design canvas. Note that no Icon can be attached to a custom object that is created in Enterprise Designer.	No

After packaging these files in a .zip file, you import the .zip file into the Page Layout Designer.

When you drag a custom object onto the design canvas, the following properties appear in the **Properties** tab:

- The **LName** property is the internal name for the object. This name appears in the mapping area of the Page Flow Designer. You can change the default value.

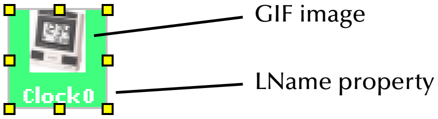
- For information about the **Class** property, see **"Applying Classes" on page 59**.
- The **ID** property contains the unique HTML identifier for the object. This property is read only.
- For information about the **Style** property, see **"Applying a Local Style" on page 57**.
- For information about the **Z-Index** property, see **"Configuring the Z-Direction of Objects" on page 49**.
- The **Default XML Input** property enables you to specify input data that will be used if no input data is available at runtime.
- The **JSP Source** property displays the content of the JSP file (if included). This property is read only.
- The **XSD Input** property displays the content of the XSD input file (if included). This property is read only.
- The **XSD Output** property displays the content of the XSD output file (if included). This property is read only.

In addition, the custom object has the following internal properties. These properties do not appear in the **Properties** tab, but you can manipulate them in the JSP file.

- The **position** property specifies whether the object has absolute or relative positioning on the Web page.
- For absolute positioning, the **top** property specifies how far the object is from the top border of the page. For relative positioning, the **top** property specifies how far the object is from the top border of its natural location.
- For absolute positioning, the **left** property specifies how far the object is from the left border of the page. For relative positioning, the **left** property specifies how far the object is from the left border of its natural location.
- The **width** property enables you to specify how wide the object is.
- The **height** property enables you to specify how tall the object is.

When you drag a custom object onto the design canvas, the object appears as a green square. The GIF image and the value of the **LName** property are displayed.

Figure 108 Custom Object on Design Canvas



If you import a Project that contains a custom object, then the custom object must already have been installed in the Page Layout Designer. Otherwise, the custom object will not appear in the Page Layouts.

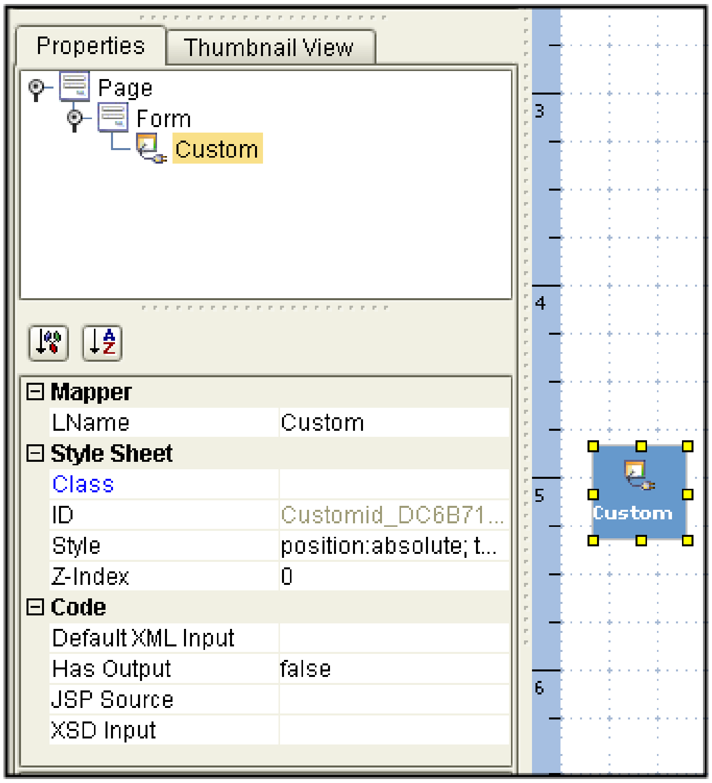
You cannot add Java libraries for the custom object. For example, you cannot add a Java library that would enable the object to use Java™ DataBase Connectivity to interact with a database.

8.2 Creating Custom Object Files with Enterprise Designer

To Create custom object files with Enterprise Designer

1. Drag the Custom icon from the object pallet into the design canvas.
2. Click on the Custom object. The properties for the custom object are displayed in the property sheet (as shown in Figure 109).

Figure 109 Creating a Custom Object in Enterprise Designer



⊟ Mapper	
LName	Custom
⊟ Style Sheet	
Class	
ID	Customid_DC6B71...
Style	position:absolute; t...
Z-Index	0
⊟ Code	
Default XML Input	
Has Output	false
JSP Source	
XSD Input	

3. In the property sheet, set the paramters for the custom object.

Note: *When creating a custom object in the Enterprise Designer, a custom image cannot be used as its icon when attaching it to the object.*

When you drop a Custom Object from the palette onto the canvas and edit the properties (mentioned above), it creates a single instance of the Custom Object. If you think you may want to reuse the code, simply save as a Template Object. This allows you to reuse the Custom Object other places just like other Template Objects. This instance is not linked to changes in the Custom Object definition. A Custom Object created in this way will not automatically update itself if the user changes the Custom Object properties. As such, this Custom Object is more like a template.

8.3 Creating a Custom Object with XML Files

You must create an XML file called **custom.xml**. You can optionally create an XSD input file, XSD output file, JSP file, and icon file.

When you are done creating the custom object files, package them in a .zip file. The files must be in the top-level directory of the .zip file.

After you package the files, follow the instructions in **"Installing Custom Objects" on page 191**.

Custom Object Zip Files are not imported and exported with Projects and must managed separately. Prior to loading a Project which uses a Custom Object Zip, you should have the Custom Object Zip installed. If the Custom Object Zip is not installed, the items will not appear on a Page Layout canvas (when opened) and the XSD fragments will not be available in the mapper.

Saving a Page Layout without the Custom Zips installed, results in the loss of all Custom Objects from the Page Layout canvas. Simply opening and closing a Page Layout has no effect.

Note: *When a Repository Custom Object is created and dropped on multiple pages, if the Custom Object definition is updated, the changes will be reflected on all pages.*

8.3.1 XML File

The XML file defines the basic characteristics of a object. The file must be called **custom.xml**.

The root element must be **<eVisionCustomObject>**.

The **<eVisionCustomObject>** tag contains the following attributes:

- The **name** attribute specifies the name of the object. This name appears in the object palette.
- The **description** attribute specifies the description of the object. This description appears in the palette when the Page Layout Designer user places the mouse pointer over the object.
- The **iconFile** attribute specifies the name of the icon file. If you are not creating an icon file, then you do not need to include this attribute.
- The **jspSourceFile** attribute specifies the name of the JSP file. If you are not creating a JSP file, then you do not need to include this attribute.
- The **xsdInputFile** attribute specifies the name of the XSD input file. If you are not creating an XSD input file, then you do not need to include this attribute.
- The **xsdOutputFile** attribute specifies the name of the XSD output file. If you are not creating an XSD output file, then you do not need to include this attribute.

Here is a sample **custom.xml** file.

```
<eVisionCustomObject
    name="hello"
```

```
        description="Hello World"
        iconFile="helloIcon.gif"
        jspSourceFile="helloJSP.jsp"
        xsdInputFile="helloIn.xml"
        xsdOutputFile="helloOut.xml">
    </eVisionCustomObject>
```

8.3.2 XSD Files (Optional)

The XML Schema Definition (XSD) files describe data structures that can be manipulated in an eInsight Business Process or an eVision Page Flow.

You can create an input file, an output file, or both an input file and output file.

XSD Input File

If you want the custom object to include one or more elements that can receive input, then create an XSD input file.

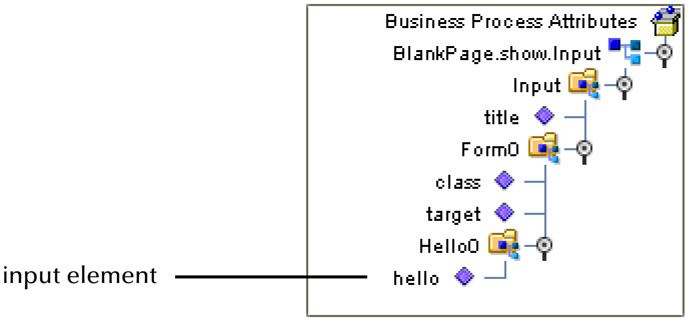
The XSD must be well formed. Data types must be **xsd:string**.

The following example defines an input element called **hello**.

```
<xsd:element type="xsd:string" name="hello"/>
```

Figure 110 shows how this input element appears in the Page Flow Designer.

Figure 110 Input Element in Page Flow Designer



The following example defines an input element that contains a subnode.

```
<xsd:element name="node">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element type="xsd:string" name="text"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

XSD Output File

If you want the custom object to include one or more elements that can produce output, then create an XSD output file.

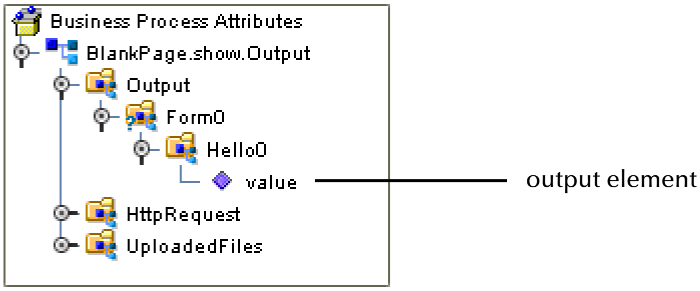
The XSD must be well formed.

The following example defines an output element called **value**.

```
<xsd:element type="xsd:string" name="value"/>
```

Figure 111 shows how this output element appears in the Page Flow Designer.

Figure 111 Output Element in Page Flow Designer



8.3.3 JSP File (Optional)

The JavaServer Pages (JSP) file contains runtime code for the object. When eVision Studio displays the Page Layout in a browser, the content of the JSP file is inserted into the HTML.

The following example creates a form element.

```
<span style="<%=customObject.getProperty("style")%>">
<input
    name="<%=customObject.getXMLOutputPath("value")%>"
    value="<%=customObject.getXMLInput().getValue("hello")%>">
</input>
</span>
```

The JSP file is optional. For example, if you want the custom object to pass data through without displaying the data, then you do not need to create a JSP file.

customObject Variable

The **customObject** runtime variable enables you to access the runtime information of the custom object.

The **customObject** runtime variable provides the following methods:

- getProperty()
- getXMLInput()
- getXMLOutputPath()

The **getProperty()** method returns the string value of an object property.

This method accepts a **java.lang.String** object that contains the internal property name. The method returns a **java.lang.String** object that contains the property value.

The following example uses this method to obtain the values of the **style** and **class** properties.

```
<div
    style="<%=customObject.getProperty("style")%>"
    class="<%=customObject.getProperty("class")%>"
>
    ...
</div>
```

If the object is not part of a top-level element, then you could obtain the values of the **style** and **class** properties from within the object. For example:

```
<input
    style="<%=customObject.getProperty("style")%>"
    class="<%=customObject.getProperty("class")%>"
    ...
</input>
```

The **getXMLInput()** method retrieves the Java representation of the input data.

This method does not accept any input parameters. The method returns an **XMLNode** object. For information about the **XMLNode** object, see **"XMLNode Object" on page 190**.

The following example calls this method.

```
<%
XMLNode XMLNode = customObject.getXMLInput();
    ...
%>
```

The **getXMLOutputPath()** method enables you to set output values for properties.

This method has two forms:

- The first form accepts one input parameter: a **java.lang.String** object that contains an XPath expression.
- The second form accepts two input parameters: a **java.lang.String** object that contains an XPath expression, and a **java.lang.String** object that contains the associated string value.

The return value for both forms is a **java.lang.String** object that represents the internal path to the specified XML node.

The following example shows both forms.

```
<input
    type="checkbox"
    name="<%=customObject.getXMLOutputPath("groups")%>"
    value="<%=customObject.getXMLOutputPath("groups/group1[0]",
    "group1")%>"
>
</input>
```

An index of 0 specifies that the XML node and value will be appended to the target path.

XMLNode Object

The **getXMLInput**() method of the **customObject** runtime variable returns an **XMLNode** object.

This object is a wrapper around the **org.w3c.dom.Node** interface.

The **XMLNode** object provides the following methods:

- getName()
- getNode()
- getNodeCount()
- getNodeList()
- getValue()

The **getName()** method returns the name of the XML node.

The **getNode()** method returns the XML node specified by the XPath expression.

The **getNodeCount()** method returns the number of XML nodes returned for an XPath expression.

The **getNodeList()** method returns a list of XML nodes for an XPath expression.

The **getValue()** method returns the value of an XML node. This method has three forms:

- The first form does not accept any input parameters.
- The second form accepts one input parameter: a **java.lang.String** object that contains an XPath expression.
- The third form accepts two input parameters: a **java.lang.String** object that contains an XPath expression, and a **java.lang.String** object that contains a default value.

8.3.4 Icon File (Optional)

The icon file contains the image that appears for the custom object in the object palette and on the design canvas. The file must be in GIF format.

The object palette displays the image as 16 by 16 pixels. On the design canvas, the image may appear larger, but does not become larger than its original size.

If you do not include an icon file, then a default icon appears.

8.4 Installing Custom Objects

After you package the custom object files in a .zip file, you import the .zip file into the Page Layout Designer.

To install a custom object

1. On the Page Layout Designer toolbar, click the **Install Custom Object** icon.
2. Navigate to the directory that contains the .zip file.
3. Select the file and click **Open**.

   If the custom object is valid, then the **Install Custom Object** dialog box indicates that the install succeeded.
4. Click **OK**.

   The custom object appears in the **Custom Objects** palette and at the bottom of the **All Objects** palette. You can now use the custom object in Page Layouts. In addition, you can access the object's properties in the Page Flow Designer.

8.5 Removing Custom Objects

You can remove a custom object from the Page Layout Designer.

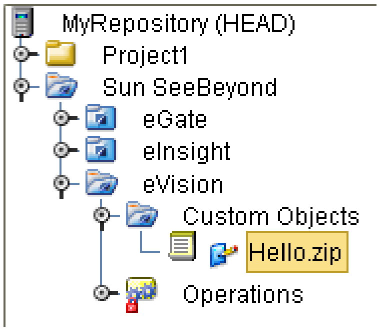
If you open a Page Layout that includes a custom object that has been removed from the Page Layout Designer, then the Page Layout will not open correctly.

To remove a custom object

1. In the Project Explorer of Enterprise Designer, open the **Sun SeeBeyond** folder, the **eVision** folder, and then the **Custom Objects** folder.

   The .zip file appears.

Figure 112 Custom Object .zip File in Project Explorer



2. Right-click the .zip file and then click **Delete**.
3. When prompted to confirm this action, click **OK**.

   The custom object no longer appears.

8.6 Sample Custom Objects

After you upload the **eVisionDocs.sar** file, the **Documentation** page of the Java Composite Application Platform Suite Installer enables you to access the sample projects file.

The sample projects file includes sample files for the custom object. See Table 26.

Table 26 Sample Files for the Custom Object

File Name	Description
Hello.zip	Contains a custom object called **Hello**.
customDynamicTable.zip	Contains a custom object called **Custom Dynamic Table**.
CustomDynamicTable.dtd	Contains a sample Document Type Definition (DTD) for using the Custom Dynamic Table object
CustomDynamicTable.xml.~in	Contains sample input data for the Custom Dynamic Table object

"Installing Custom Objects" on page 191 describes how to import the **Hello.zip** and **customDynamicTable.zip** files into the Page Layout Designer.

8.6.1 Hello Object

The **Hello** object creates a simple text box. The XSD input file (**helloIn.xml**) defines an input element called **hello**. The XSD output file (**helloOut.xml**) defines an output element called **value**. The JSP file (**helloJSP.jsp**) creates the text box at runtime.

You can map data into the object from a variety of sources, such as the Sun SeeBeyond eWay™ File Adapter or a Page Link.

The **Default XML Input** property enables you to specify input data that will be used if there is no input data at runtime. As a test, set the value of the property to the following XML:

```
<hello>enter value</hello>
```

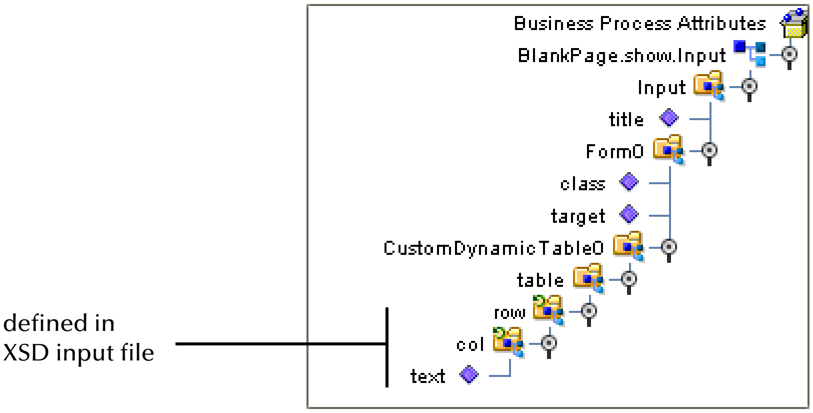
When you click the **Preview** icon in the Page Layout Designer toolbar, the string **enter value** appears in the text box.

8.6.2 Custom Dynamic Table Object

The **Custom Dynamic Table** object creates a table that has a varying number of rows and columns, depending on the input data.

The XSD input file (**customDynamicTableXSDInput.xml**) defines an input element called **row**, which contains a subnode called **col**.

Figure 113 Input Elements in Page Flow Designer



The object does not include an XSD output file, because mapping output from the table is not necessary.

The JSP file (**customDynamicTableJSPSource.jsp**) creates the table at runtime.

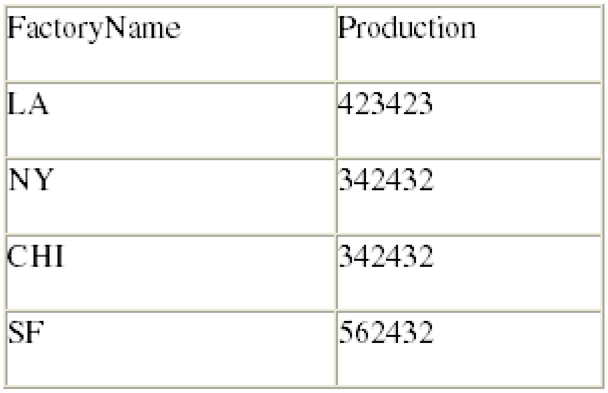
Deploying the Custom Dynamic Table Object

You can deploy the **Custom Dynamic Table** object by using the **CustomDynamicTable.dtd** and **CustomDynamicTable.xml.~in** files, which are included in the sample projects .zip file.

To deploy the Custom Dynamic Table object

1. Create a DTD-based Object Type Definition (OTD) based on the **CustomDynamicTable.dtd** file.
2. Add the custom object to a Page Layout.
3. In the Page Flow, unmarshal the OTD and map the **ColumnData** node in the Output pane to the **text** node in the Input pane.
4. When you deploy the Project, use the **CustomDynamicTable.xml.~in** file as sample input data. The default version of this file produces a table with two columns and five rows (including the header row) at runtime.

Figure 114 Dynamic Table at Runtime



FactoryName	Production
LA	423423
NY	342432
CHI	342432
SF	562432

Modifying the Input File

To test the dynamic capability of the object, modify the **CustomDynamicTable.xml.~in** file so that a new column appears at runtime. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CustomDynamicTable SYSTEM ".\CustomDynamicTable.dtd">

<CustomDynamicTable>

    <Row>
        <ColumnData>FactoryName</ColumnData>
        <ColumnData>Production</ColumnData>
        <ColumnData>Production</ColumnData>
    </Row>

    <Row>
        <ColumnData>LA</ColumnData>
        <ColumnData>423423</ColumnData>
        <ColumnData>534534</ColumnData>
    </Row>

    <Row>
        <ColumnData>NY</ColumnData>
        <ColumnData>342432</ColumnData>
        <ColumnData>444555</ColumnData>
    </Row>

    <Row>
        <ColumnData>CHI</ColumnData>
        <ColumnData>342432</ColumnData>
        <ColumnData>434343</ColumnData>
    </Row>

    <Row>
        <ColumnData>SF</ColumnData>
        <ColumnData>562432</ColumnData>
        <ColumnData>451321</ColumnData>
    </Row>

</CustomDynamicTable>
```

Chapter 9

Generating Portlets

You can create portlets that are compliant with Java™ Specification Request (JSR) 168.

What's in This Chapter

9.1 Portals and Portlets

A *portal* is a web site that serves as a gateway for web-based services and applications.

A *portlet* is a Java-based web component that runs inside a portal.

JSR 168 is a standard that ensures interoperability between portals and portlets.

You can generate a JSR 168-compliant portlet in eVision Studio, and then expose the portlet in a JSR 168-compliant portal such as Sun Java™ System Portal Server.

9.2 Creating and Assigning Preferences

JSR 168 includes the concept of *preferences*, which enable a portal to be customized for different users.

Preferences contain one or more name-value pairs. For example, the name in a name-value pair could be **StockSymbol**, while the value could specify the company that the end user is interested in.

9.2.1 Creating a Preferences Object

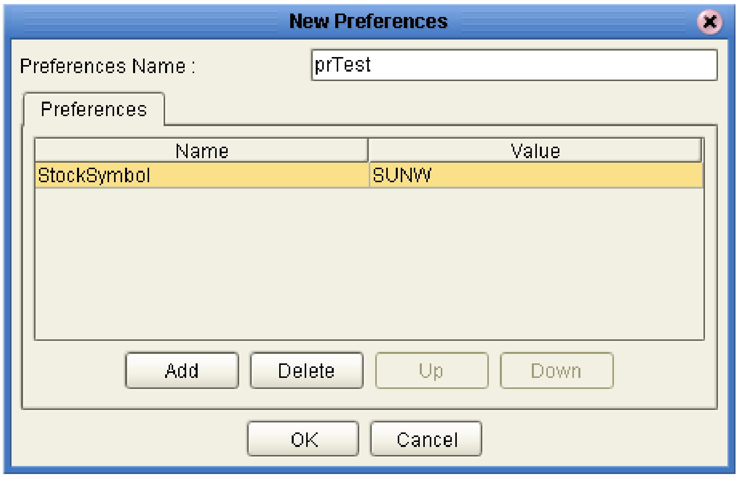
You create preferences in the Page Layout Designer.

To create a Preferences object

1. In the Project Explorer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page Preferences**.

   The **New Preferences** dialog box appears.
3. In the **Preferences Name** field, type a name for the Preferences.
4. Add a name/value pair.

   Figure 115 New Preferences Dialog Box

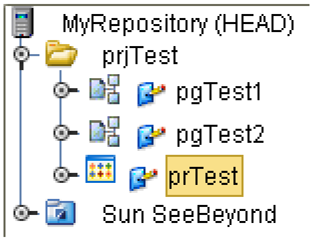
   Preferences Name : prTest

Name	Value
StockSymbol	SUNW

   At runtime, the value is used if the user has not specified a default value for the preference.
5. To add another name/value pair, click **Add** and enter the name and value. To delete a name/value pair, select the name/value pair and click **Delete**. To move a name/value pair up one level in the stack, select the name/value pair and click **Up**. To move a name/value pair down one level in the stack, select the name/value pair and click **Down**.
6. Click **OK**.

   The Preferences object is added to the Project Explorer.

   Figure 116 Preferences Object in the Project Explorer

9.2.2 Assigning Preferences

After you create a Preference object, you can assign the object to a Page Layout or a Page Link.

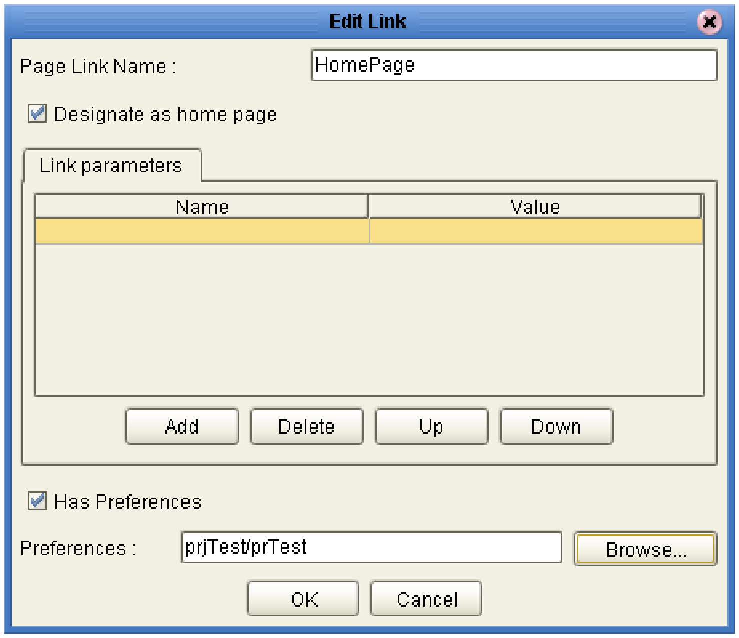
To assign preferences to a Page Layout

1. In the **Properties** tab, select the **Page** property type.
2. In the left column, select the **Preferences** property.
3. In the right column, set the value to the Preferences object.

To assign preferences to a Page Link

1. When creating or modifying the Page Link, do the following:
   - A Select the **Has Preferences** check box.
   - B Click **Browse**.
   - C Select the Preferences object.
   - D Click **Open**.

Figure 117 Assigning Preferences to a Page Link



Page Link Name : HomePage

☑ Designate as home page

☑ Has Preferences

Preferences : prjTest/prTest

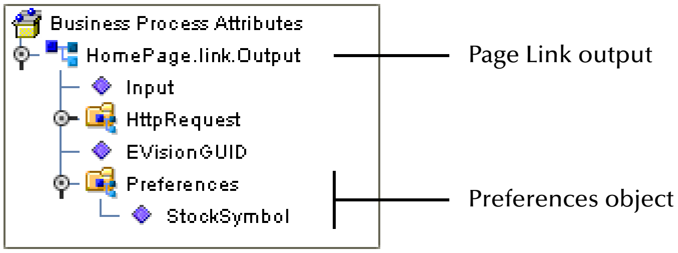
2. Click **OK**.

9.3 Mapping Preferences Data

When you assign preferences to a Page Link or Page Layout, you can access the preferences in the Business Rule Designer of the Page Flow Designer.

The name portion of each name/value pair appears as a node. Figure 118 shows the output of a Page Link that has a Preferences object.

Figure 118 Preferences Example in the Business Rule Designer



You can map data into the node. For example, you could obtain a stock price from a database or web service.

9.4 Generating the Portlet Files

The final task in eVision Studio is to generate the JSR 168-compliant portlet. You can use the portlet in any portlet server that supports JSR 168.

To generate the portlet files

1. In the Connectivity Map Editor, open the Service binding box and double-click the connection icon between the **WSPProvider** Implemented Service and the Web Connector.

   The **Properties** dialog box appears.

2. Set the value of the **Is JSR 168** property to **true**.
3. Click **OK**.
4. When you build the Project, the application file is generated in the **Sun_JavaCAPS_install_dir\edesigner\builds** directory.

9.5 Using Sun Java System Portal Server

Sun Java™ System Portal Server delivers capabilities and components necessary for advanced portal solutions. Using Portal Server, you are able to access and find relevant applications and information and personalize your portal environments to best meet your needs.

Portal Server Benefits

Portal Server’s major benefits include:

- Aggregation
- Personalization
- Security, including single sign-on (SSO)
- Collaboration and Communities
- Secure remote access
- Mobile access
- Search
- Content management and delivery
- You can refer to Sun’s Portal Server Web site for information on additional available features at the following URL:

  http://www.sun.com/software/products/portal_srvr/index.xml

  See also **Reference Documentation** on page 199.

These benefits and their related features securely deliver the Portal Server capabilities that you require. The overall result is that you can work more efficiently and flexibly, doing a variety of business-related activities, for example, allowing:

- Employees to collaborate
- Sales personnel to access and update customer information on the road
- Partners to get advance access to the latest product designs

Portal Server and the Java CAPS

You can integrate Portal Server with the Java CAPS, which supports the following Portal Server releases:

- Version 6 2005Q4
- Version 7

Portal Server is part of the Sun Java Enterprise System (Java ES). Use Portal Server Version 6 2005Q4 when deploying on a Windows platform. Use Version 7 for all other platforms.

The rest of this section explains operations you need to perform to allow Portal Server to work seamlessly with eVision Studio applications.

9.5.1 Reference Documentation

This section provides general information on Portal Server features and procedures relevant to the Java CAPS. For more information on Portal Server, Java ES, and Sun Java System Application Server, you can refer to the appropriate Sun Microsystems documentation.

You can use the Documentation Web site’s Search feature, located in the upper right corner of the page, to find the appropriate documentation. This feature allows you to search by a variety of methods, for example, document title, key word, or product name.

Sun Documentation Web Page for Portal Server 6 2005Q4

To access Portal Server 6 2005Q4 and related documentation, see the following Sun Web page:

http://docs.sun.com/app/docs/coll/1293.1

Scroll down the page to find the product documentation.

You can also use this Web page as a convenient list of the Portal Server-related applications and systems. The following information helps you navigate this Web page:

- Because Portal Server operates with Java ES, its documentation is integrated with the suite of user's guides for Java ES.
- This documentation set includes installation instructions for overall Java ES, including all its component products, such as Sun Java System Application Server, and Portal Server.

Sun Documentation Web Page for Portal Server 7

To access Portal Server 7 and related documentation, see the following Sun Web page:

http://docs.sun.com/coll/1303.1

You can also use this Web page as a convenient list of the Portal Server-related applications and systems. The following information helps you navigate this Web page:

- This documentation set includes installation instructions for overall Java ES, including all its component products, such as Sun Java System Application Server, and Portal Server.
- This documentation set includes a technical note on integrating Portal Server with Java CAPS. See **Necessary Patches** on page 201 for details.

9.5.2 Before Installing Portal Server

Before you install Portal Server, ensure that:

- eGate Integrator and eVision Studio (and any related applications, as needed) are installed and operational.
- You have installed the necessary patches for Portal Server and Java CAPS.
- You have created or imported at least one Project for eVision Studio using Enterprise Designer; not necessary but recommended.
- You have read the eVision Studio **Readme.txt** file for information on pre-installation requirements, for example, patches or Portal Server prerequisites.

For complete instructions on how to create and/or import a Project, using Enterprise Designer, see the *eGate Integrator User's Guide*.

Necessary Patches

You must install patches for your system, to ensure the proper operation of Portal Server. These patches are necessary to:

- Allow Portal Server to operate correctly with Java CAPS.
- Allow Java CAPS to operate correctly with the Sun Java System Application Server version 8.1 UR2 (Patch 8 EE).

The patch you need depends on your operating system. You can access the patches directly from SunSolve Online. The URL for SunSolve is:

http://www.sunsolve.sun.com

For example, at the time this document was being readied for publication, the following patches were current:

- **For Portal Server 7:**
  - Solaris-SPARC — 121913-01
  - Solaris-i386 — 121914-01
  - Linux — 121915-01
- **For Sun Java System Application Server 8.1:**
  - Solaris-SPARC — 119166-15 (package-based); or 119169-07 (file-based)
  - Solaris-x86 — 119167-15 (package-based); or 119170-07 (file-based)
  - Linux — 119168-15 (package-based); or 119171-07 (file-based
  - Windows XP — 122848-01 (package-based); or 119172-07 (file-based)

Because patches are frequently revised or superseded, do not install any patch until you have checked that it is the latest available.

You can find specific information on locating and accessing these patches as follows:

- **Portal Server**: See the Portal Server technical note “Integrating Java CAPS With Sun Java System Portal Server.” This technical note provides the patch numbers you need to search for these patches on SunSolve. You can find this technical note at the following URL:

  http://docs.sun.com/source/819-6117/index.html

- **Sun Java System Application Server**: See the main Java CAPS **Readme.txt** file, which provides the patch numbers you need to search for these patches on SunSolve.

Portal Server 6 2005Q4 and Portal Server 7

Regardless of whether you install Portal Server 6 2005Q4 or Portal Server 7, all other related systems and applications are compatible with both Portal Server versions (see the list on the Web page cited under **Sun Documentation Web Page for Portal Server 6 2005Q4** on page 200).

Release Notes and Additional Information

Before you begin installing Portal Server for use with Java CAPS, be sure to read the release notes that accompany the documentation. For general information on how to obtain Portal Server documentation, see **Reference Documentation** on page 199.

It is especially recommended that you read the following documents:

- "Integrating Java CAPS With Sun Java System Portal Server" (see the URL for the Portal Server technical note referenced under **Sun Documentation Web Page for Portal Server 7** on page 200).
- "Sun Java System Portal Server Release Notes for Microsoft Windows, Version 6 2005Q4" at the following URL:

  http://docs.sun.com/source/819-4270/index.html
- *Sun Java System Portal Server 7 Installation Guide*, "Chapter 1: Pre Installation Requirements for the Sun Java System Portal Server 7 Software" at the following URL:

  http://docs.sun.com/app/docs/doc/819-3027

9.5.3 Running eVision Studio Applications with Portal Server 6 2005Q4

This section explains how to set up Sun Java System Portal Server 6 2005Q4 to run with the Java CAPS applications, including eVision Studio.

Before You Start

Take care to ensure you have completed the prerequisite actions listed under **Before Installing Portal Server** on page 200.

System Requirements

You must run Portal Server 6 2005Q4 in:

- Sun Java System Application Server, Version 8.1

If you want to utilize a Windows XP operating system, use this version of Java ES and Portal Server and display your Web interfaces (frames) on eVision Studio using an IFrame channel.

IFrame channels can reference any URL accessible to your Sun Java System Application Server installation. As a result, eVision Studio Projects displayed by IFrame do not have to be deployed on the same server as Portal Server.

For more information on supported operating systems, see the **Readme.txt** file that accompanies eVision Studio.

Supported Operating Systems

For specific information on all operating systems supported by Portal Server 6 2005Q4 in conjunction with the Java CAPS and eVision Studio, see the **Readme.txt** file for eVision Studio.

Portal Server 6 2005Q4 on Windows: Procedural Overview

This section explains procedures you need to know, to successfully set up eVision Studio applications to display on Portal Server 6 2005Q4, utilizing IFrames on Windows.

The following list provides an overview of these procedures:

- Configuring the Sun Java System Application Server and Java ES so that Portal Server deploys on the correct application server.
- Deploying eVision Studio applications to the Sun Java System Application Server.
- Creating users in the Sun Java System Application Server using Access Manager.

Note: *For more information on Access Manager, see the following URL:* http://docs.sun.com/app/docs/coll/1292.1

- Creating new IFrame channels using Access Manager.
- Displaying the Portal Desktop.

Important

The port numbers provided in all access URL examples are the system defaults. The actual port numbers used by your system's configuration may be different. Before you begin using the procedures provided in the rest of this section, make sure you have verified the correct port number for any URL referenced by a given procedure.

Installing Portal Server 6 2005Q4 to Run in Sun Java System Application Server on Windows

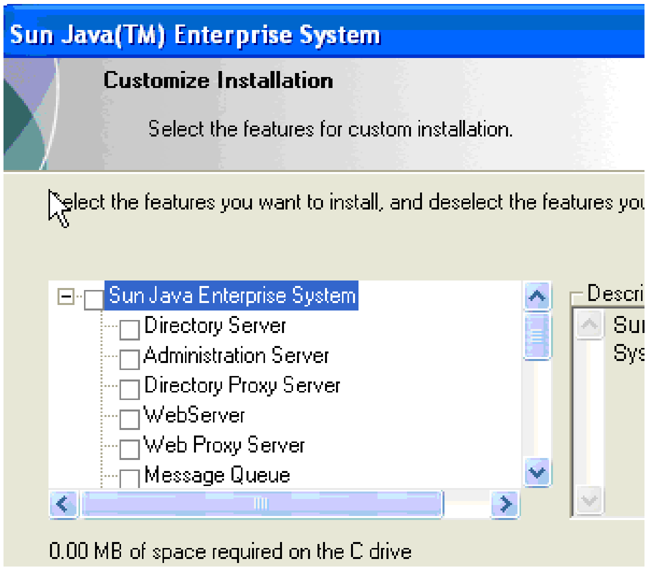
Portal Server 6 2005Q4 uses the Java ES installer and, by default, is configured to run in the Java ES Web Server. You must change this default configuration.

To install

1. Start the Java ES installer.
2. Choose **Configure Automatically**.
3. Choose the **Custom Installation** option.

   The Customize Installation user interface appears. See Figure 119.

Figure 119 Customize Installation User Interface



Sun Java(TM) Enterprise System

Customize Installation

Select the features for custom installation.

Select the features you want to install, and deselect the features you

0.00 MB of space required on the C drive

4 Deselect **Java ES**. See Figure 119.

5 Select **Sun Java System Application Server**, **Directory Server**, **Access Manager**, and **Portal Server** from the list shown in Figure 119.

Automatically, several other items are selected, including the Java ES Web server, which you do not want installed.

6 Expand **Sun Java System Application Server**; all the boxes are checked by default.

7 Deselect **Load Balancer Plugin**.

8 Make sure that **Web Server** is not selected.

9 Click **Next** to continue.

10 Finish the installation procedure. You are prompted appropriately as other components are needed.

Important

During the installation, you are asked to create and enter an administrator user name and password. Make sure you keep a record of these exact entries.

Starting Sun Java System Application Server on Windows

This section explains how to start Sun Java System Application Server and the Node Agent on Windows.

To Start Sun Java System Application Server and the Node Agent

1 On the Windows **Start > All Programs** menu, choose **Sun Microsystems > Application Server > Start Administration Server**.

A command line window displays, and you are prompted to enter your password.

Whenever you are prompted for your administrator user name and/or password, be sure to use the ones you created during your initial Java ES Portal Server installation.

2 After you enter the password, click **Enter**.

There is no confirmation the password is accepted. Several minutes may pass before the server finishes starting.

3 On the Windows **Start > All Programs** menu, choose **Sun Microsystems > Application Server > Start Node Agent**.

A command line window displays, and you are prompted to enter your password.

4 After you enter the password, click **Enter**.

Again, there is no confirmation the password is accepted. Several minutes may pass before the node agent finishes starting.

Deploying eVision Studio Applications

This section explains how to deploy eVision Studio applications on Sun Java System Application Server.

To deploy eVision Studio applications

1 Make sure the Sun Java System Application Server is running.

2 Java CAPS applications require special permissions. Make sure the appropriate permissions have been set up correctly in the **server.policy** file. For more information on permissions in Java CAPS, see the *eGate Integrator System Administration User's Guide*.

3 Deploy the eVision Studio **.ear** file to the application server on which Portal Server is running. You can do this operation using:

- Enterprise Designer
- Enterprise Manager
- A Sun Java System Application Server's administrator interface, either the Admin Console or the command line

A command-line example follows:

```
./asadmin deploy --user admin /tmp/prj2dp8.ear
```

See the *eGate Integrator User's Guide* and *eGate Integrator System Administration Guide* for details on how to deploy an **.ear** file using Enterprise Designer and Enterprise Manager. See the **procedure on page 206** for information on how to deploy an **.ear** file using Sun Java System Admin Console.

To access the Application Server Admin Console

1 On the Windows **Start > All Programs** menu, choose **Sun Microsystems > Application Server > Admin Console**.

You can also access this page by entering either of the following URLs, using your Web browser:

- On any machine:

```
https://<host name>:4850/
```

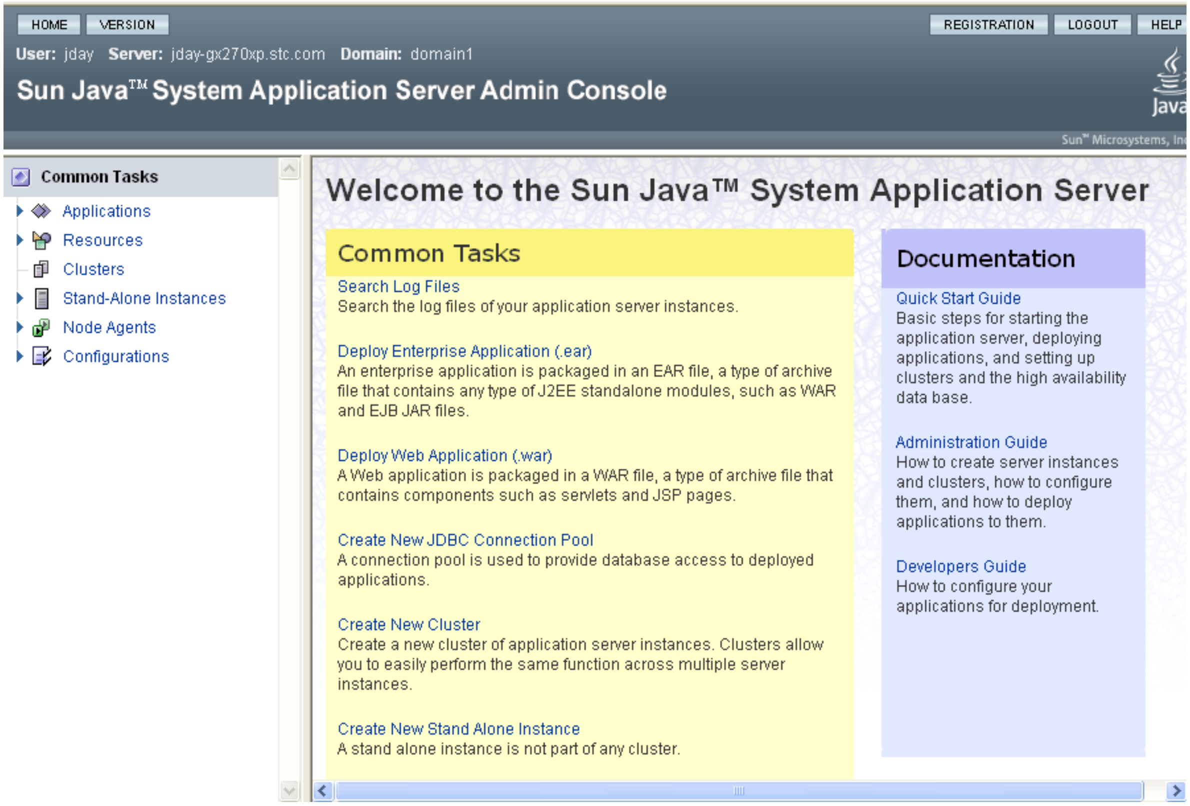
- On your own machine:

```
https://localhost:4850/
```

2 In the resulting dialog box, enter your administrator user name and password, and click **Enter**.

The Admin Console page displays. See Figure 120.

Figure 120 Admin Console Page



User: jday **Server:** jday-gx270xp.stc.com **Domain:** domain1

Sun Java™ System Application Server Admin Console

Welcome to the Sun Java™ System Application Server

Common Tasks

Search Log Files
Search the log files of your application server instances.

Deploy Enterprise Application (.ear)
An enterprise application is packaged in an EAR file, a type of archive file that contains any type of J2EE standalone modules, such as WAR and EJB JAR files.

Deploy Web Application (.war)
A Web application is packaged in a WAR file, a type of archive file that contains components such as servlets and JSP pages.

Create New JDBC Connection Pool
A connection pool is used to provide database access to deployed applications.

Create New Cluster
Create a new cluster of application server instances. Clusters allow you to easily perform the same function across multiple server instances.

Create New Stand Alone Instance
A stand alone instance is not part of any cluster.

Documentation

Quick Start Guide
Basic steps for starting the application server, deploying applications, and setting up clusters and the high availability data base.

Administration Guide
How to create server instances and clusters, how to configure them, and how to deploy applications to them.

Developers Guide
How to configure your applications for deployment.

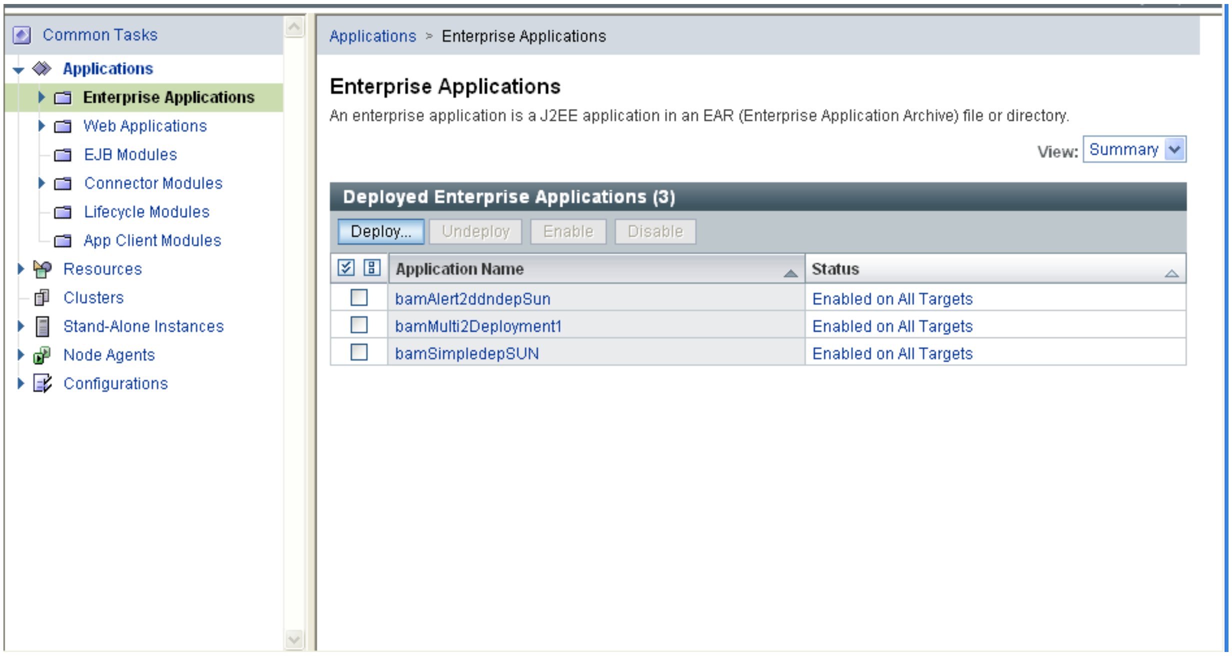
To deploy eVision Studio applications using Admin Console

1 On the **Common Tasks** frame (on the left), select **Applications** in the tree, to show the available operations under this option.

2 Select **Enterprise Applications** under **Applications**.

The **Enterprise Applications** frame appears on the right. See Figure 121.

Figure 121 Admin Console Page With Enterprise Applications Frame



Enterprise Applications

An enterprise application is a J2EE application in an EAR (Enterprise Application Archive) file or directory.

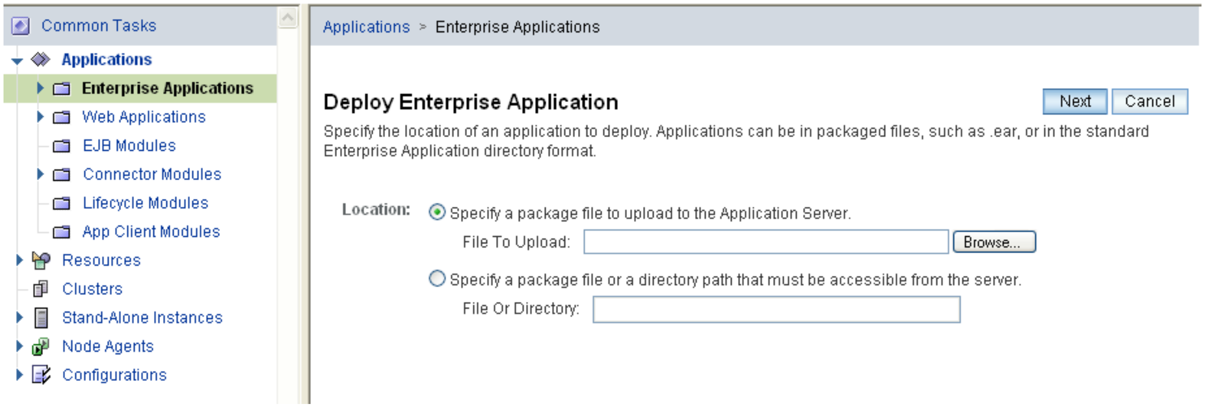
Deployed Enterprise Applications (3)

	Application Name	Status
□	bamAlert2ddndepSun	Enabled on All Targets
□	bamMulti2Deployment1	Enabled on All Targets
□	bamSimpledepSUN	Enabled on All Targets

3 Click **Deploy**.

The **Deploy Enterprise Application** frame appears. See Figure 122.

Figure 122 Deploy Enterprise Application Frame: Initial



Deploy Enterprise Application

Specify the location of an application to deploy. Applications can be in packaged files, such as .ear, or in the standard Enterprise Application directory format.

Location: ● Specify a package file to upload to the Application Server.

File To Upload:

○ Specify a package file or a directory path that must be accessible from the server.

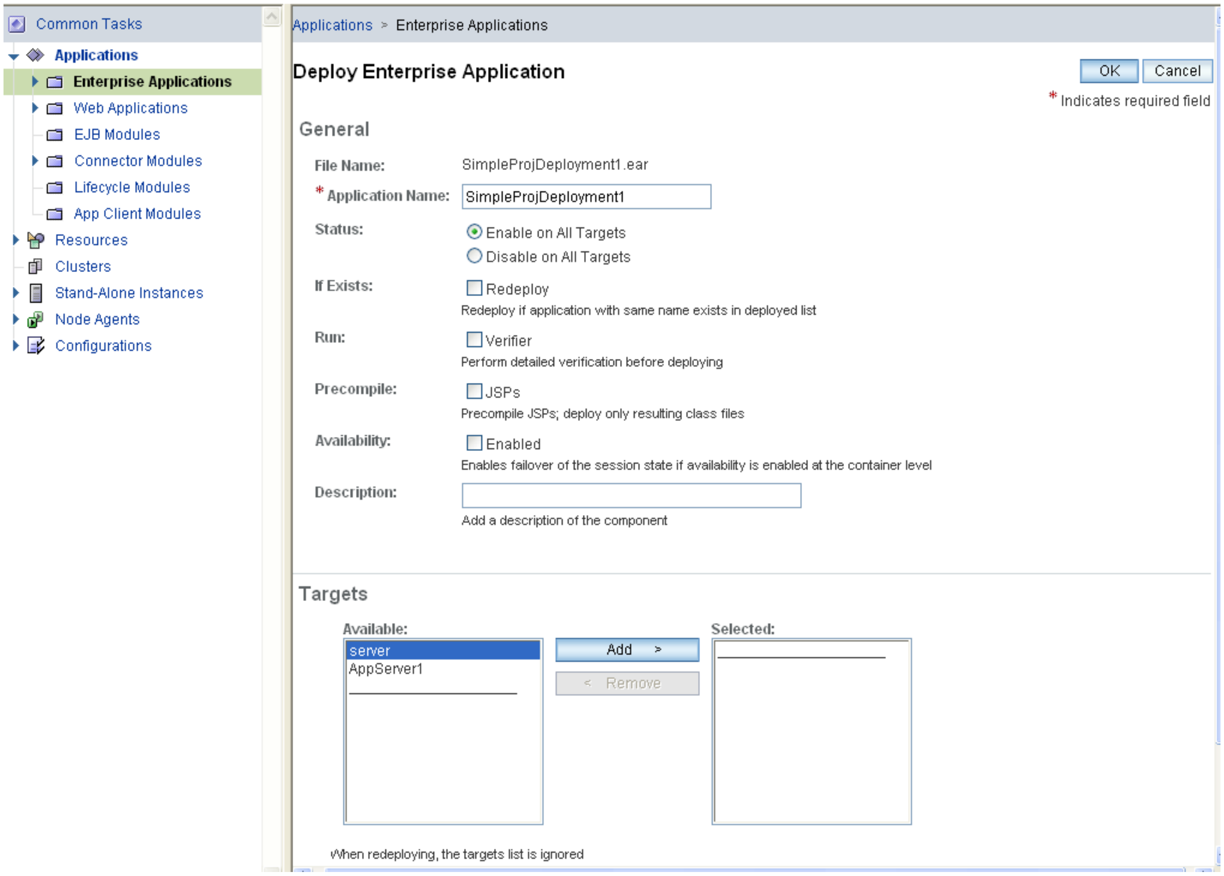
File Or Directory:

4 Browse to the **.ear** file you generated using Enterprise Designer.

5 Select the file and click **Next**.

The **Deploy Enterprise Application** frame refreshes to display another page, as shown in Figure 123.

Figure 123 Deploy Enterprise Application Frame: After .ear File Selection



Deploy Enterprise Application

* Indicates required field

General

File Name: SimpleProjDeployment1.ear

* Application Name: SimpleProjDeployment1

Status: ● Enable on All Targets
○ Disable on All Targets

If Exists: □ Redeploy
Redeploy if application with same name exists in deployed list

Run: □ Verifier
Perform detailed verification before deploying

Precompile: □ JSPs
Precompile JSPs; deploy only resulting class files

Availability: □ Enabled
Enables failover of the session state if availability is enabled at the container level

Description:
Add a description of the component

Targets

Available: server, AppServer1

Selected:

When redeploying, the targets list is ignored

6 Scroll to the bottom of this frame, to **Targets**, if this section is not already displayed.

7 Under **Available**, click **server** then **Add** to move your selected file to the **Selected** column on the right.

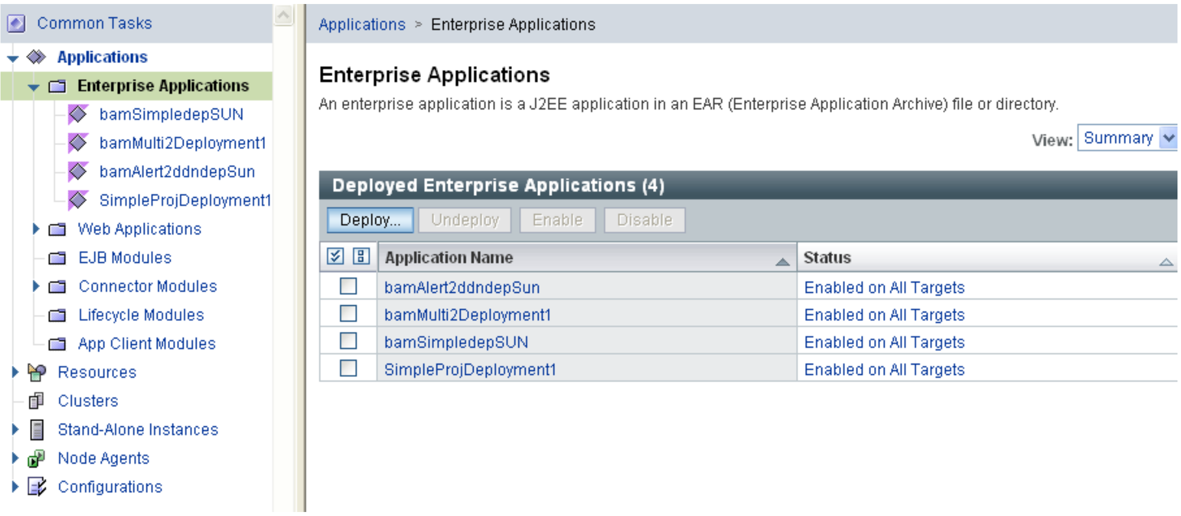
Note: *If you have deployed this application before, select the* **Redeploy** *check box.*

8 Click **OK**.

You have now finished deploying the current application. The **Deploy Enterprise Application** frame refreshes to display the initial frame, **Enterprise Applications**.

The name of your application appears in the list of deployed applications on this frame. See Figure 124.

Figure 124 New Enterprise Applications List



Enterprise Applications

An enterprise application is a J2EE application in an EAR (Enterprise Application Archive) file or directory.

Deployed Enterprise Applications (4)

☑	Application Name	Status
□	bamAlert2ddndepSun	Enabled on All Targets
□	bamMulti2Deployment1	Enabled on All Targets
□	bamSimpledepSUN	Enabled on All Targets
□	SimpleProjDeployment1	Enabled on All Targets

Creating New Portal Server Users

Use Sun Java System Access Manager to create a new Portal Server user.

To view the Access Manager Console page

1 Perform either of the following operations:

- On the Windows **Start > All Programs** menu, choose **Sun Microsystems > Access Manager > Administration Console**.
- Enter the following URL on your Web browser:

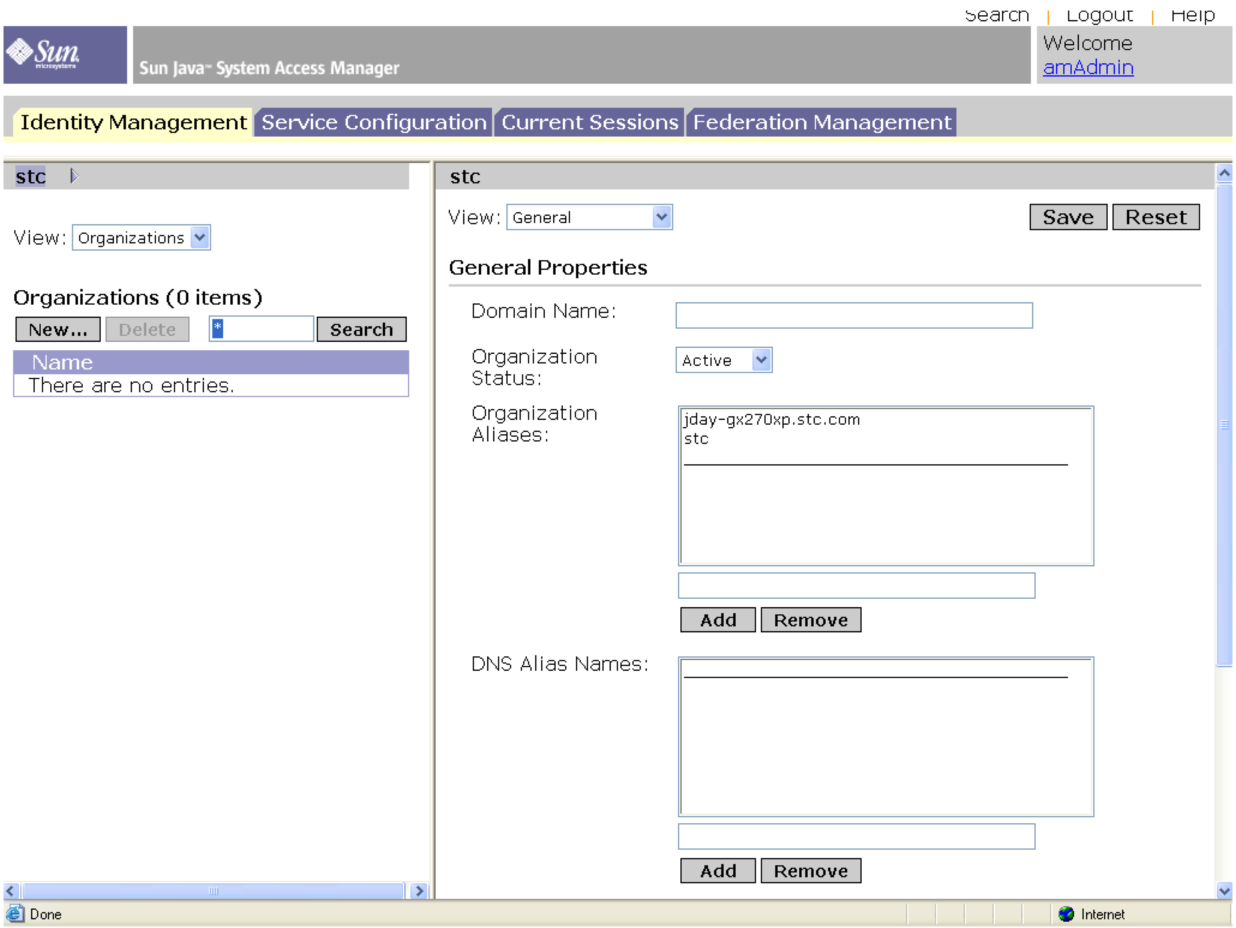
  `http://`*<host name>*`:38080/amconsole`

2 Use the default user name **amadmin**, and enter your administrator password.

3 Click **Enter**.

The Access Manager Console page appears. See Figure 125.

Figure 125 Access Manager Console Page



View: Organizations

Organizations (0 items)

There are no entries.

View: General

Organization Status: Active

Organization Aliases:
jday-gx270xp.stc.com
stc

To create a new user

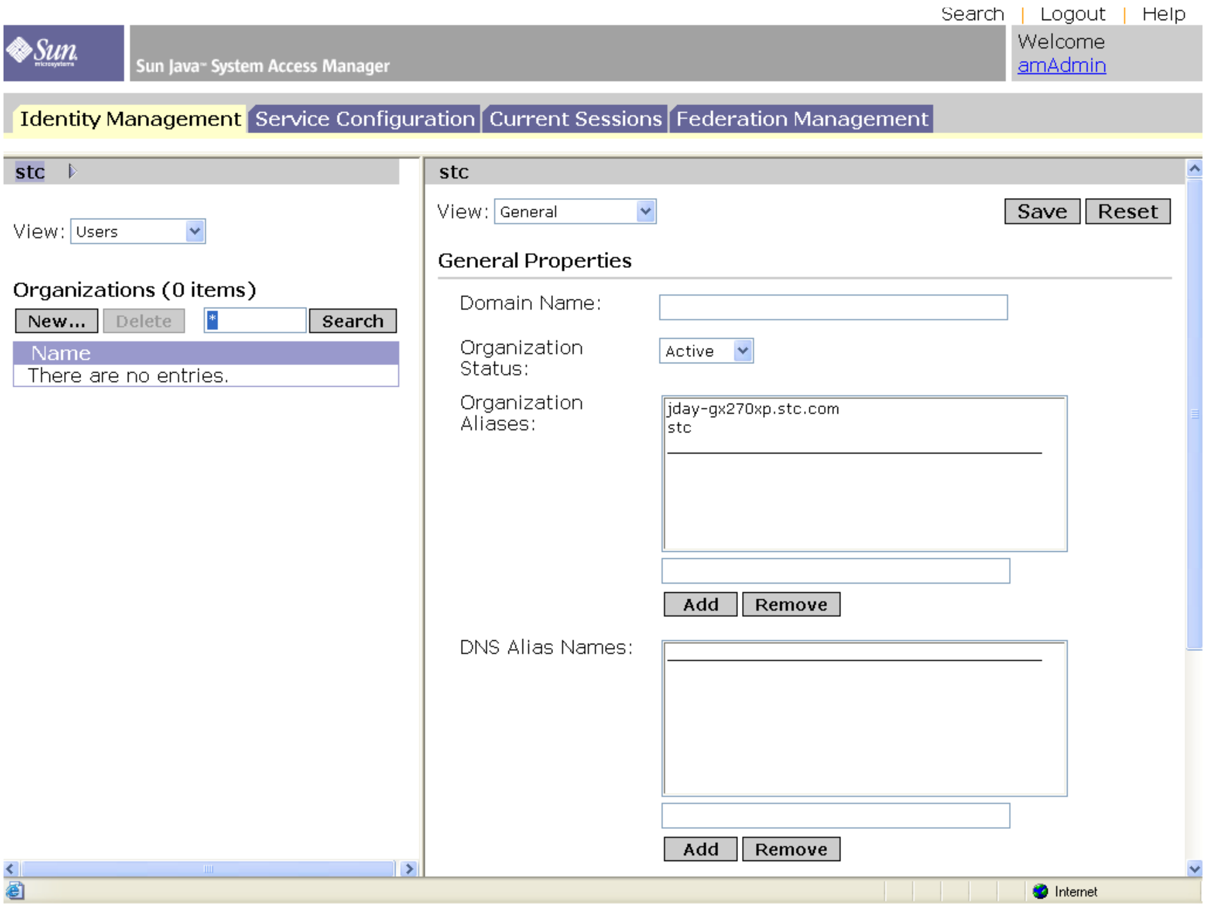
1. On the Access Manager Console page, make sure the **Identity Management** tab content is displayed.

   If there are multiple organizations displayed, choose the organization that has the appropriate permissions for you to use the Portal Desktop.

2. From the **View** drop-down list choose **Users**.

   A frame appears on the right, allowing you to edit user properties, as well as create new users. See Figure 126.

Figure 126 Access Manager Console With User General Properties



stc

View: Users

Organizations (0 items)

There are no entries.

stc

View: General

General Properties

Domain Name:

Organization Status: Active

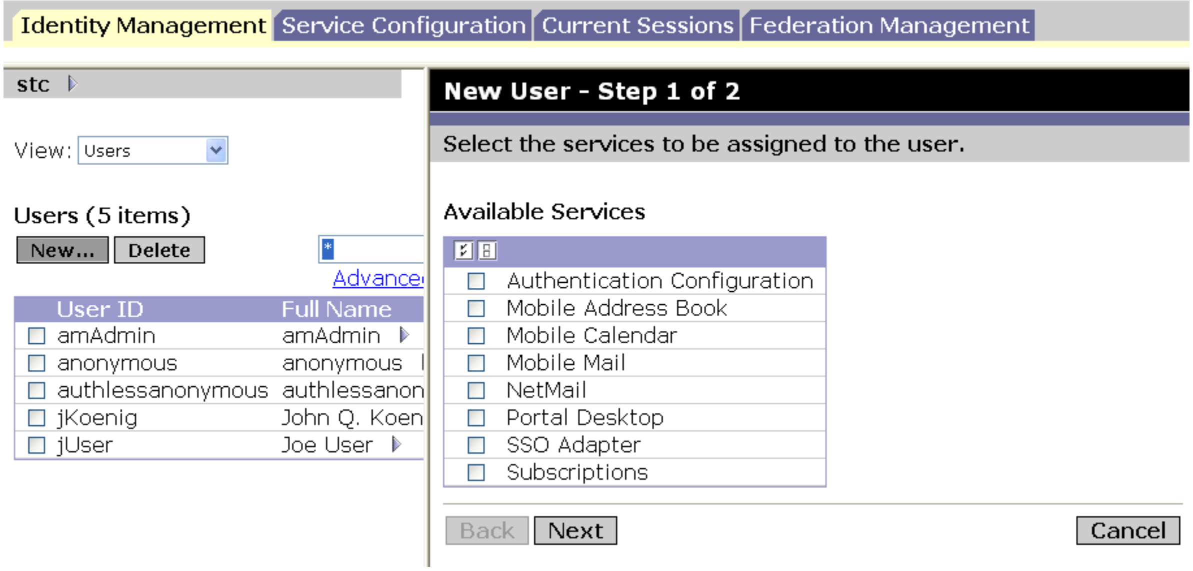
Organization Aliases:
jday-gx270xp.stc.com
stc

DNS Alias Names:

3 Click **New**.

A frame appears, allowing you to select the appropriate services for the new user. The right frame also displays a list of users. See Figure 127.

Figure 127 Available Services for New User



Users (5 items)

User ID	Full Name
□ amAdmin	amAdmin
□ anonymous	anonymous
□ authlessanonymous	authlessanon
□ jKoenig	John Q. Koen
□ jUser	Joe User

New User - Step 1 of 2

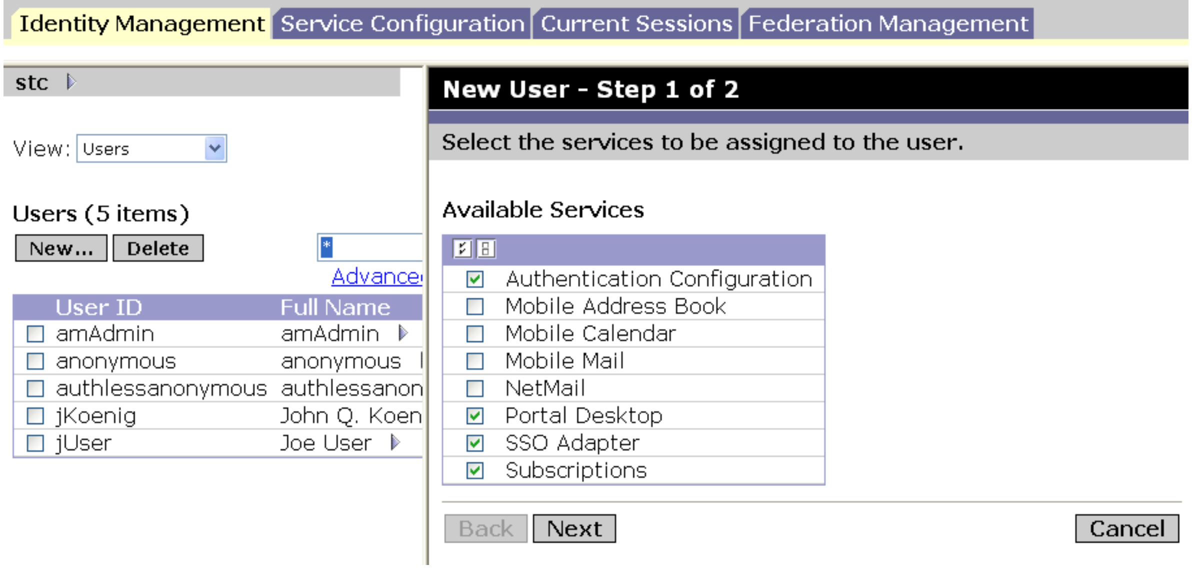
Select the services to be assigned to the user.

Available Services

- □ Authentication Configuration
- □ Mobile Address Book
- □ Mobile Calendar
- □ Mobile Mail
- □ NetMail
- □ Portal Desktop
- □ SSO Adapter
- □ Subscriptions

4 Select the services you want. Be sure to include **Portal Desktop**. For an example, see Figure 128.

Figure 128 Available Services for New User List: Selected



Users (5 items)

User ID	Full Name
□ amAdmin	amAdmin
□ anonymous	anonymous
□ authlessanonymous	authlessanon
□ jKoenig	John Q. Koen
□ jUser	Joe User

New User - Step 1 of 2

Select the services to be assigned to the user.

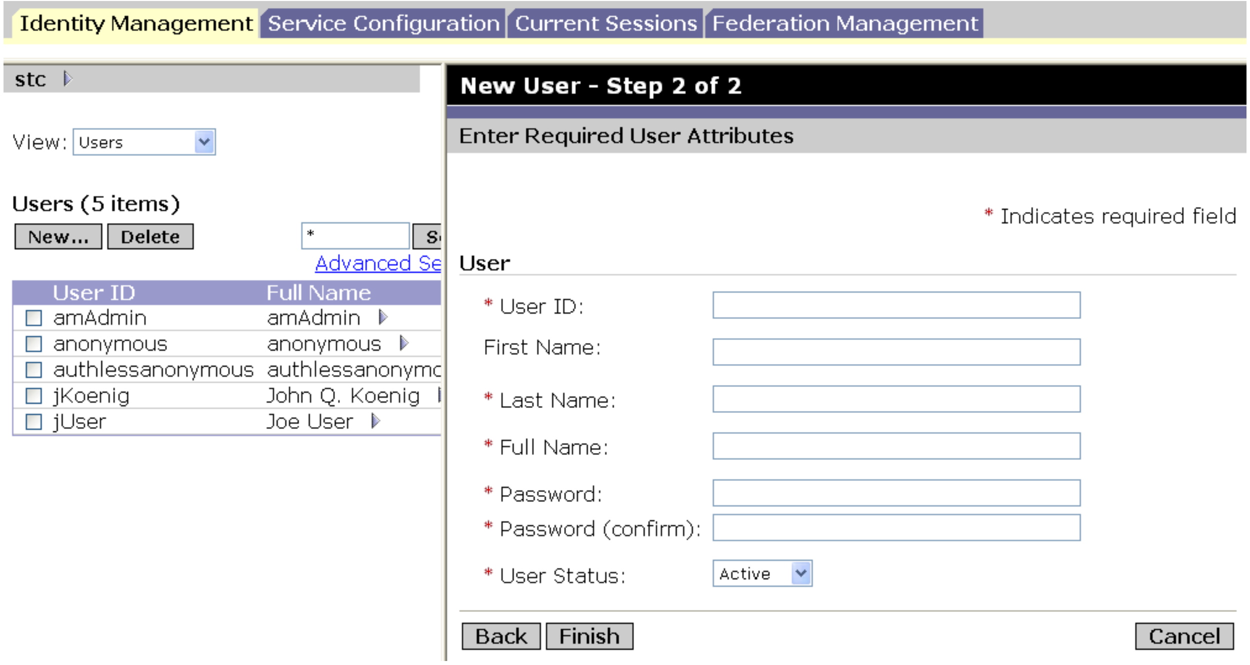
Available Services

- ☑ Authentication Configuration
- □ Mobile Address Book
- □ Mobile Calendar
- □ Mobile Mail
- □ NetMail
- ☑ Portal Desktop
- ☑ SSO Adapter
- ☑ Subscriptions

5 Click **Next**.

A frame appears, allowing you to enter new user information. See Figure 129.

Figure 129 Information for New User List



View: Users

* User Status: Active

	User ID	Full Name
□	amAdmin	amAdmin
□	anonymous	anonymous
□	authlessanonymous	authlessanonymc
□	jKoenig	John Q. Koenig
□	jUser	Joe User

6 Enter the necessary user information and click **Finish**.

You are finished creating the new user. The list of users refreshes and the Access Manager Console page now shows the new user.

Creating a New Channel

Use Access Manager Console to create a new channel.

To view the Access Manager Console page

- Use the **procedure on page 209**.

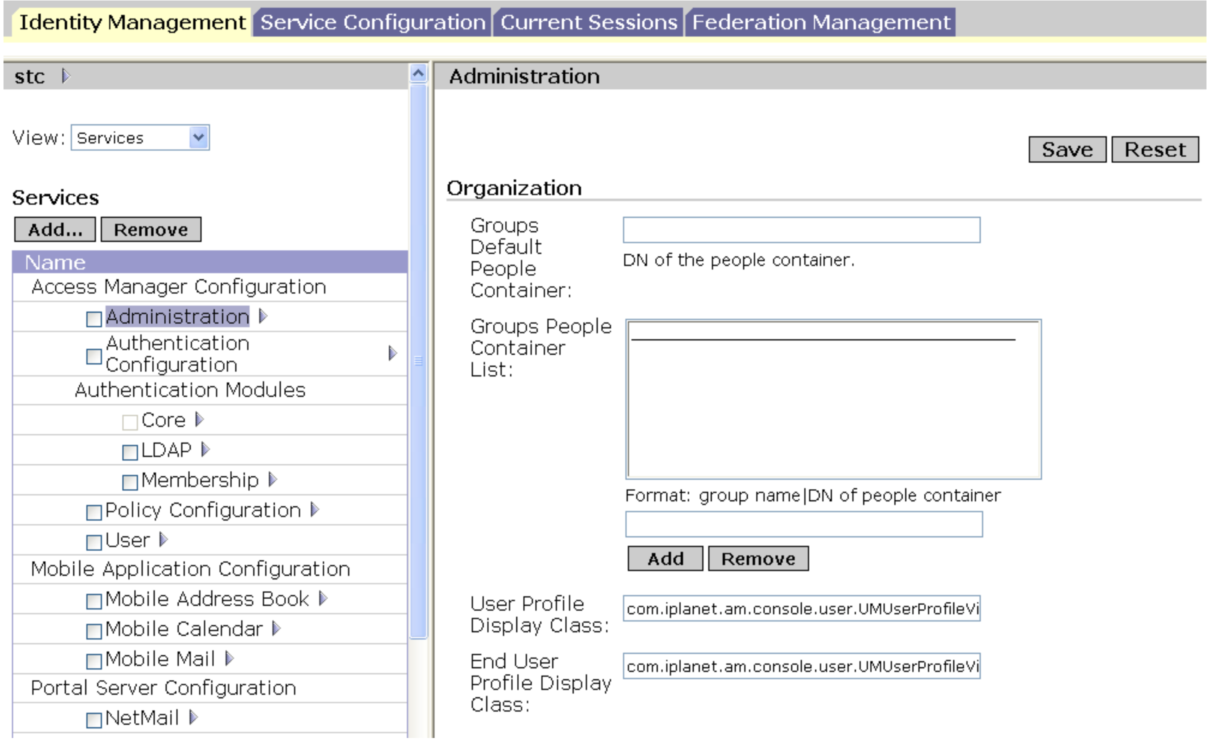
To create a new channel

1 On the Access Manager page, make sure the **Identity Management** tab content is displayed.

2 From the **View** drop-down list choose **Services**.

A frame appears on the right, allowing you to manage Portal Server services, under Identity Management. See Figure 130.

Figure 130 Initial Window: Portal Server Services (Identity Management)



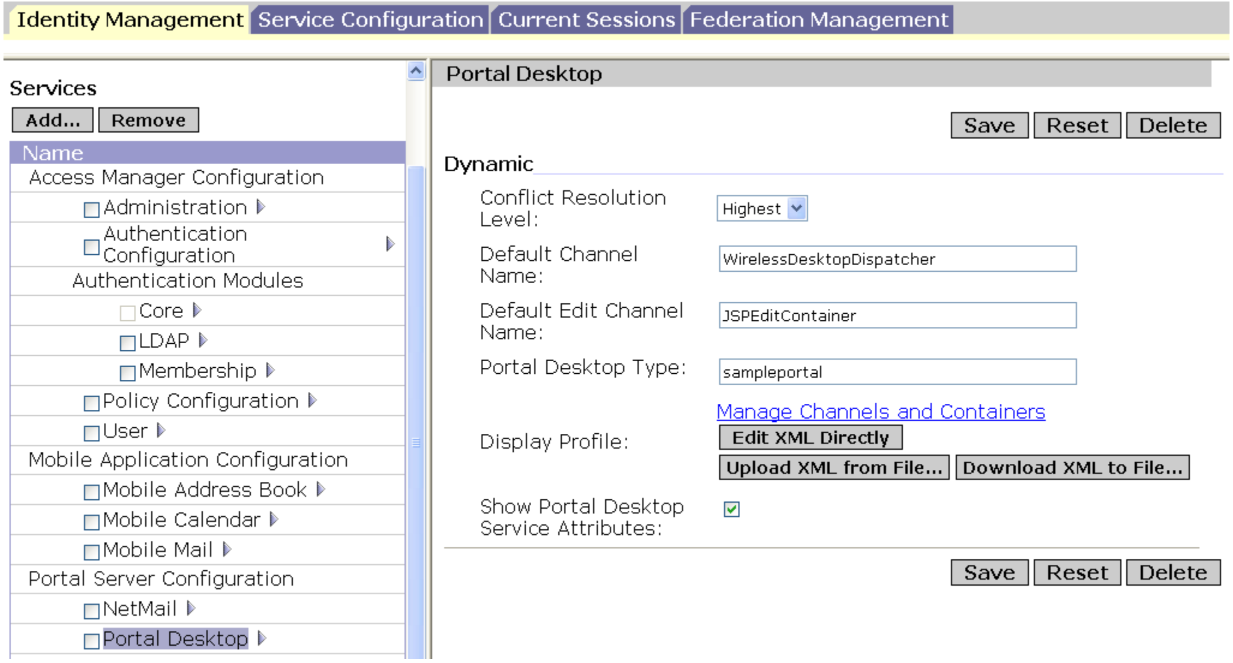
User Profile Display Class: com.iplanet.am.console.user.UMUserProfileVi

End User Profile Display Class: com.iplanet.am.console.user.UMUserProfileVi

3 Scroll down the **Services** (left) frame to display the **Portal Desktop** options and click the pointer icon to the right of **Portal Desktop**.

The **Portal Desktop** frame appears on the right. See Figure 131.

Figure 131 Access Manager Page With Portal Desktop



Identity Management | Service Configuration | Current Sessions | Federation Management

Services

Add... | Remove

Name
- Access Manager Configuration
  - □ Administration ▸
  - □ Authentication Configuration ▸
  - Authentication Modules
    - □ Core ▸
    - □ LDAP ▸
    - □ Membership ▸
  - □ Policy Configuration ▸
  - □ User ▸
- Mobile Application Configuration
  - □ Mobile Address Book ▸
  - □ Mobile Calendar ▸
  - □ Mobile Mail ▸
- Portal Server Configuration
  - □ NetMail ▸
  - □ Portal Desktop ▸

Portal Desktop

Save | Reset | Delete

Dynamic

Conflict Resolution Level: Highest

Default Channel Name: WirelessDesktopDispatcher

Default Edit Channel Name: JSPEditContainer

Portal Desktop Type: sampleportal

Display Profile: Manage Channels and Containers | Edit XML Directly | Upload XML from File... | Download XML to File...

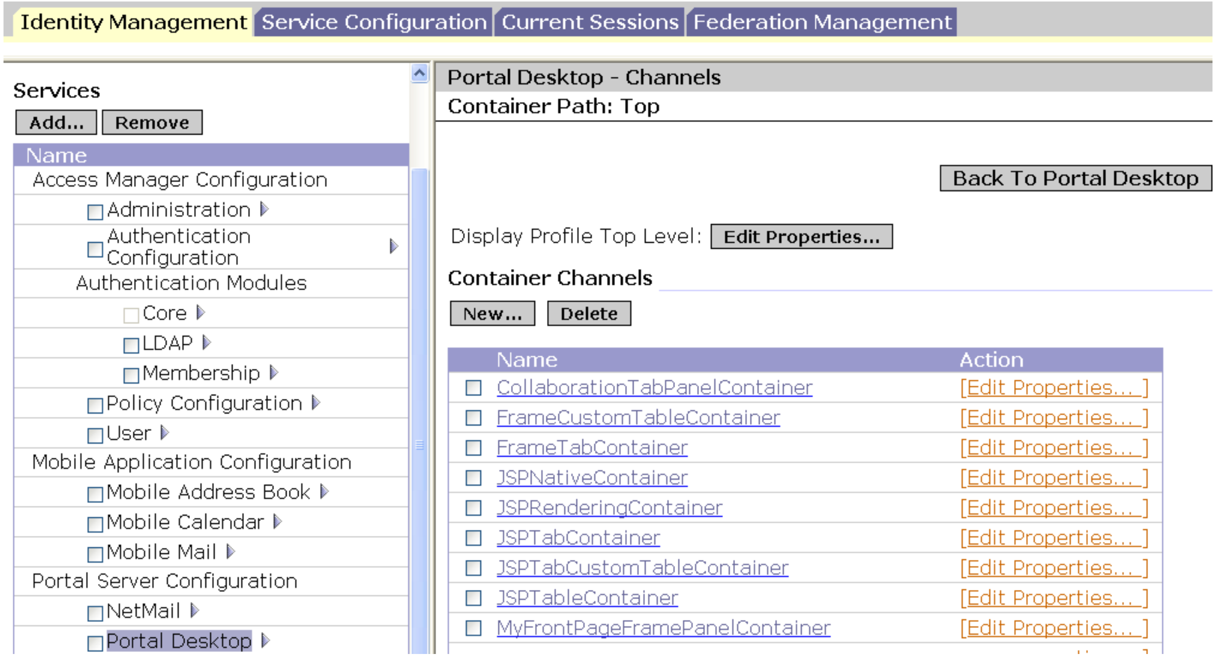
Show Portal Desktop Service Attributes: ☑

Save | Reset | Delete

4 On the **Portal Desktop** frame, click the **Manage Channels and Containers** link.

The **Portal Desktop: Channels** frame appears, which allows you to create and edit Portal Desktop channels, along with related features. See Figure 132.

Figure 132 Access Manager Page: Portal Desktop - Channels



Identity Management | Service Configuration | Current Sessions | Federation Management

Services

Add... | Remove

Name
- Access Manager Configuration
  - □ Administration ▸
  - □ Authentication Configuration ▸
  - Authentication Modules
    - □ Core ▸
    - □ LDAP ▸
    - □ Membership ▸
  - □ Policy Configuration ▸
  - □ User ▸
- Mobile Application Configuration
  - □ Mobile Address Book ▸
  - □ Mobile Calendar ▸
  - □ Mobile Mail ▸
- Portal Server Configuration
  - □ NetMail ▸
  - □ Portal Desktop ▸

Portal Desktop - Channels

Container Path: Top

Back To Portal Desktop

Display Profile Top Level: Edit Properties...

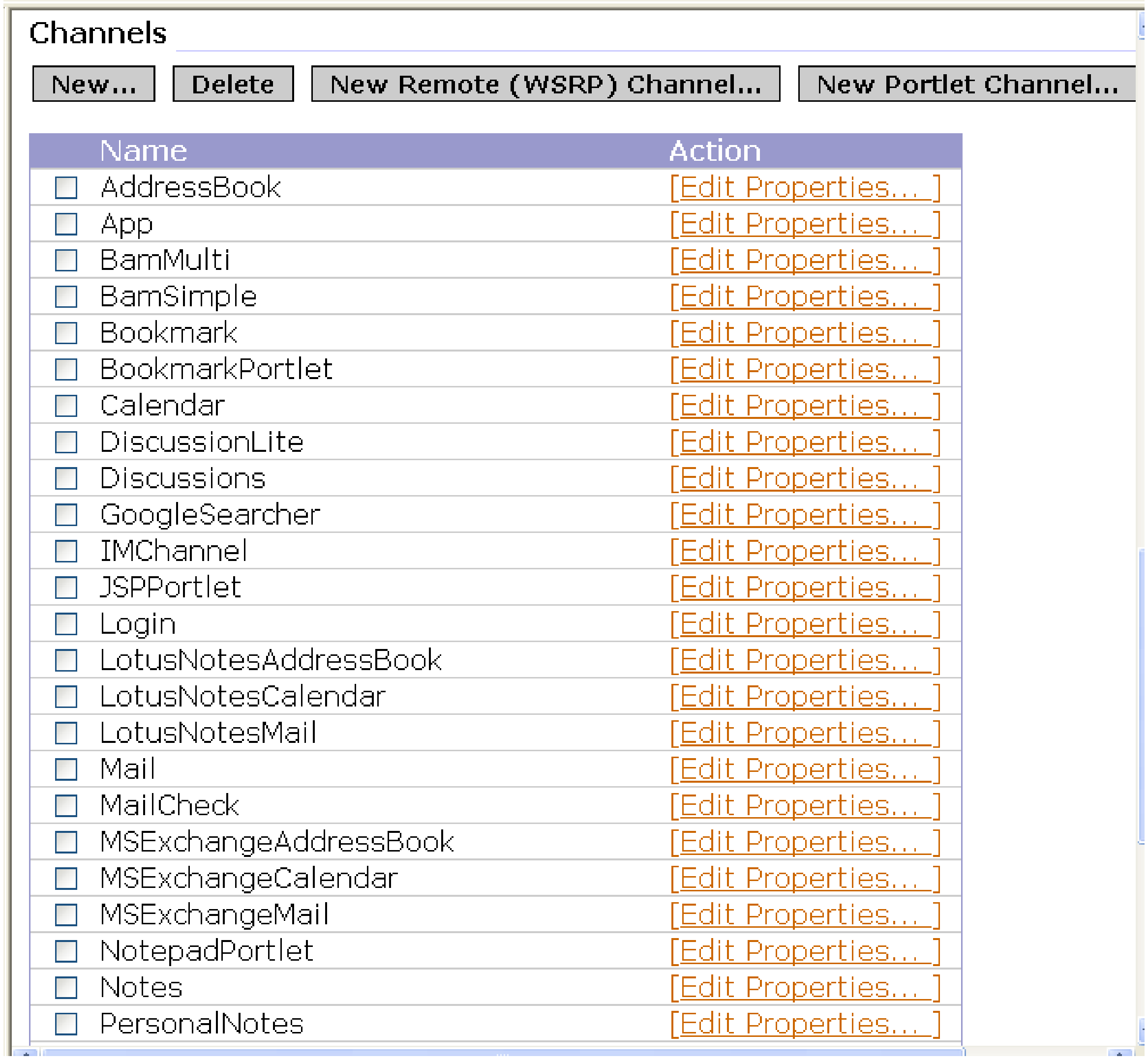
Container Channels

New... | Delete

	Name	Action
□	CollaborationTabPanelContainer	[Edit Properties...]
□	FrameCustomTableContainer	[Edit Properties...]
□	FrameTabContainer	[Edit Properties...]
□	JSPNativeContainer	[Edit Properties...]
□	JSPRenderingContainer	[Edit Properties...]
□	JSPTabContainer	[Edit Properties...]
□	JSPTabCustomTableContainer	[Edit Properties...]
□	JSPTableContainer	[Edit Properties...]
□	MyFrontPageFramePanelContainer	[Edit Properties...]

Scroll down the **Portal Desktop - Channels** frame to the **Channels** section, which displays a list of existing channels. See Figure 133.

Figure 133 List of Existing Channels



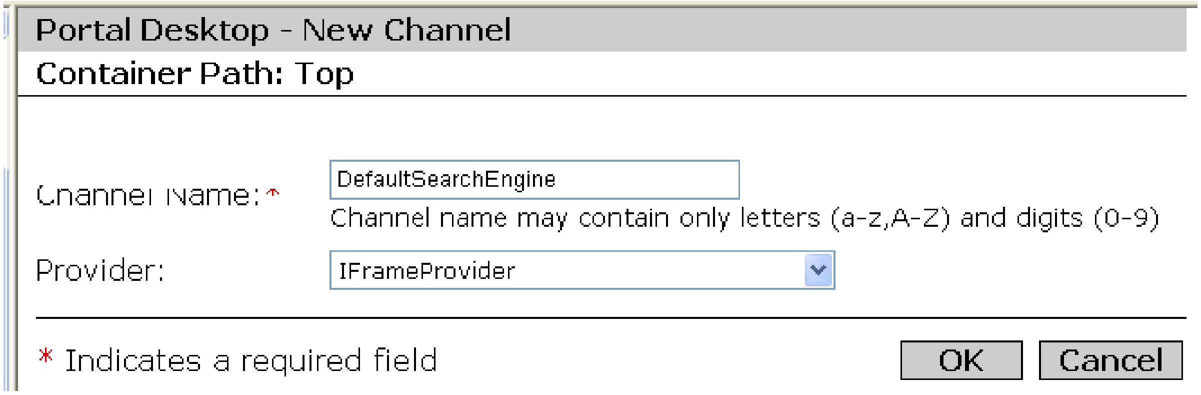
	Name	Action
□	AddressBook	[Edit Properties...]
□	App	[Edit Properties...]
□	BamMulti	[Edit Properties...]
□	BamSimple	[Edit Properties...]
□	Bookmark	[Edit Properties...]
□	BookmarkPortlet	[Edit Properties...]
□	Calendar	[Edit Properties...]
□	DiscussionLite	[Edit Properties...]
□	Discussions	[Edit Properties...]
□	GoogleSearcher	[Edit Properties...]
□	IMChannel	[Edit Properties...]
□	JSPPortlet	[Edit Properties...]
□	Login	[Edit Properties...]
□	LotusNotesAddressBook	[Edit Properties...]
□	LotusNotesCalendar	[Edit Properties...]
□	LotusNotesMail	[Edit Properties...]
□	Mail	[Edit Properties...]
□	MailCheck	[Edit Properties...]
□	MSExchangeAddressBook	[Edit Properties...]
□	MSExchangeCalendar	[Edit Properties...]
□	MSExchangeMail	[Edit Properties...]
□	NotepadPortlet	[Edit Properties...]
□	Notes	[Edit Properties...]
□	PersonalNotes	[Edit Properties...]

5 Click **New**.

6 The **New Channel** frame appears.

7 Enter the name of your new channel and choose **IFrameProvider** on the **Provider** pull-down menu. Enter the **Channel Name** of your new channel. In this example, the name **DefaultSearchEngine** is used. See Figure 134.

Figure 134 New Channel Frame With IFrame Provider



Portal Desktop - New Channel

Container Path: Top

Channel Name:* DefaultSearchEngine

Channel name may contain only letters (a-z,A-Z) and digits (0-9)

Provider: IFrameProvider

* Indicates a required field

8 Click **OK**.

Portal Desktop - Channels frame appears again. See **Figure 132 on page 215**.

9 Scroll down this frame to the **Channels** section again. See **Figure 133 on page 216**

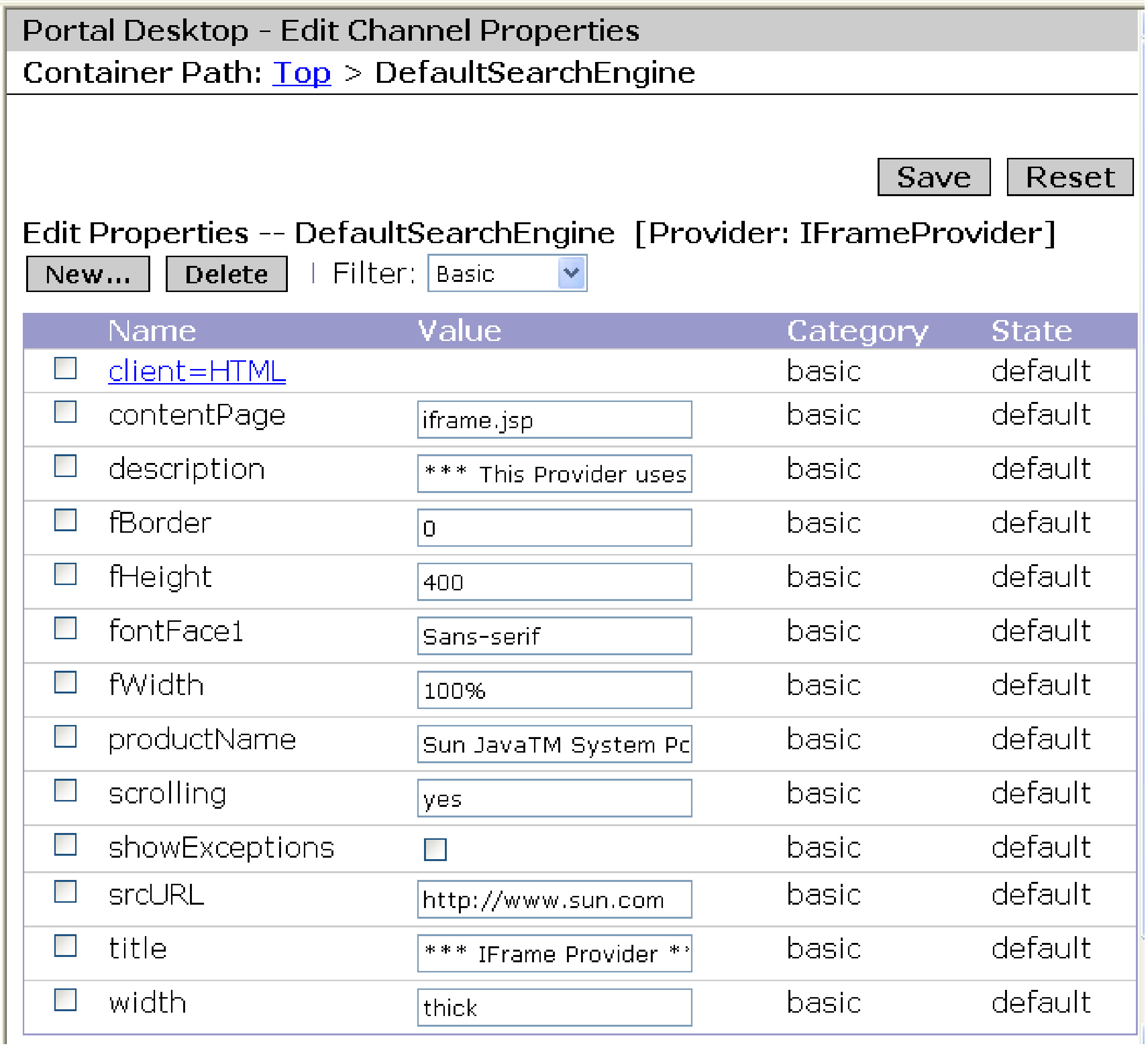
However, the section now contains the name of the new channel you just created.

10 Locate this channel’s name. If you used the example shown in Figure 134, the name is **DefaultSearchEngine**.

11 Click the **Edit Properties** link to the right of **DefaultSearchEngine**, or the alternate channel name you entered, if applicable.

The **Edit Properties** frame for the channel appears, displaying the channel’s default properties. See Figure 135.

Figure 135 Edit Properties Frame for New Channel



Portal Desktop - Edit Channel Properties

Container Path: Top > DefaultSearchEngine

Edit Properties -- DefaultSearchEngine [Provider: IFrameProvider]

Filter: Basic

	Name	Value	Category	State
□	client=HTML		basic	default
□	contentPage	iframe.jsp	basic	default
□	description	*** This Provider uses	basic	default
□	fBorder	0	basic	default
□	fHeight	400	basic	default
□	fontFace1	Sans-serif	basic	default
□	fWidth	100%	basic	default
□	productName	Sun JavaTM System Pc	basic	default
□	scrolling	yes	basic	default
□	showExceptions	□	basic	default
□	srcURL	http://www.sun.com	basic	default
□	title	*** IFrame Provider **	basic	default
□	width	thick	basic	default

12 Edit the properties, as necessary. Make sure you enter the appropriate access URL for your new channel, using the **srcURL** text box.

It is recommended that you enter a descriptive **title** property name of your own choosing, to replace the default. This name appears in the new channel's frame title bar.

13 Click **Save**.

Your new channel is now ready to be placed on the Portal Desktop. See **Placing a New Channel Into a Portal Desktop Tab** on page 218 for an explanation of how to associate a channel with a tap on the Portal Desktop.

Placing a New Channel Into a Portal Desktop Tab

Use Access Manager Console to place a new channel into a tab on the Portal Desktop.

To view the Access Manager Console page

- Use the procedure on page 209.

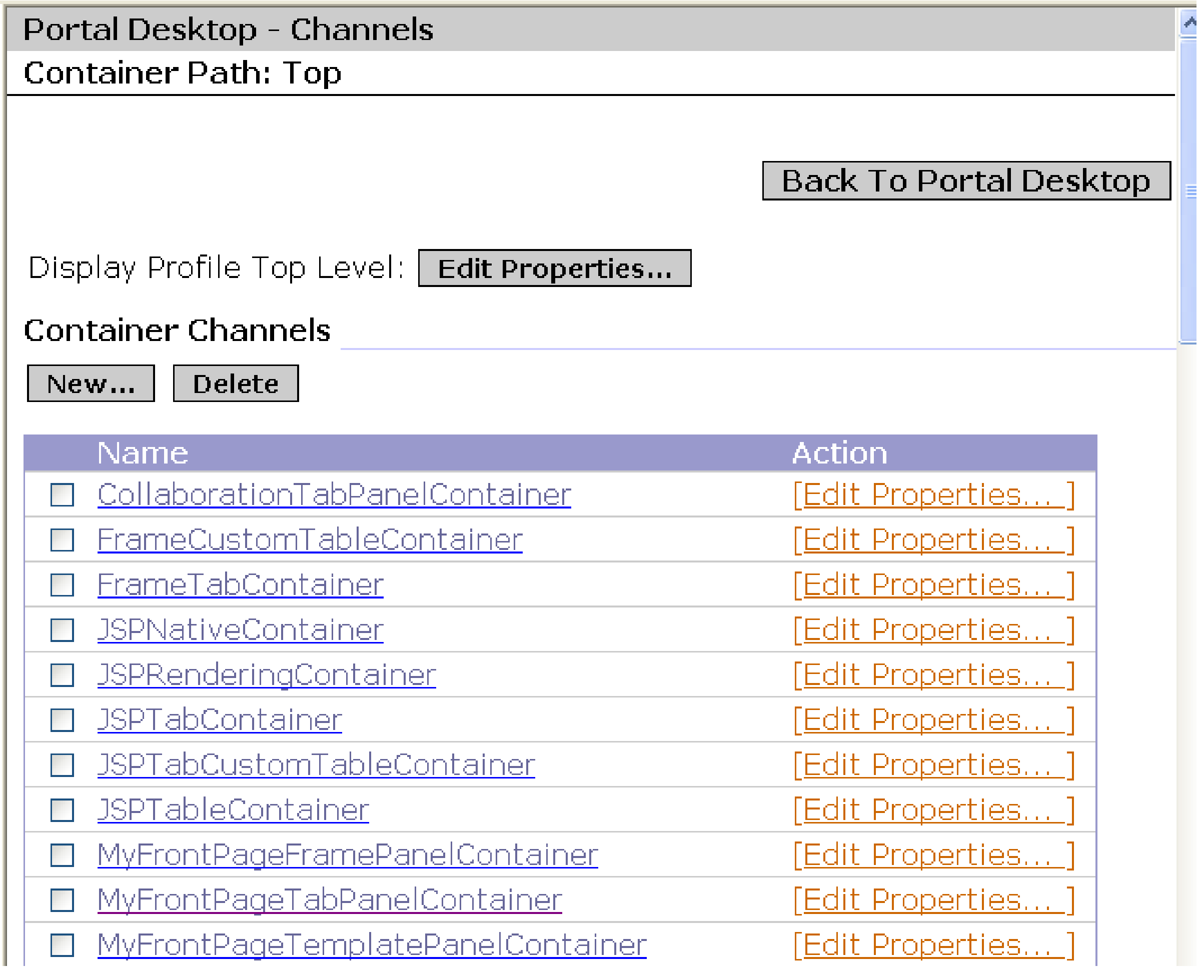
To place a channel on the Portal Desktop

1. On the Access Manager page, make sure the **Identity Management** tab content is displayed.
2. From the **View** drop-down list select **Services**. See Figure 130 on page 214
3. Scroll down the **Services** (left) frame to display the **Portal Desktop** options and click the pointer icon to the right of **Portal Desktop**.

   The **Portal Desktop** frame appears on the right. See Figure 131 on page 215.
4. On the **Portal Desktop** frame, click the **Manage Channels and Containers** link.

   The **Portal Desktop - Channels** frame appears. See Figure 132 on page 215.
5. Scroll down the **Portal Desktop - Channels** frame to the **Container Channels** section (if it is not already displayed. This section displays a list of existing Portal Desktop tab container channels. See Figure 136.

Figure 136 List of Container Channels



Portal Desktop - Channels

Container Path: Top

Display Profile Top Level:

Container Channels

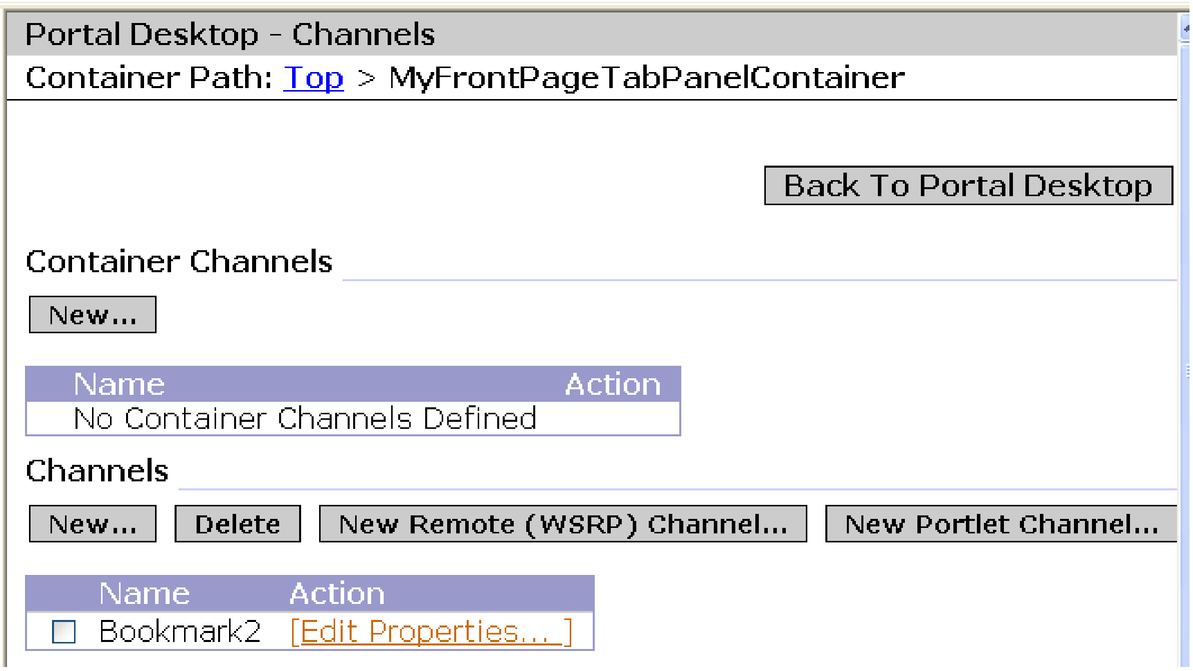
	Name	Action
□	CollaborationTabPanelContainer	[Edit Properties...]
□	FrameCustomTableContainer	[Edit Properties...]
□	FrameTabContainer	[Edit Properties...]
□	JSPNativeContainer	[Edit Properties...]
□	JSPRenderingContainer	[Edit Properties...]
□	JSPTabContainer	[Edit Properties...]
□	JSPTabCustomTableContainer	[Edit Properties...]
□	JSPTableContainer	[Edit Properties...]
□	MyFrontPageFramePanelContainer	[Edit Properties...]
□	MyFrontPageTabPanelContainer	[Edit Properties...]
□	MyFrontPageTemplatePanelContainer	[Edit Properties...]

6 Click **MyFrontPageTabPanelContainer**.

Do **not** click the **Edit Properties** link.

7 A frame appears, which allows you to manage the channels that appear on your **Front Page** tab. See Figure 137.

Figure 137 Initial MyFrontPageTabPanelContainer Frame



Portal Desktop - Channels

Container Path: Top > MyFrontPageTabPanelContainer

Container Channels

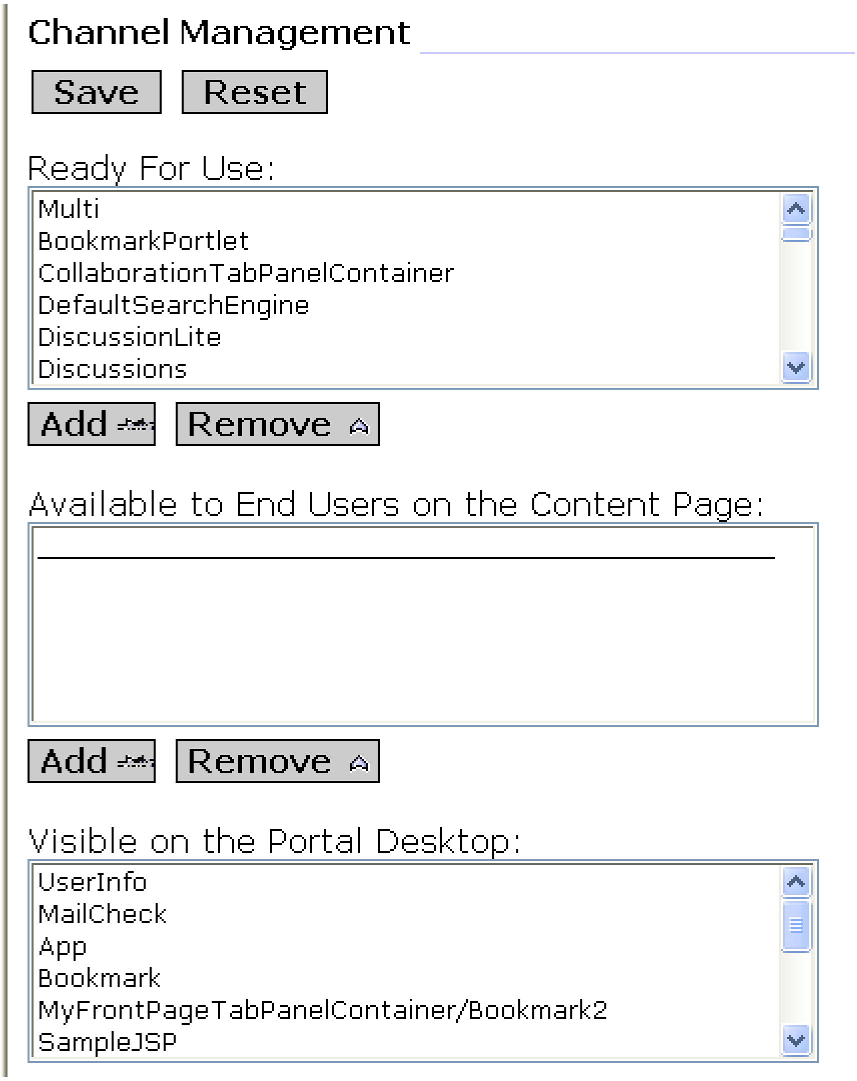
Name	Action
No Container Channels Defined	

Channels

	Name	Action
□	Bookmark2	[Edit Properties...]

8 Scroll down this frame to the **Channel Management** section. See Figure 138.

Figure 138 Channel Management Section: Initial



Ready For Use:

Multi
BookmarkPortlet
CollaborationTabPanelContainer
DefaultSearchEngine
DiscussionLite
Discussions

Available to End Users on the Content Page:

Visible on the Portal Desktop:

UserInfo
MailCheck
App
Bookmark
MyFrontPageTabPanelContainer/Bookmark2
SampleJSP

The **Ready for Use** pane contains a list of existing channels.

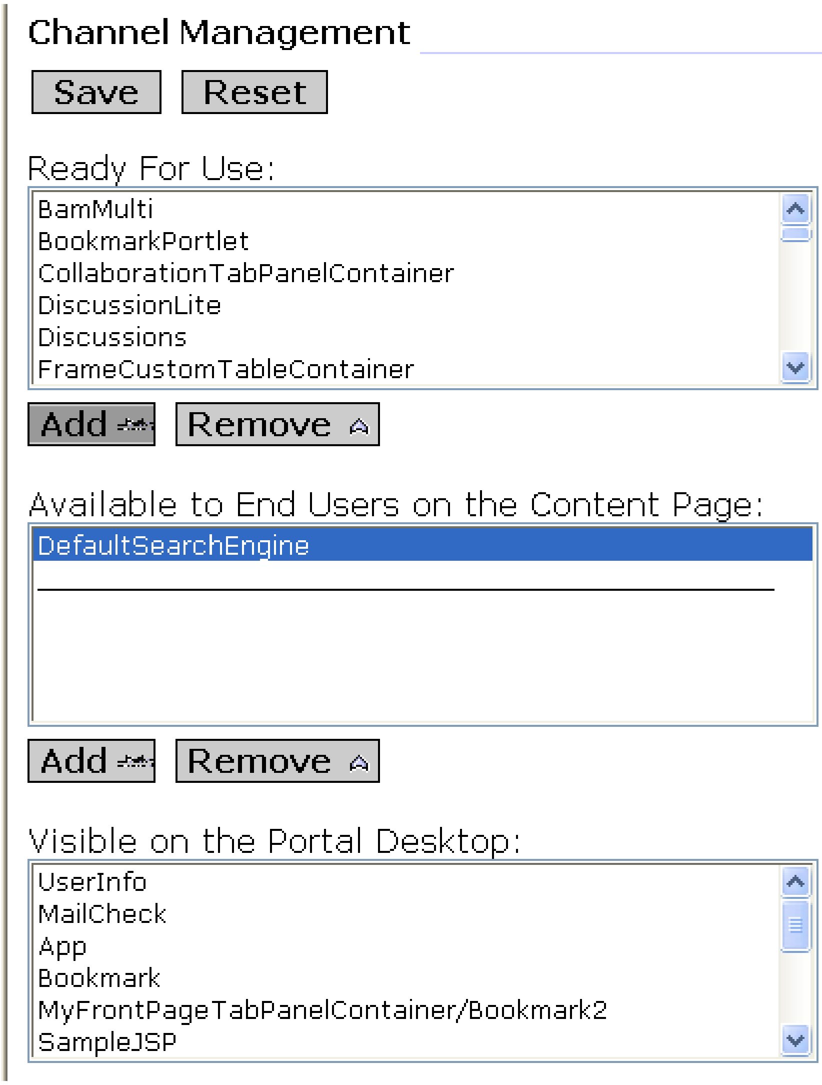
The section shown in the figure is an example. Your ***Channel Management*** section contains the Portal Server defaults plus any additional channels that have been created for your system.

9 Under **Ready for Use**, select the new channel you created under the **procedure on page 213**. In the example, the new channel is **DefaultSearchEngine**.

10 Click **Add**.

Your new channel name appears under **Available to End Users on the Content Page**. See Figure 139.

Figure 139 Channel Management Section: First Add



Channel Management

Ready For Use:

BamMulti
BookmarkPortlet
CollaborationTabPanelContainer
DiscussionLite
Discussions
FrameCustomTableContainer

Available to End Users on the Content Page:

DefaultSearchEngine

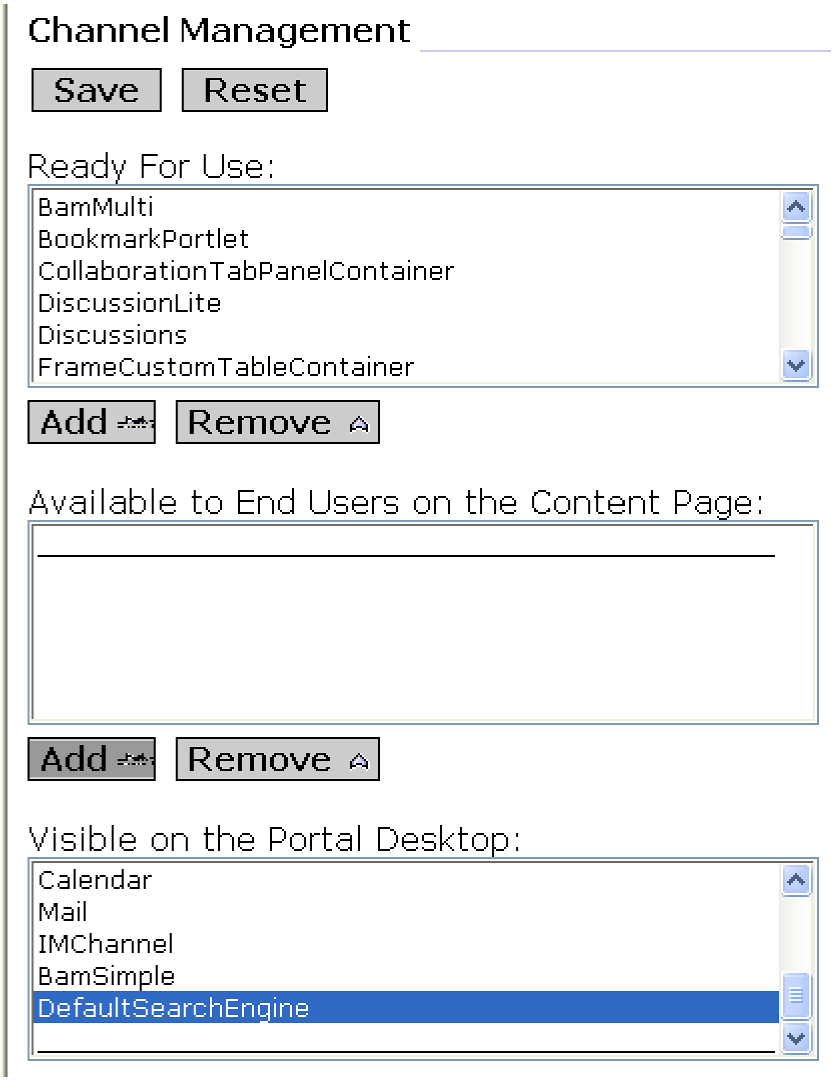
Visible on the Portal Desktop:

UserInfo
MailCheck
App
Bookmark
MyFrontPageTabPanelContainer/Bookmark2
SampleJSP

11 Select the new channel (if necessary) and click **Add**.

Your new channel name appears under **Visible on the Portal Desktop**. The name appears at the end of the list, so you may need to scroll down the pane to view it. See Figure 140.

Figure 140 Channel Management Section: Second Add



12 Click **Save**.

Your new channel is now a part of the **Front Page** tab. This channel appears the next time a user with permission to view it accesses the Portal Desktop and selects the **Front Page** tab.

Viewing a New Channel

Using this procedure, you can place any existing channel on any existing tab panel page you choose, and then view that channel.

To view your new channel on the Portal Desktop

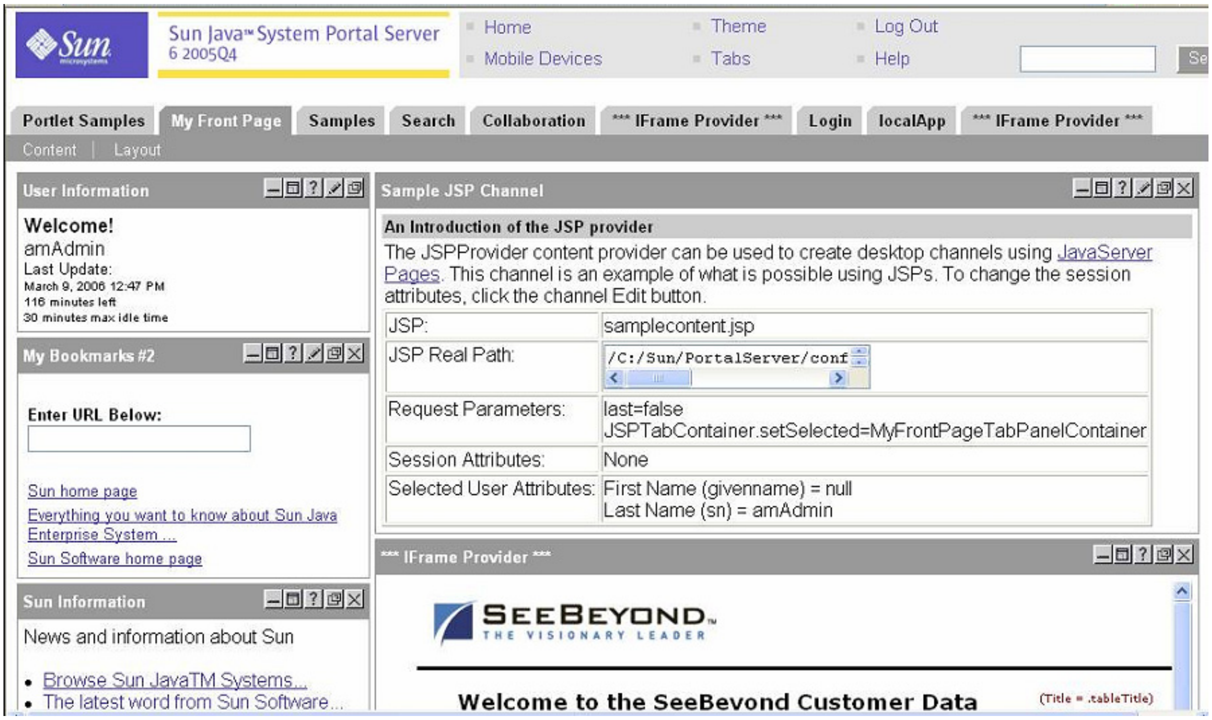
1 Perform either of the following operations:

- On the Windows **Start > All Programs** menu, choose **Sun Microsystems > Portal Server > Portal Desktop**.
- Enter the following URL on your Web browser:

```
http://<host name>:38080/amserver/portal/dt
```

The Portal Desktop Welcome page appears with the **My Front Page** tab displayed. See Figure 141, which shows an example.

Figure 141 Portal Server 6 Portal Desktop Welcome Page



Welcome!
amAdmin
Last Update:
March 9, 2006 12:47 PM
116 minutes left
30 minutes max idle time

Enter URL Below:

Sun home page
Everything you want to know about Sun Java Enterprise System ...
Sun Software home page

News and information about Sun

- Browse Sun JavaTM Systems...
- The latest word from Sun Software...

An Introduction of the JSP provider

The JSPProvider content provider can be used to create desktop channels using JavaServer Pages. This channel is an example of what is possible using JSPs. To change the session attributes, click the channel Edit button.

JSP:	samplecontent.jsp
JSP Real Path:	/C:/Sun/PortalServer/conf

Request Parameters:	last=false
	JSPTabContainer.setSelected=MyFrontPageTabPanelContainer

Session Attributes:	None
Selected User Attributes:	First Name (givenname) = null Last Name (sn) = amAdmin

Welcome to the SeeBeyond Customer Data

2 You can scroll down this page to view your new Portal Server channel.

9.5.4 Running eVision Studio Applications with Portal Server 7

This section explains how to set up Sun Java System Portal Server 7 to run with the Java CAPS applications, including eVision Studio.

Before You Start

Take care to ensure you have completed the prerequisite actions listed under **Before Installing Portal Server** on page 200.

System Requirements

You must run Portal Server 7 in:

- Sun Java System Application Server, Version 8.1

If you want to deploy JSR 168 portlets with eVision Studio, you must deploy Portal 7 together with the application server on either of the following platforms:

- Solaris 10 OS for SPARC and x86 platforms

- Solaris 9 OS for SPARC and x86 platforms
- Solaris 8 OS for the SPARC platform
- Red Hat Enterprise Linux WS/AS/ES 2.1 to 2.1U6
- Red Hat Enterprise Linux WS/AS/ES 3.0 to 3.0U4

Supported Operating Systems

For specific information on all operating systems supported by Portal Server 7 in conjunction with the Java CAPS and eVision Studio, see the **Readme.txt** file for eVision Studio.

The port numbers provided in all access URL examples are the system defaults. The actual port numbers used by your system's configuration may be different. Before you begin using the procedures provided in the rest of this section, make sure you have verified the correct port number for any URL referenced by a given procedure.

Installation

For complete instructions on installation, see the Sun Microsystems *Sun Java System Portal Server 7 Installation Guide*. See **Reference Documentation** on page 199 for an explanation of how to find this document.

Setting Up an IFrame Channel

This section describes how to create and configure an IFrame channel for Portal Server 7.

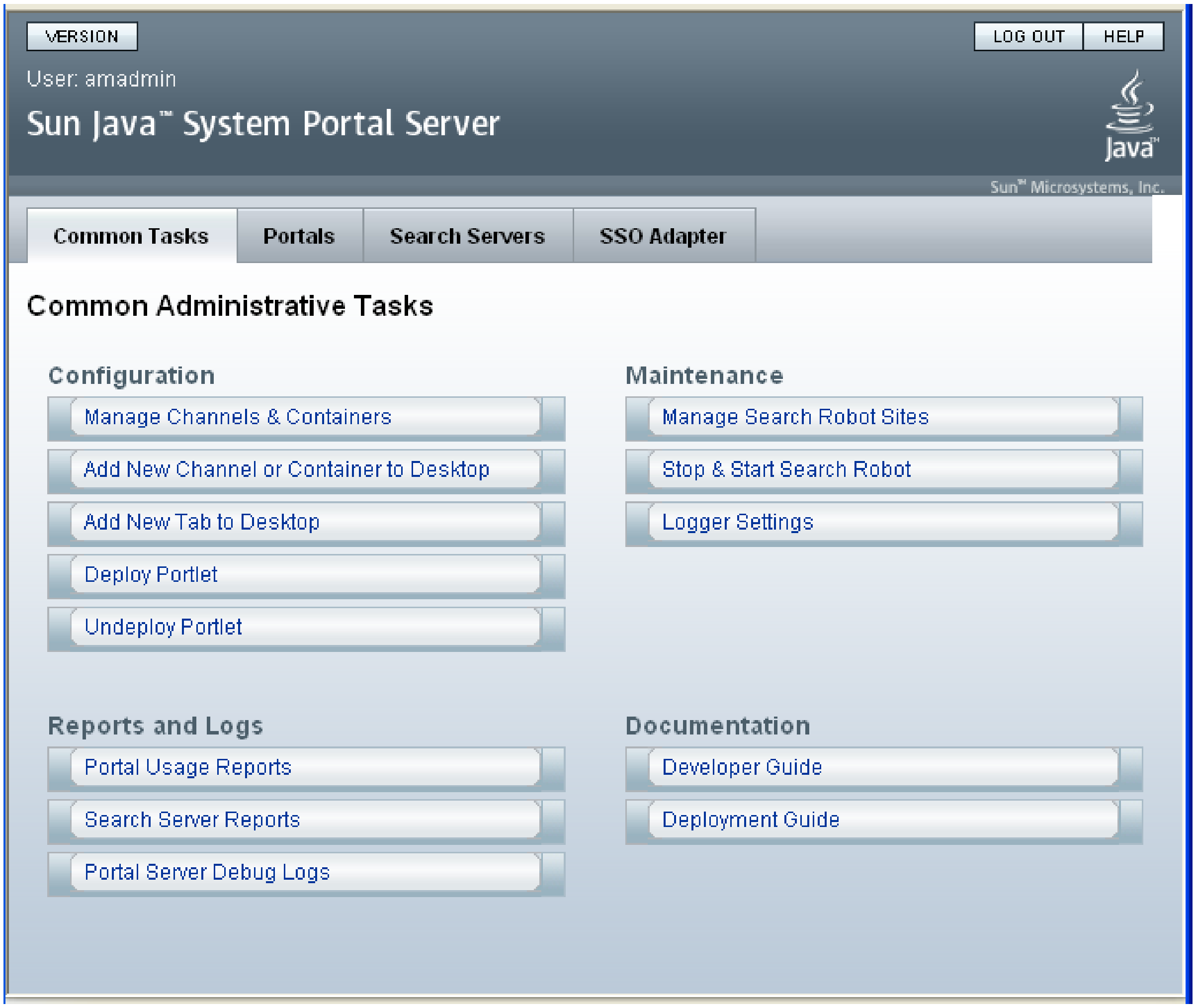
To set up an IFrame channel

1. Log on to Portal Server 7 Console (**psconsole**) with Access Manager privileges, using the following URL:

   ```
   http://<host name>:38080/psconsole
   ```

   The Portal Server 7 Console page appears. See Figure 142.

Figure 142 Portal Server 7 Console Page



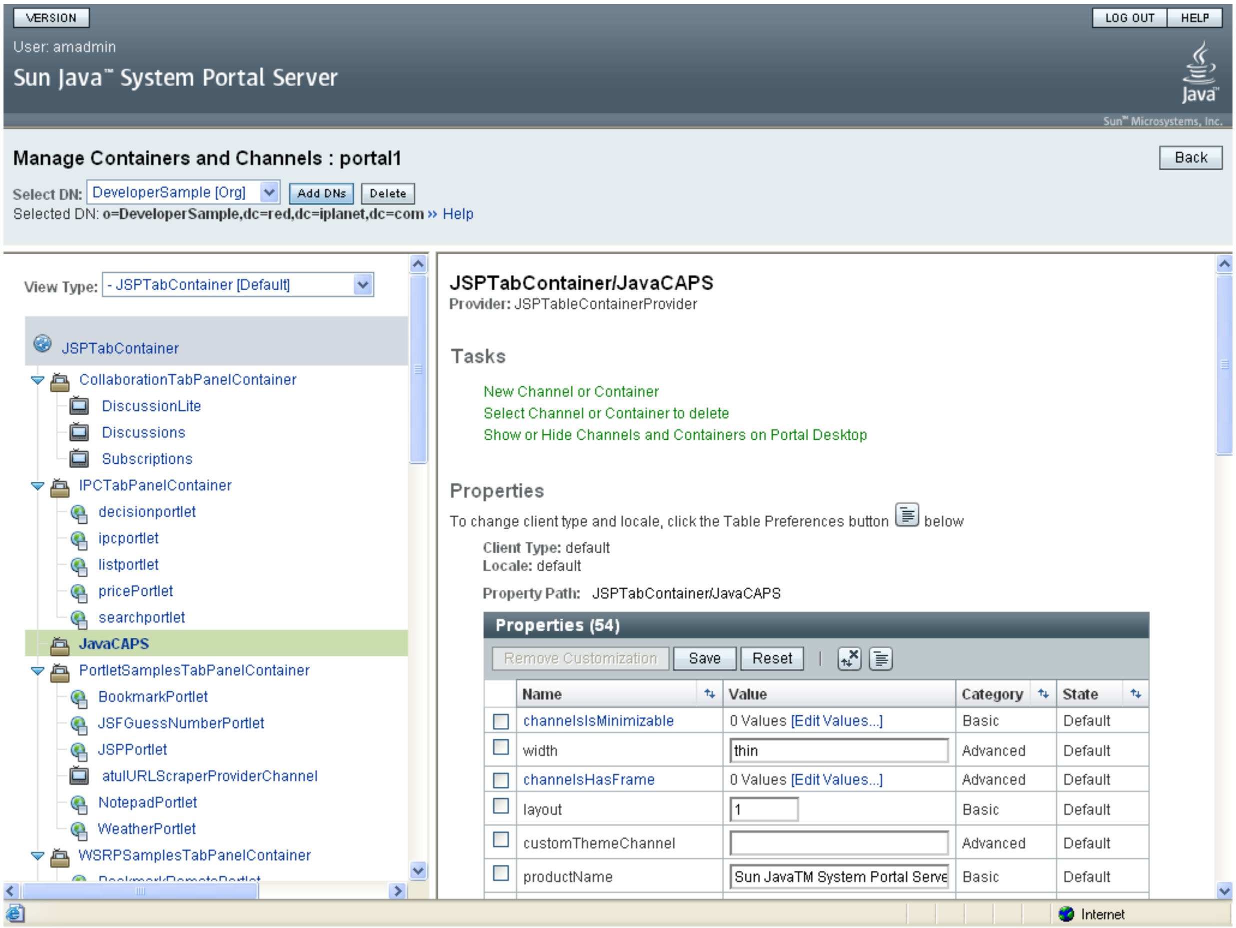
2 On the **Common Tasks** tab, under **Configuration**, click **Manage Channels And Containers**.

A window appears that allows you to choose the portal and distinguished name (DN) you want to create the channel for.

3 For this example, choose **portal1** for the portal and **DeveloperSample [Org]** for the DN, and click **OK**.

The Manage Containers and Channels: portal1 page appears. See Figure 143.

Figure 143 Manage Containers and Channels : portal1 Page: Initial



Manage Containers and Channels : portal1

Select DN: DeveloperSample [Org]

Selected DN: o=DeveloperSample,dc=red,dc=iplanet,dc=com » Help

View Type: - JSPTabContainer [Default]

JSPTabContainer/JavaCAPS
Provider: JSPTableContainerProvider

Tasks

New Channel or Container
Select Channel or Container to delete
Show or Hide Channels and Containers on Portal Desktop

Properties

To change client type and locale, click the Table Preferences button below

Client Type: default
Locale: default

Property Path: JSPTabContainer/JavaCAPS

Properties (54)

	Name	Value	Category	State
□	channelsIsMinimizable	0 Values [Edit Values...]	Basic	Default
□	width	thin	Advanced	Default
□	channelsHasFrame	0 Values [Edit Values...]	Advanced	Default
□	layout	1	Basic	Default
□	customThemeChannel		Advanced	Default
□	productName	Sun JavaTM System Portal Serve	Basic	Default

Using this page, you can choose an existing tab or create a new one. For an explanation of how to create a new tab, see the Sun Microsystems **Sun Java System Portal Server 7 Technical Overview** document.

4 In the left frame, select the container in which you want to create the new channel. In the example in Figure 143 you select the **JavaCAPS** container.

5 In the right frame, select **New Channel or Container**.

A new window appears. Using that window, perform the following steps:

A Select the Portal instance and DN you want and click **Next**.

B On the next window, choose **Provider Channel** for the **Channel Type** and click **Next**.

C On the next window, choose **IFrameProvider** for the **Provider** and click **Next**.

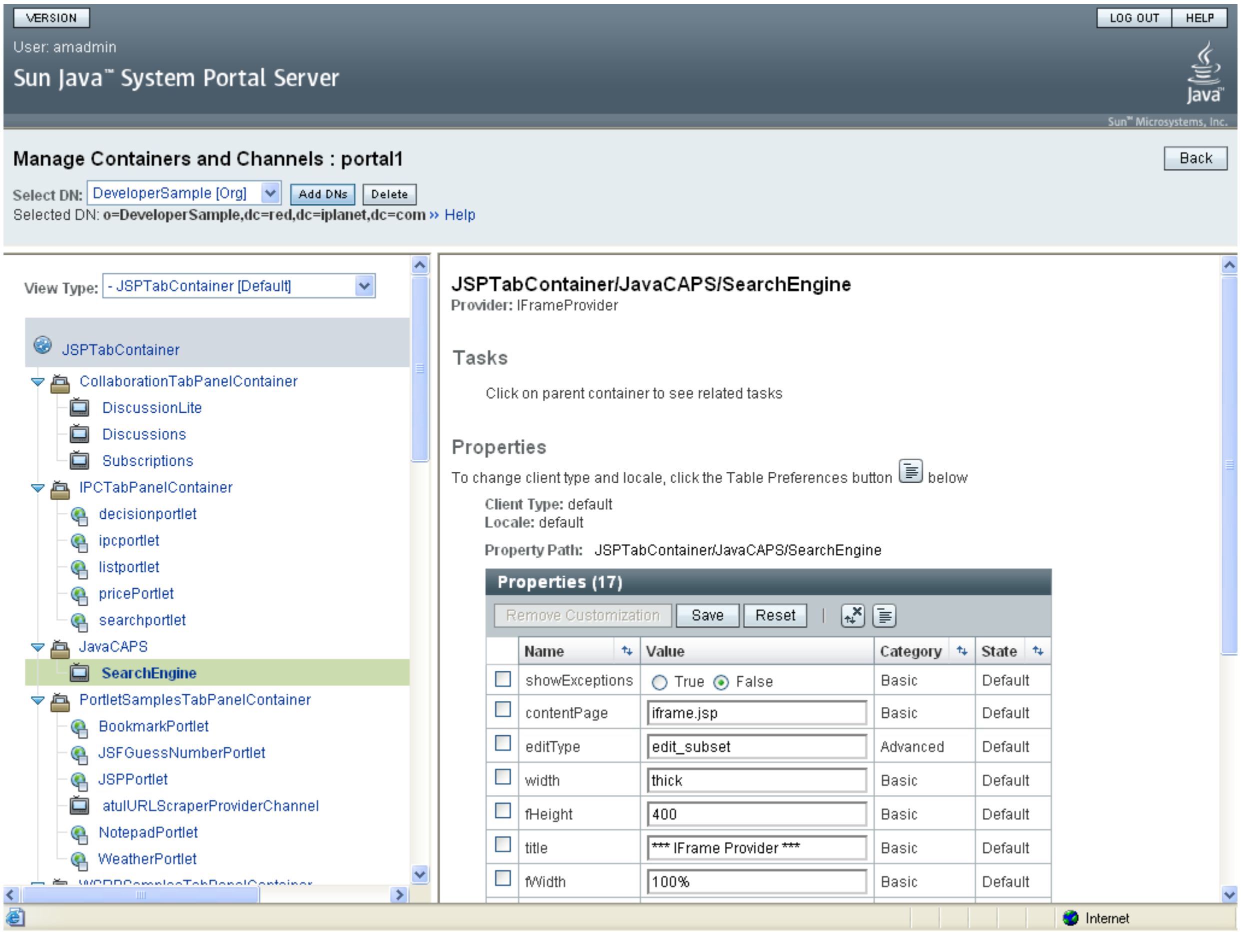
D On the next window, enter a name for your channel, **SearchEngine** for this example, and click **Next**.

E In the next window, view the information you have entered to make sure it is correct, and then click **Finish**.

F On the next window, which indicates you have successfully created a new channel, click **Close**.

The Manage Containers and Channels: portal1 page appears again. The page now displays properties for your new channel, in the right frame. See Figure 144.

Figure 144 Manage Containers and Channels : portal1 Page: With New Channel



User: amadmin

Manage Containers and Channels : portal1

Select DN: DeveloperSample [Org]

Selected DN: o=DeveloperSample,dc=red,dc=iplanet,dc=com

View Type: - JSPTabContainer [Default]

JSPTabContainer/JavaCAPS/SearchEngine

Provider: IFrameProvider

Tasks

Click on parent container to see related tasks

Properties

To change client type and locale, click the Table Preferences button below

Client Type: default
Locale: default

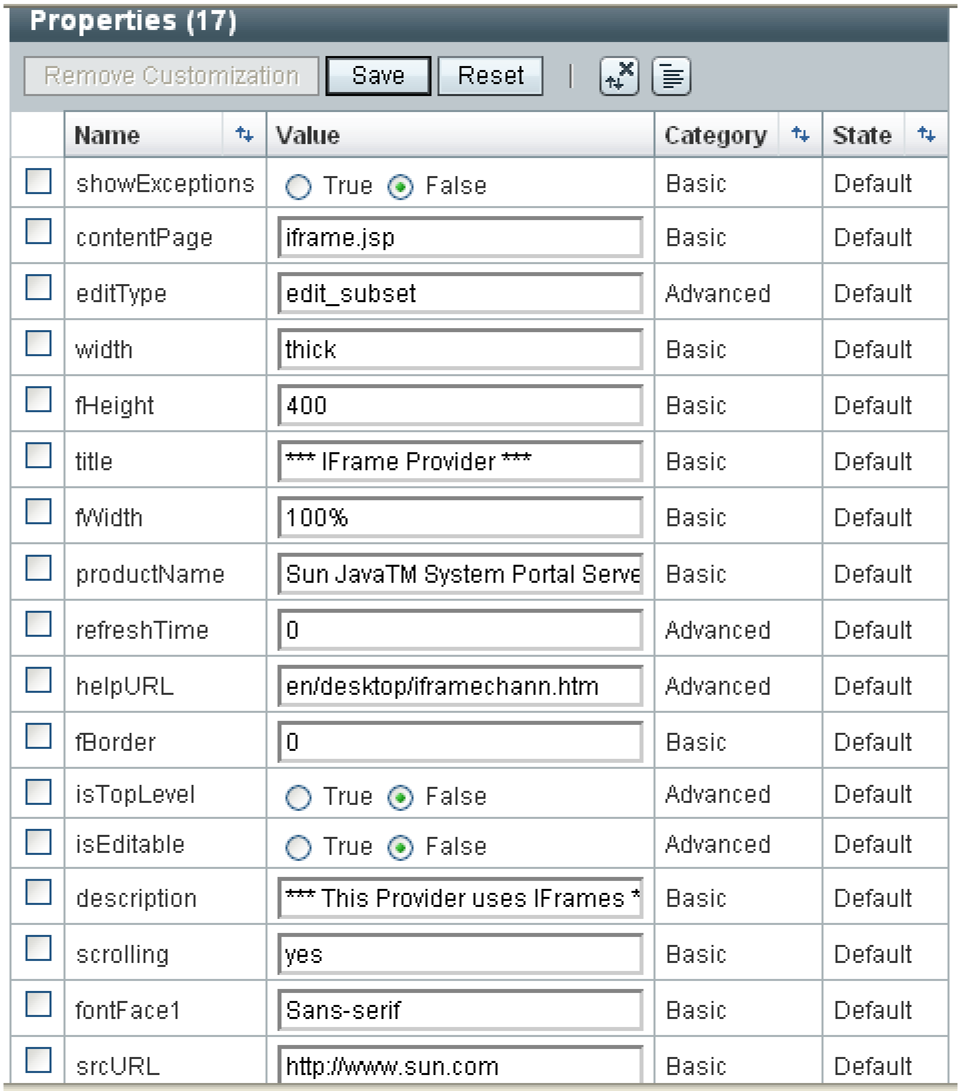
Property Path: JSPTabContainer/JavaCAPS/SearchEngine

Properties (17)

	Name	Value	Category	State
□	showExceptions	○ True ● False	Basic	Default
□	contentPage	iframe.jsp	Basic	Default
□	editType	edit_subset	Advanced	Default
□	width	thick	Basic	Default
□	fHeight	400	Basic	Default
□	title	*** IFrame Provider ***	Basic	Default
□	fWidth	100%	Basic	Default

6 On the right side of the page, scroll down the pane to display all the new channel's properties. See Figure 145.

Figure 145 Channel Properties Pane



	Name	Value	Category	State
□	showExceptions	○ True ● False	Basic	Default
□	contentPage	iframe.jsp	Basic	Default
□	editType	edit_subset	Advanced	Default
□	width	thick	Basic	Default
□	fHeight	400	Basic	Default
□	title	*** IFrame Provider ***	Basic	Default
□	fWidth	100%	Basic	Default
□	productName	Sun JavaTM System Portal Serve	Basic	Default
□	refreshTime	0	Advanced	Default
□	helpURL	en/desktop/iframechann.htm	Advanced	Default
□	fBorder	0	Basic	Default
□	isTopLevel	○ True ● False	Advanced	Default
□	isEditable	○ True ● False	Advanced	Default
□	description	*** This Provider uses IFrames *	Basic	Default
□	scrolling	yes	Basic	Default
□	fontFace1	Sans-serif	Basic	Default
□	srcURL	http://www.sun.com	Basic	Default

7 Be sure to enter the URL for your new channel under **srcURL** and give your new channel a **title**.

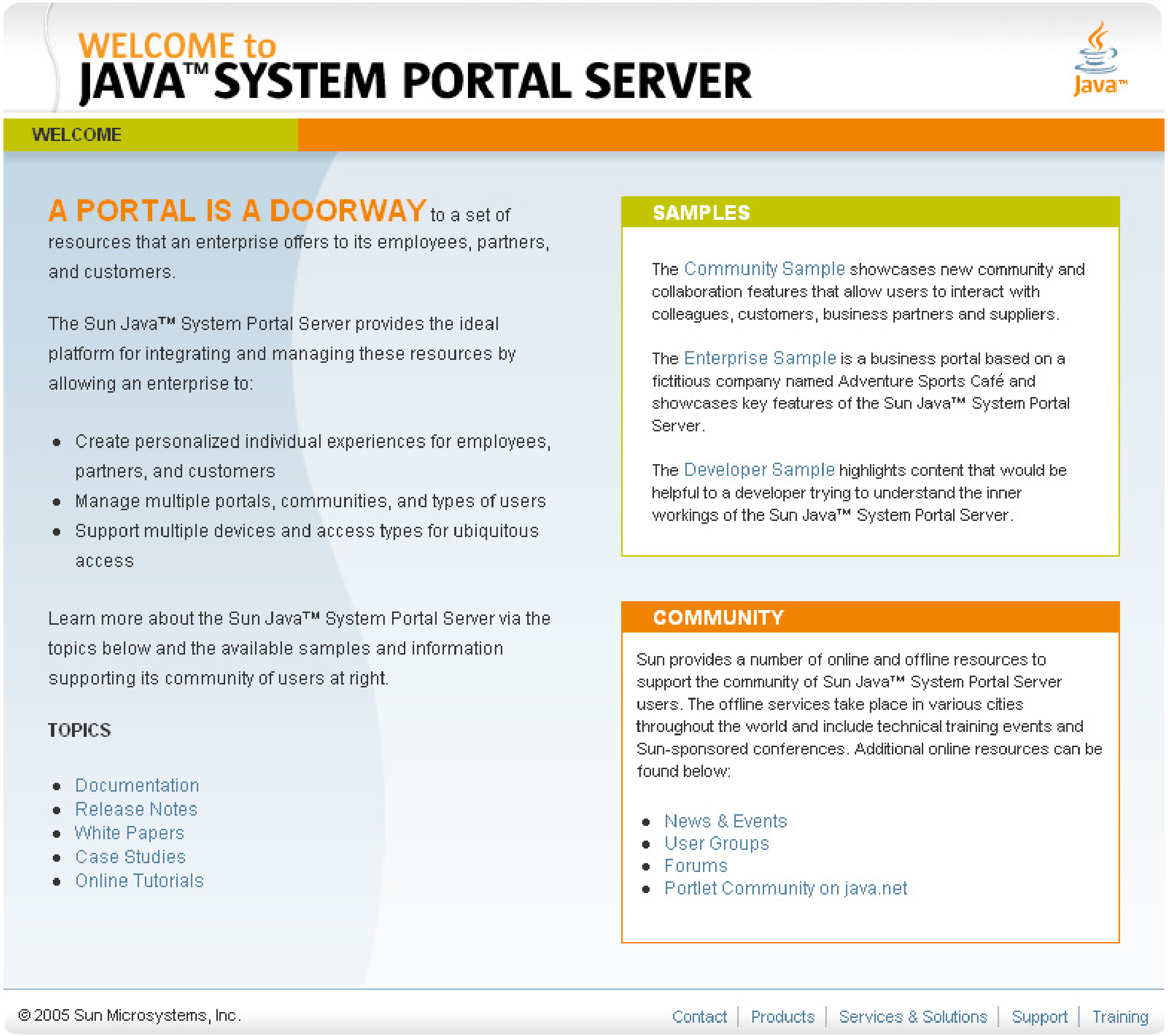
8 Supply other configuration parameters, as needed.

9 When you are finished, click **Save**.

10 Access the Portal Desktop, using `http://<host name>:38080/portal`, to view your new channel.

The Portal Desktop Welcome page appears. See Figure 146.

Figure 146 Portal Server 7 Portal Desktop Welcome Page



WELCOME to
JAVA™ SYSTEM PORTAL SERVER

WELCOME

A PORTAL IS A DOORWAY to a set of resources that an enterprise offers to its employees, partners, and customers.

The Sun Java™ System Portal Server provides the ideal platform for integrating and managing these resources by allowing an enterprise to:

- Create personalized individual experiences for employees, partners, and customers
- Manage multiple portals, communities, and types of users
- Support multiple devices and access types for ubiquitous access

Learn more about the Sun Java™ System Portal Server via the topics below and the available samples and information supporting its community of users at right.

TOPICS

- Documentation
- Release Notes
- White Papers
- Case Studies
- Online Tutorials

SAMPLES

The Community Sample showcases new community and collaboration features that allow users to interact with colleagues, customers, business partners and suppliers.

The Enterprise Sample is a business portal based on a fictitious company named Adventure Sports Café and showcases key features of the Sun Java™ System Portal Server.

The Developer Sample highlights content that would be helpful to a developer trying to understand the inner workings of the Sun Java™ System Portal Server.

COMMUNITY

Sun provides a number of online and offline resources to support the community of Sun Java™ System Portal Server users. The offline services take place in various cities throughout the world and include technical training events and Sun-sponsored conferences. Additional online resources can be found below:

- News & Events
- User Groups
- Forums
- Portlet Community on java.net

© 2005 Sun Microsystems, Inc.

Contact | Products | Services & Solutions | Support | Training

11 On the Welcome page, select the **Developer Sample** link.

The Developer Sample page appears.

12 In this example, your new channel appears under the **Java CAPS** tab.

Setting Up a JSR 168 Portlet Channel

This section describes how to create and configure a JSR 168 portlet channel for Portal Server 7.

Portal Server 7 patches are required to register Java CAPS JSR 168 portlets. See the Portal Server technical note "Integrating Java CAPS With Sun Java System Portal Server" referenced under **Necessary Patches** on page 201, for installation instructions.

Before You Start

Before creating a JSR 168 channel, you must deploy the eVision Studio **.ear** file to the application server on which Portal Server is running and register the same **.ear** file with Portal Server.

To deploy eVision Studio applications

1. Make sure the Sun Java System Application Server is running.
2. Java CAPS applications require special permissions. Make sure the appropriate permissions have been set up correctly in the **server.policy** file.
3. Deploy the eVision Studio **.ear** file to the application server on which Portal Server is running. You can do this operation using:
   - Enterprise Designer
   - Enterprise Manager
   - A Sun Java System Application Server's administrator interface, either Admin Console or the command line; a command-line example follows:

     ```
     ./asadmin deploy --user admin /tmp/prj2dp8.ear
     ```

See the *eGate Integrator User's Guide* and *eGate Integrator System Administration Guide* for details on how to deploy an **.ear** file using Enterprise Designer and Enterprise Manager. See the **procedure on page 206** for information on how to deploy an **.ear** file using Admin Console.

Registering eVision Studio JSR 168 application with Portal Server 7

Use the **psadmin register-portlet** command in the system's command line to register the portlet **.ear** file to a particular portal and DN. The following text shows an example of this command:

```
./psadmin register-portlet -u amadmin -f pfile -p portal1
-d o=Sample,dc=sun,dc=com
/tmp/168/simplejsr168dpsun.ear
```

Note: *For details on how to use the command line, see the Sun Microsystems* **Sun Java System Portal Server 7 Command-Line Reference** *document.*

Note: *If you redeploy an application without changing the application* **.ear** *file name, the ServletContext name and the Portlet name, you do not need to perform registration again.*

After you have done this action, the portlet becomes available in the list of portlets for use with a JSR 168 channel.

To create a JSR 168 portlet channel

1. Follow steps 1 through 4 under **Setting Up an IFrame Channel** on page 225.
2. In the right frame, select **New Channel or Container**.

   A new window appears. Using that window, perform the following steps:

   A Select the Portal instance and DN you want and click **Next**.

B On the next window, choose **JSR 168 Channel** for the **Channel Type** and click **Next**

C On the next window, choose one of the portlet providers you have registered, from the drop-down list.

After registering the portlet with **psadmin**, a provider for your portlet is available from the list of portlet providers. Your provider appears as *<servlet-context-name>.<portlet-name>*. For example, in eVision, *<portlet-name>* defaults to **PageFlowPortlet**. So, if you generate an eVision portlet using **myContext** for *<servlet-context-name>*, the portlet provider is listed as **myContext.PageFlowPortlet**.

D On the next window, enter a name for your channel and click **Next**.

E In the next window, view the information you have entered to make sure it is correct, and then click **Finish**.

F On the next window, which indicates you have successfully created a new channel, click **Close**.

The Manage Containers and Channels: portal1 page appears again. The page now displays your new JSR 168 channel.

3 On the right side of the page, scroll down the pane to display all the new channel's properties.

4 Enter the **title** you want.

5 Supply other configuration parameters, as needed.

6 When you are finished, click **Save**.

7 Access the Portal Desktop, using `http://<host name>:38080/portal`, to view your new channel.

Unregistering eVision Studio JSR 168 application with Portal Server 7

When appropriate, you can use the **psadmin unregister-portlet** command to unregister a portlet. This step requires the name of the **.ear** file that was registered, for example:

```
./psadmin unregister-portlet -u amadmin -f pfile -p portal1
-d o=Sample,dc=sun,dc=com simplejsr168dpsun
```

Chapter 10

Authentication, Authorization, and Error Handling

You can add authentication to eVision Studio web applications, return preconfigured pages for certain errors, and authorize users based on roles.

What's in This Chapter

10.1 Authentication and Error Handling

eVision Studio provides the following authentication options:

- No authentication.
- Using the default Java CAPS authentication.
- Using the preconfigured authentication pages that are provided with eVision Studio.

Table 27 describes the preconfigured authentication pages.

Table 27 Preconfigued Authentication Pages

Authentication Page	Description	Required?
Login Page	This page allows the user to enter login information.	yes
Login Error	This page is returned if the user enters an invalid user name and/or password.	yes
Access Denied Error	This page is returned if the user is not allowed to see the requested page. Access Denied Error corresponds to the HTTP response codes 401 and 403. These response codes are two variations of the same error.	no

You can also return preconfigured error-handling pages to the user when a requested page cannot be found or when a more serious internal error has occurred. You can create these pages regardless of which authentication option you are using.

Table 28 describes the preconfigured error-handling pages that are provided with eVision Studio.

Table 28 Preconfigured Error-Handling Pages

Authentication Page	Description
No Such Resource Error	This page is returned if the requested page cannot be found. No Such Resource corresponds to the HTTP response code 404.
Internal Server Error	This page is returned if there is a serious problem with the business process (for example, the business process is corrupt or the session was lost). Internal Server Error corresponds to the HTTP response code 500.

10.1.1 Creating Authentication and Error-Handling Pages

You can add a preconfigured authentication page or a preconfigured error-handling page to your Project. Perform this procedure for each page that you want to create.

Note: *You do not need to add these pages to the Page Flow.*

To create an authentication or error-handling page

1. In the Project Explorer of Enterprise Designer, right-click the Project.
2. On the shortcut menu, point to **New**, and then click **Page Layout**.

   The **New Page Layout** dialog box appears.
3. Select the authentication or error-handling page that you want to create. Most of the pages are located on the **Error** tab.
4. If desired, change the default name for the Page Layout.
5. Click **OK**.

   The Page Layout Designer appears with the page that you chose.
6. Because the page is preconfigured, you do not need to make any changes in the Page Layout Designer. However, you might want to add a background or modify the title. You can perform these tasks in the **Properties** tab.

10.1.2 Configuring the Connectivity Map

If you are using the default Java CAPS authentication or the preconfigured authentication pages, then you must set configuration parameters for the Web Connector in the Connectivity Map.

If you are using either of the preconfigured error-handling pages, then you must also set configuration parameters for the Web Connector in the Connectivity Map.

To configure authentication in the Connectivity Map

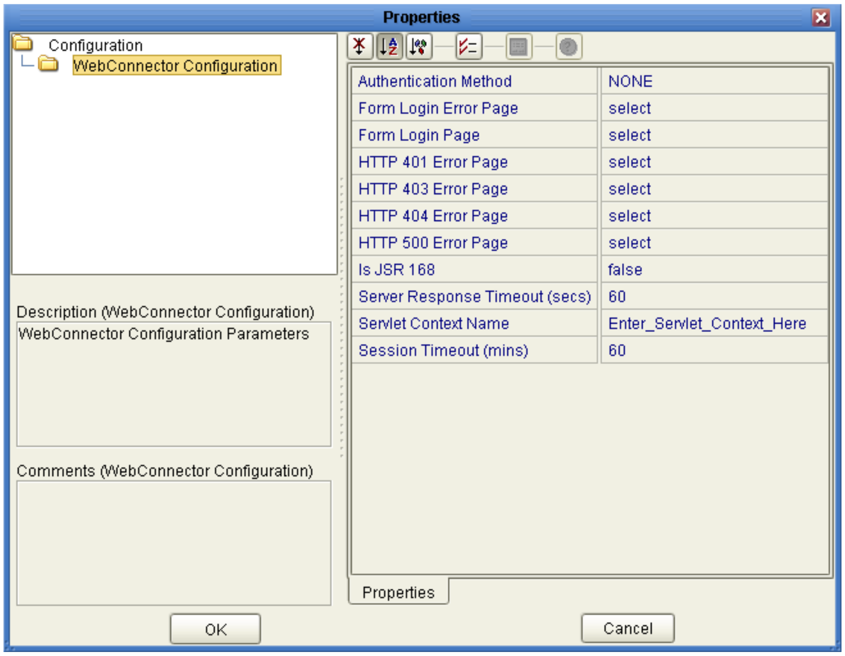
1. In the Project Explorer of Enterprise Designer, right-click the Connectivity Map and select **Open**.

   The Connectivity Map Editor appears.

2. Open the service binding box and double-click the connection icon between the **WSPProvider** Implemented Service and the Web Connector.

   The **Properties** dialog box appears.

Figure 147 Web Connector Configuration Properties



Description (WebConnector Configuration)
WebConnector Configuration Parameters

Comments (WebConnector Configuration)

Authentication Method	NONE
Form Login Error Page	select
Form Login Page	select
HTTP 401 Error Page	select
HTTP 403 Error Page	select
HTTP 404 Error Page	select
HTTP 500 Error Page	select
Is JSR 168	false
Server Response Timeout (secs)	60
Servlet Context Name	Enter_Servlet_Context_Here
Session Timeout (mins)	60

3. If you want to use the default Java CAPS authentication, then set the **Authentication Method** property to **BASIC**.

4. If you want to use the preconfigured authentication pages, then do the following:

   A. Set the **Authentication Method** property to **FORM**.

   B. Set the **Form Login Error Page** property to the Login Error page that you created.

   C. Set the **Form Login Page** property to the Login page that you created.

   D. If you created an Access Denied Error page, then set the **HTTP 401 Error Page** and **HTTP 403 Error Page** properties to the Access Denied Error page.

5. If you created a No Such Resource Error page, then set the **HTTP 404 Error Page** property to the No Such Resource page.

6. If you created an Internal Server Error page, then set the **HTTP 500 Error Page** property to the Internal Server Error page.

Note: *To clear the value of a page property, select the value, press Delete, and press Enter.*

7 Click **OK**.

10.1.3 Specifying Users and Roles

The final step of implementing the default Java CAPS authentication or the preconfigured authentication pages is to specify which users and roles can access the application.

You must add the roles to the eVision External System. In addition, you must add the users and roles to the application server.

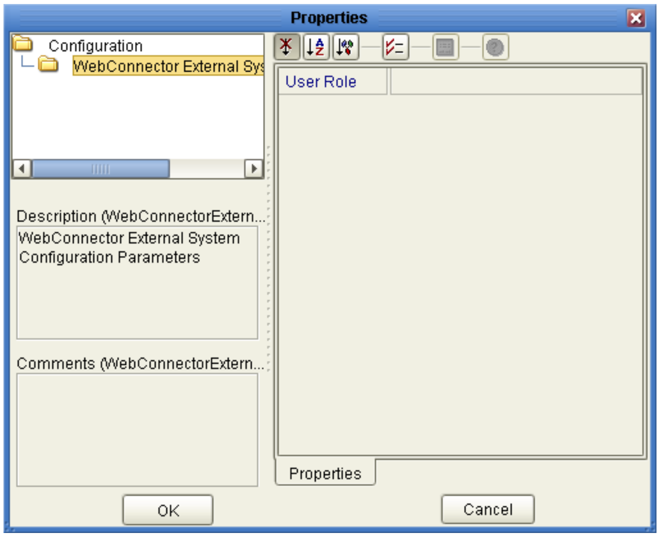
The following procedures assume that you have created an Environment for your application. The Environment must include an eVision External System.

To modify the eVision External System

1 In the Environment Explorer of Enterprise Designer, right-click the eVision External System and then click **Properties**.

The **Properties** dialog box appears.

Figure 148 eVision External System Properties



User Role

Description (WebConnectorExtern...

WebConnector External System Configuration Parameters

Comments (WebConnectorExtern...

2 Select the **User Role** property and click the Command button (**...**).

3 For each role that you want to have access to the application, do the following:

A Click **Add**. The **Input** dialog box appears.

B Enter the role name (for example, **analyst**).

C Click **OK**.

4 When you are done adding roles, click **OK** to return to the **Properties** dialog box.

5 Click **OK**.

To add users and roles to a Logical Host

The *Sun SeeBeyond eGate Integrator System Administration Guide* contains detailed information about how to add Logical Host users and roles by using Enterprise Manager.

To add users and roles to a WebLogic Server

Note: *For detailed information about how to perform the following steps, see the documentation provided with WebLogic Server.*

1. Log in to the WebLogic Server Administration Console.
2. While deploying the eVision Studio application, set the security model to **CustomRoles**.
3. Activate the changes.
4. Create a scoped role for the application. The name of the scoped role must match the role name in the Java CAPS Environment. Do not change the default value of the provider name.
5. Add a condition to the scoped role. The condition will be used to map users to the scoped role. For example, select the **Group** predicate and set the argument name to **eVision**.
6. Save the changes.
7. Go to the default security realm and add a group called **eVision**. Do not change the default value of the provider.
8. Add users to the **eVision** group.
9. Save the changes.
10. Start the application.

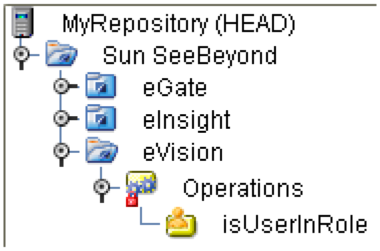
    All users who have **eVision** as one of the parent groups can now log in to the application.

10.2 Authorizing Users Based on Roles

You can use the **isUserInRole** operation to determine whether an authenticated user is authorized to view certain content, at the object level or page level.

The **isUserInRole** operation appears in the Project Explorer of Enterprise Designer. Expand the Sun SeeBeyond node, expand the eVision node, and then expand the Operations node.

Figure 149 isUserInRole Operation



You can drag the operation into a Page Flow or Business Process. The Business Rule Designer exposes the operation for input and output mappings:

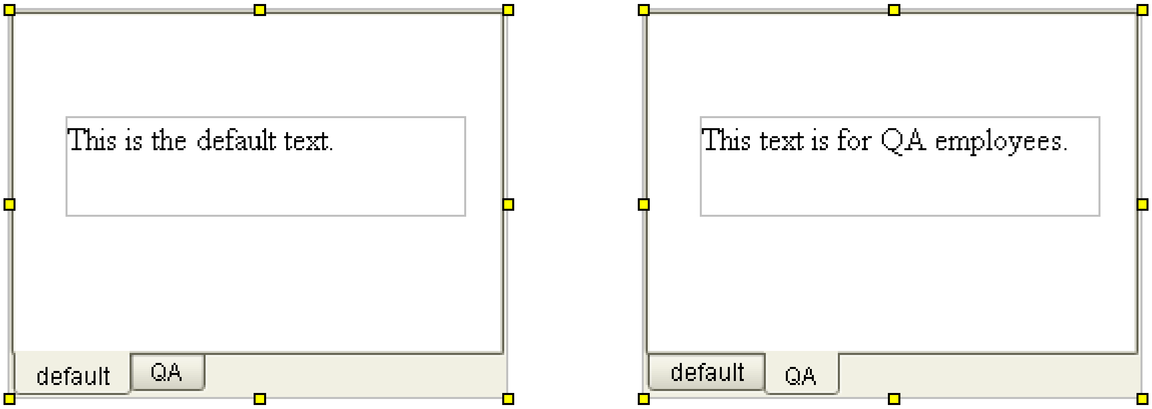
- The operation expects a role name as input. For example, you could pass in a string literal such as **Sales**. The string literal can consist of a Boolean expression, such as **Sales || Marketing** or **(Sales || Marketing) && Finance**.
- The operation produces a Boolean value as output. If the user has the specified role, then the value is **true**. If the user does not have the specified role, then the value is **false**.

10.2.1 Object Level

You can use the **isUserInRole** operation with the **If** and **Switch** objects to control which case appears.

For example, assume that you want to show different content at the top of a Page Layout, based on whether an employee is a member of the Quality Assurance (QA) department. Figure 150 illustrates the two cases of a **Switch** object. The **LName** property for the QA case is set to **QA**.

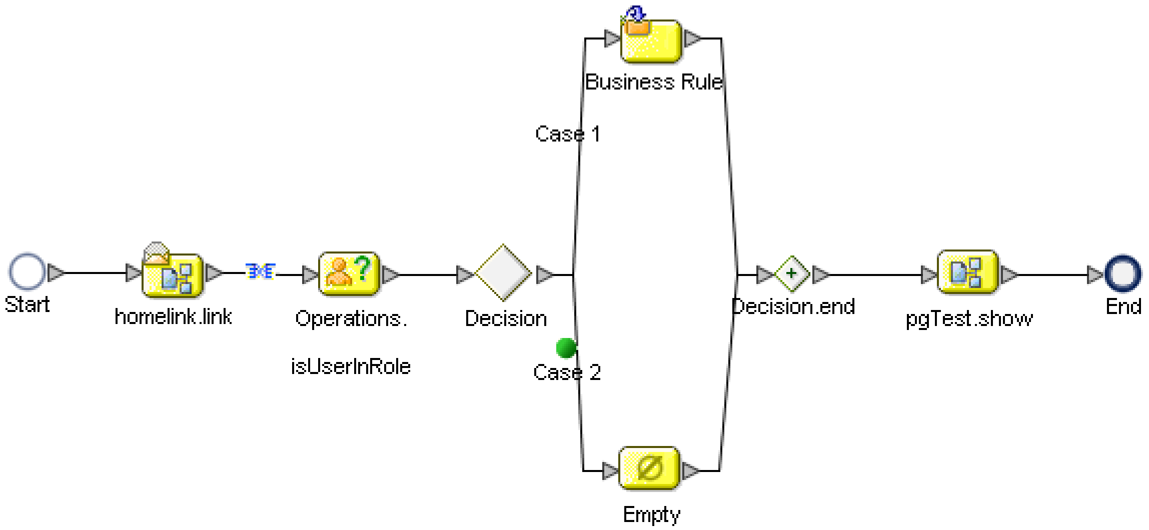
Figure 150 Switch Object with Two Cases


This is the default text.

This text is for QA employees.

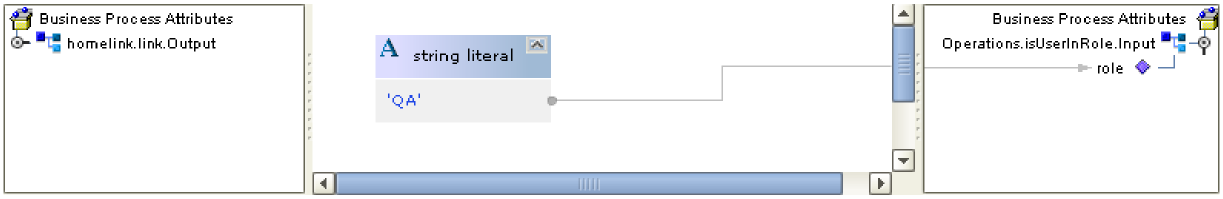
In the Page Flow, place the **isUserInRole** operation and a Decision Gate element before the Page Layout. Add an inline Business Rule before the operation. Add a Business Rule element and an Empty element to the Decision Gate.

Figure 151 Page Flow for Object Level Example



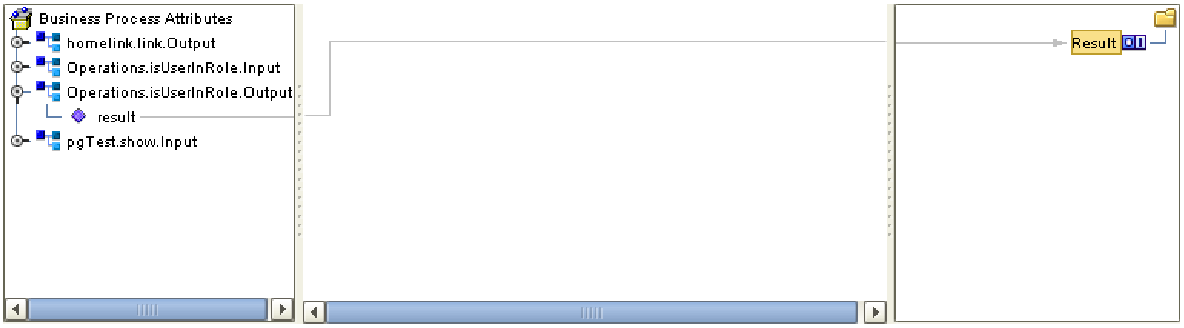
Select the inline Business Rule before the operation, create a string literal that contains the role name for QA employees, and then map the literal to the **isUserInRole** operation input.

Figure 152 Mapping the Role Name into the isUserInRole Operation


Business Process Attributes
homelink.link.Output
string literal
'QA'
Business Process Attributes
Operations.isUserInRole.Input
role

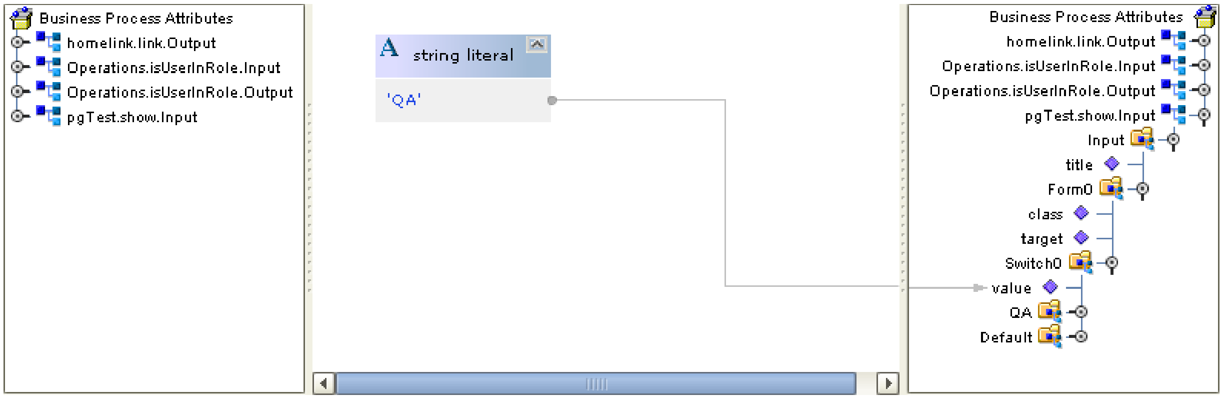
The output from the **isUserInRole** operation is a Boolean value. In the example, the value is **true** if the user has the QA role. Open the properties for the Decision Gate element and map the output to the **Result** node in the right pane.

Figure 153 Mapping the Output in the Decision Gate


Business Process Attributes
homelink.link.Output
Operations.isUserInRole.Input
Operations.isUserInRole.Output
result
pgTest.show.Input
Result

Select the Business Rule element in the Decision Gate, create a string literal that contains the role name for QA employees, and then map the literal into the QA case of the **Switch** object. This steps ensures that if the employee is a member of the QA department, then the **Switch** object displays the QA case.

Figure 154 Mapping the Role Name into the Switch Object


Business Process Attributes
homelink.link.Output
Operations.isUserInRole.Input
Operations.isUserInRole.Output
pgTest.show.Input
string literal
'QA'
Business Process Attributes
homelink.link.Output
Operations.isUserInRole.Input
Operations.isUserInRole.Output
pgTest.show.Input
Input
title
Form0
class
target
Switch0
value
QA
Default

10.2.2 Page Level

You can use the **isUserInRole** operation to control which Page Layout is displayed to a user.

For example, assume that you want to display Page Layout A to employees who are not in the QA department, and Page Layout B to employees who are in the QA department.

In the Page Flow or Business Process, create a **Decision Gate** element that contains two paths, one for Page Layout A and one for Page Layout B.

Place the **isUserInRole** operation before the **Decision Gate** element. Add an inline Business Rule before the operation. Create a string literal that contains the role name for QA employees and then map the literal to the operation input.

The output is a Boolean value. In the example, the value is **true** if the user has the QA role. Use the **equal** and **string literal** methods to determine whether the output is **true**, and map the Boolean result into the QA case of the **Switch** object.

10.3 Single Sign-On

eVision Studio supports single-sign on with Sun Java System Portal Server.

In the Connectivity Map Editor of Enterprise Designer, you must open the Web Connector configuration properties and set the **Authentication Method** property to **SSO Sun Java System Access Manager J2EE Agent Filter**.

In addition, the J2EE Agent in Sun Java™ System Access Manager Policy Agent 2.2 must be installed on the Sun Java™ System Application Server. Be sure to set the filter mode of the installation to **J2EE_POLICY** or **ALL**.

Chapter 11

eVision Studio Tutorial

This tutorial guides you through various tasks in the creation and deployment of an eVision Studio web application.

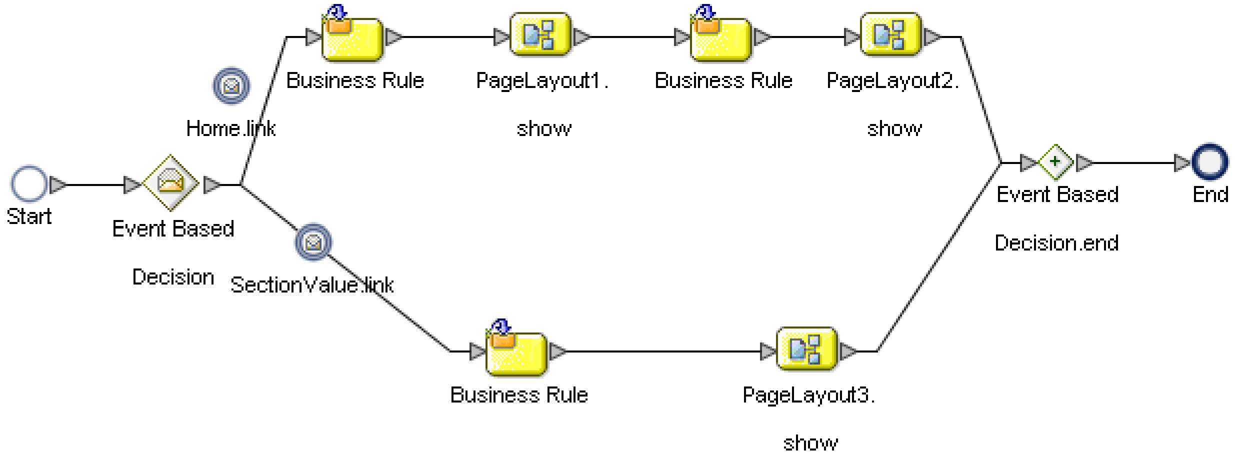
What's in This Chapter

11.1 eVision Studio Tutorial Overview

The sample project contains a reporting application for a supermarket.

Figure 155 shows the Page Flow in the sample project.

Figure 155 Page Flow in Sample Project



The Page Flow contains an **Event Based Decision** element that has two paths. The upper path is associated with a Page Link that has been designated as the home page. Therefore, the upper path is always executed first.

Table 29 describes the Page Layouts that appear in the Page Flow. **"Access the Web Application" on page 255** shows how these Page Layouts appear at runtime.

Table 29 Page Layouts in Page Flow

Page Layout	Description
PageLayout1	Enables the user to specify the food items and sales for the weekly sales report, as well as properties of the chart.
PageLayout2	Displays the weekly sales report as a table and a chart.
PageLayout3	Displays a detail view of the weekly sales report.

The **Template** Page Layout serves as a template for the **PageLayout1**, **PageLayout2**, and **PageLayout3** Page Layouts. The **Template** Page Layout does not appear in the Page Flow.

The sample project includes the following authentication and error-handling Page Layouts: **AccessDeniedError**, **InternalServerError**, **LoginError**, **LoginPage**, and **NoSuchResourceError**. These Page Layouts do not appear in the Page Flow.

Table 30 describes the Page Links that appear in the Page Flow.

Table 30 Page Links in Page Flow

Page Link	Description
Home	Points to the home page.
SectionValue	Contains the **section** and **value** clickability parameters.

The Connectivity Map has been preconfigured. The **Servlet Context Name** property is set to **eVisionTutorial**. Therefore, the URL that you will use to access the application at runtime has the following form:

```
http://hostname:portnumber/eVisionTutorial
```

"Deploying Page Flows" on page 161 contains detailed information about the format of application URLs.

11.2 Initial Tasks

The tutorial assumes that you have uploaded the **eVision.sar** and **eVisionDocs.sar** files to the Repository. Chapter 3 **"Installing eVision Studio"** describes how to upload these files.

The initial tasks of the tutorial are:

- Download the sample project
- Import the sample project into the Repository

11.2.1 Download the Sample Project

You download the sample project from the Java CAPS Installer.

To download the sample project

1. Log in to the Java CAPS Installer.
2. Go to the **Documentation** page.
3. In the left pane, click the **Core Products** tab.
4. In the left pane, click **Sun SeeBeyond eVision Studio**.
5. In the right pane, click the icon to the right of **Sample Projects**.
6. Save the **eVision_Sample.zip** file.
7. Using an archive utility, open the **eVision_Sample.zip** file and extract the **eVisionTutorial.zip** file.

11.2.2 Import the Sample Project into the Repository

You import the sample project into the Repository by using Enterprise Designer.

To import the sample project into the Repository

1. Log in to Enterprise Designer.
2. In the Project Explorer, right-click the Repository and then click **Import Project**.

   The **Import Manager** dialog box appears.
3. Click **Browse**.
4. Select the **eVisionTutorial.zip** file and click **Open**.
5. Click **Import**.
6. If the **Missing APIs** dialog box appears, then click **Continue**.

   The project is imported.
7. Click **OK**, and then click **Close**.
8. In the Project Explorer, expand the **eVisionTutorial** Project.

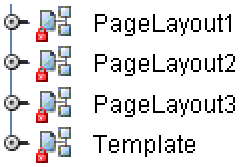
11.3 Page Layout Tasks

This section guides you through the following tasks in the Page Layout Designer:

- Change the background color of the Page Layout template
- Set property values for multiple selected objects
- Modify a local style

Notice that the icon for each Project component includes a red lock. The red lock indicates that the component is checked into the version control system.

Figure 156 Examples of Checked-In Components



PageLayout1
PageLayout2
PageLayout3
Template

During the following procedures, you check components out of and into the version control system. To perform these tasks, right-click the component, point to **Version Control**, and click **Check In** or **Check Out**.

The *Sun SeeBeyond eGate Integrator User's Guide* contains detailed information about the version control system.

11.3.1 Change the Background Color of the Page Layout Template

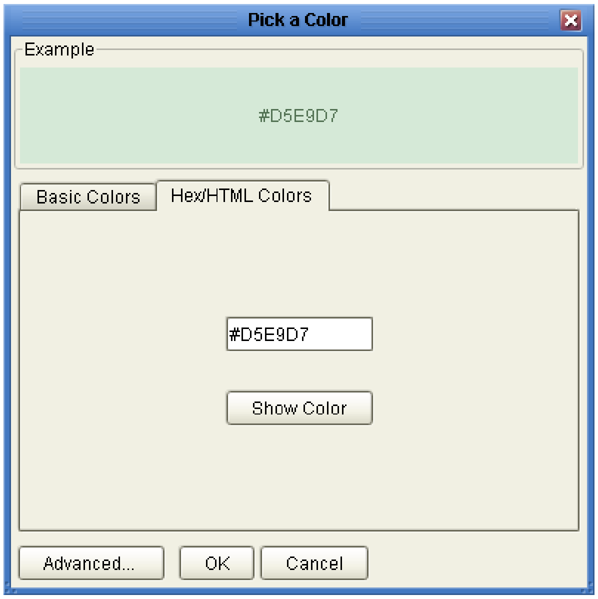
The default background color of the **Template** Page Layout is the hexadecimal value **#F0FFFF**. In this procedure, you change the color. The Page Layouts that use the **Template** Page Layout as a template will inherit the new background color.

To change the background color of the Page Layout template

1. Check the **Template** Page Layout out of the version control system.
2. Double-click the **Template** Page Layout to open it.
3. In the left column of the **Properties** tab, select the **Background Color** property.
4. In the right column of the **Properties** tab, click the **Command** button (**...**).

   The **Pick a Color** dialog box appears.
5. Click the **Hex/HTML Colors** tab.
6. Change the value to **#D5E9D7**.
7. If you want to preview the color, then click **Show Color**.

Figure 157 Previewing the Color

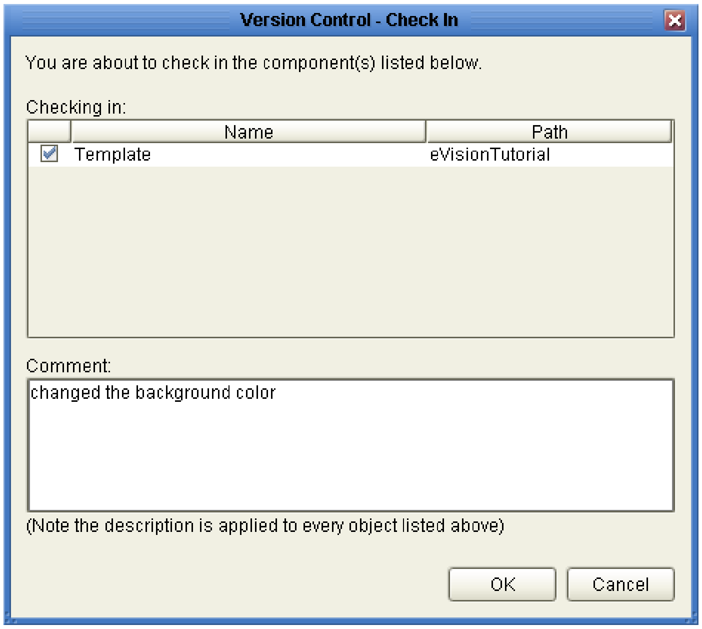


8 Click **OK**.

9 On the **File** menu, click **Save All**.

10 Check the **Template** Page Layout into the version control system. Be sure to enter a comment such as **changed the background color**.

Figure 158 Checking In the Page Layout



You are about to check in the component(s) listed below.

Checking in:

	Name	Path
☑	Template	eVisionTutorial

Comment: changed the background color

(Note the description is applied to every object listed above)

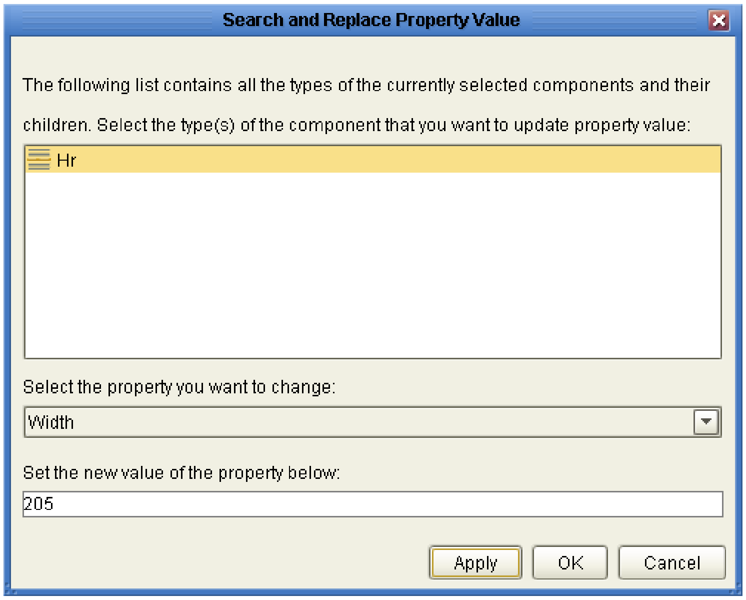
11.3.2 Set Property Values for Multiple Selected Objects

The Page Layout Designer allows you to set property values for multiple selected objects at the same time. In this procedure, you increase the width of the two horizontal lines in the **PageLayout1** Page Layout.

To set property values for multiple selected objects

1. Check the **PageLayout1** Page Layout out of the version control system.
2. Double-click the **PageLayout1** Page Layout to open it.
3. Press the Shift key and select the two horizontal lines.
4. On the Page Layout Designer toolbar, click the **Replace Properties** icon.
5. Select the **Width** property from the drop-down list. This property specifies the width in pixels.
6. Change the value from 200 to 205.

Figure 159 Increasing the Width of Horizontal Lines



Search and Replace Property Value

The following list contains all the types of the currently selected components and their children. Select the type(s) of the component that you want to update property value:

Hr

Select the property you want to change:

Width

Set the new value of the property below:

205

7. Click **Apply**, and then click **OK**.
8. On the **File** menu, click **Save All**.
9. Check the **PageLayout1** Page Layout into the version control system. Be sure to enter a comment such as **increased the width of horizontal lines**.

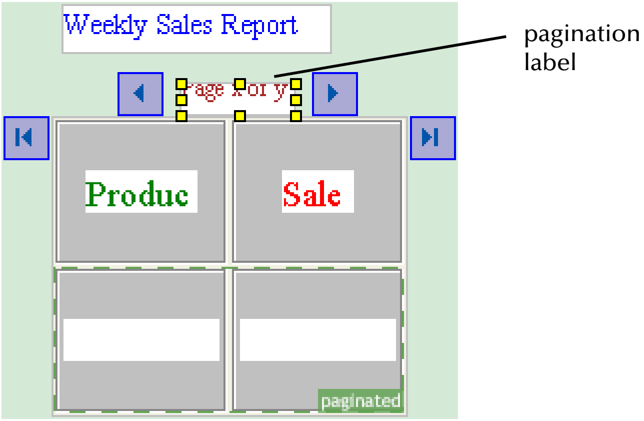
11.3.3 Modify a Local Style

eVision Studio uses the Cascading Style Sheets (CSS) standard. Local styles are applied to many objects in the sample project. In this procedure, you modify the color in a local style.

To modify a local style

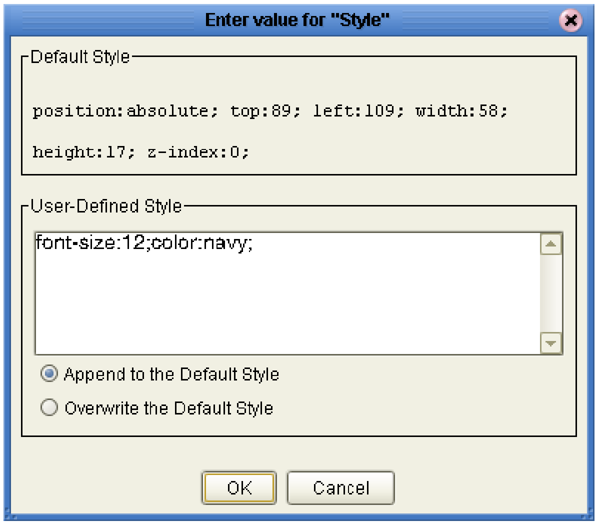
1. Check the **PageLayout2** Page Layout out of the version control system.
2. Double-click the **PageLayout2** Page Layout to open it.
3. Select the pagination label that appears above the dynamic table.

Figure 160 Selecting the Pagination Label



4. In the left column of the **Properties** tab, select the **Style** property.
5. In the right column of the **Properties** tab, click the **Command** button (**...**).
6. In the **User-Defined Style** field, change the color from brown to navy.

Figure 161 User-Defined Style



Enter value for "Style"

Default Style

```
position:absolute; top:89; left:109; width:58;

height:17; z-index:0;
```

User-Defined Style

```
font-size:12;color:navy;
```

● Append to the Default Style
○ Overwrite the Default Style

7. Click **OK**.
8. On the **File** menu, click **Save All**.
9. Check the **PageLayout2** Page Layout into the version control system. Be sure to enter a comment such as **changed the color of the pagination label**.

11.4 Page Flow Tasks

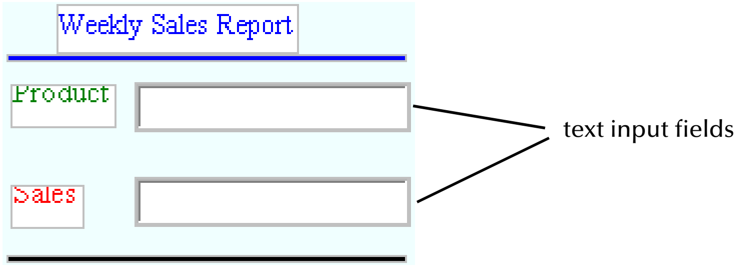
This section guides you through the following tasks in the Page Flow Designer:

- Configure a Business Rule
- Validate the Page Flow

11.4.1 Configure a Business Rule

The **PageLayout3** Page Layout contains a variety of objects, including two text input fields.

Figure 162 Text Input Fields in PageLayout3



The Business Rule in the lower path of the **Event Based Decision** element contains a mapping from the **SectionValue** Page Link to the **PageLayout3** Page Layout. The mapping sets the **Value** property of the text input field that appears to the right of the **Product** label.

In this procedure, you add a second mapping. The mapping sets the **Value** property of the text input field that appears to the right of the **Sales** label.

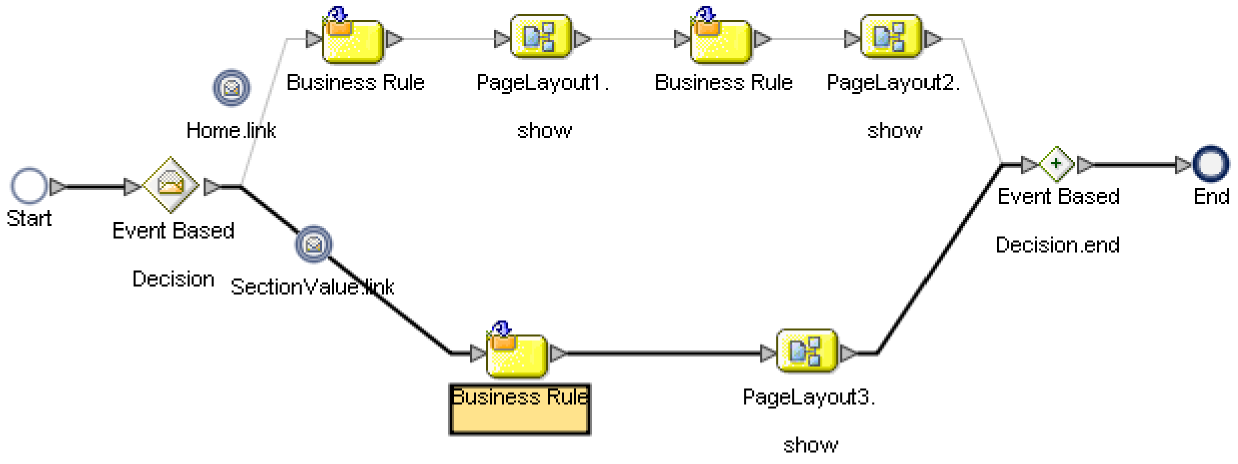
To configure a Business Rule

1. Check the **PageFlow1** Page Flow out of the version control system.
2. Double-click the Page Flow to open it.

Note: *If you want to display the background grid, then right-click the design canvas and select the* **Show Grid** *check box.*

3. Click the Business Rule in the lower path of the **Event Based Decision** element.

Figure 163 Business Rule in Lower Path of Event Based Decision

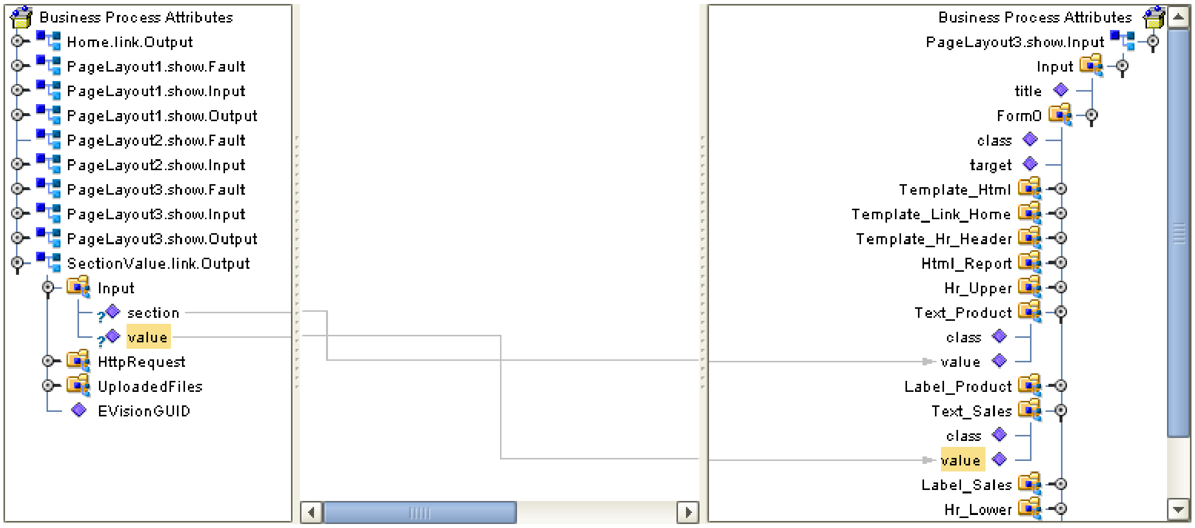


4 On the Page Flow Designer toolbar, click the **Display Business Rule Designer** icon.

5 In the right pane of the Business Rule Designer, expand the **Text_Sales** node. This node represents the text input field that appears to the right of the **Sales** label.

6 Create a mapping from the **Input.value** node in the left pane to the **Text_Sales.value** node in the right pane. Figure 164 shows how the mapping should appear.

Figure 164 Section and Value Node Mapping



7 On the **File** menu, click **Save All**.

11.4.2 Validate the Page Flow

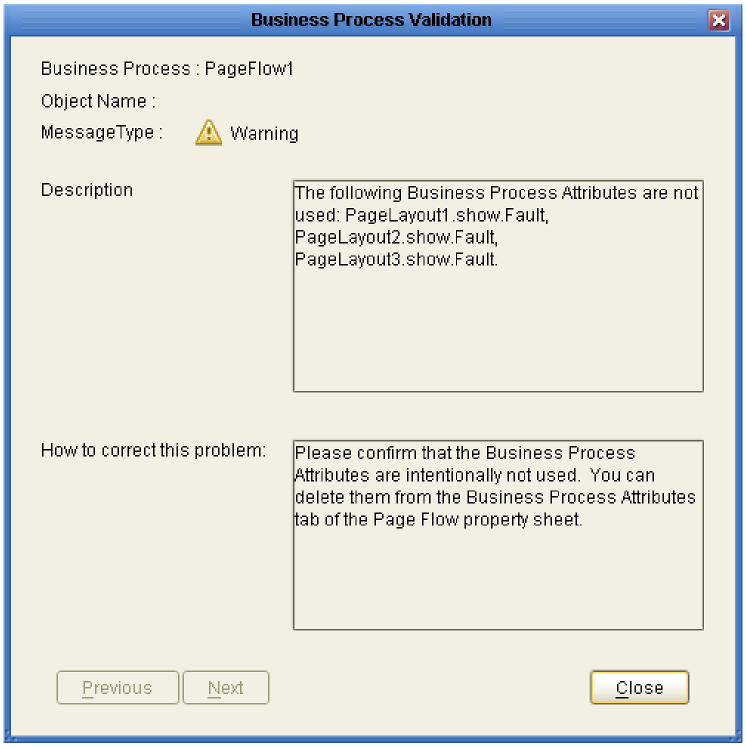
The validation feature enables you to check the Page Flow for various problems.

To validate the Page Flow

1 On the Page Flow Designer **toolbar, click the Validate Page Flow Model** icon.

The **Business Process Validation** dialog box appears.

Figure 165 Business Process Validation Dialog Box



Business Process Validation

Business Process : PageFlow1

Object Name :

MessageType : Warning

Description: The following Business Process Attributes are not used: PageLayout1.show.Fault, PageLayout2.show.Fault, PageLayout3.show.Fault.

How to correct this problem: Please confirm that the Business Process Attributes are intentionally not used. You can delete them from the Business Process Attributes tab of the Page Flow property sheet.

The warning indicates that some of the attributes are not used. This situation will not cause the web application to fail. Therefore, you can ignore the warning.

2 Click **Close**.

3 On the **File** menu, click **Save All**.

4 Check the **PageFlow1** Page Flow into the version control system. Be sure to enter a comment such as **added mapping to Sales input field**.

11.5 Deployment Tasks

The deployment tasks are:

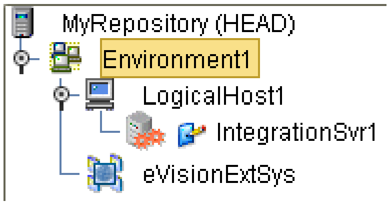
- Create the Environment
- Create the Deployment Profile
- Build the EAR file
- Deploy the EAR file

The following instructions assume that you are deploying to the Sun SeeBeyond Integration Server. For information about deploying to other supported application servers, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

11.5.1 Create the Environment

In this procedure, you create the runtime environment for the eVision Studio web application.

Figure 166 Tutorial Environment

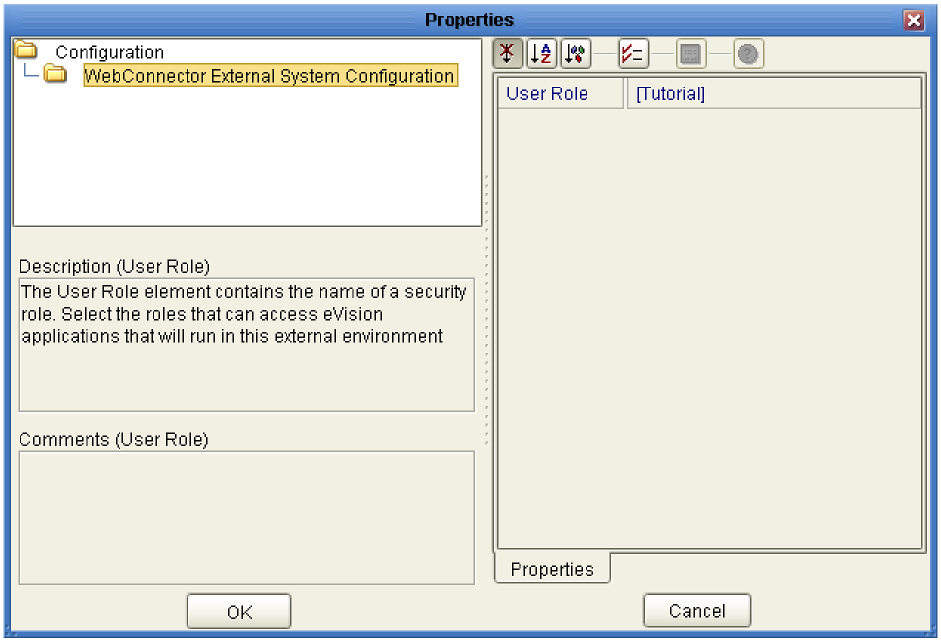


To create the Environment

1. On the **View** menu, click **Environment Explorer**.
2. Right-click the Repository. On the shortcut menu, click **New Environment**.
3. Right-click the Environment. On the shortcut menu, point to **New**, and then click **Logical Host**.
4. Right-click the Logical Host. On the shortcut menu, point to **New**, and then click **Sun SeeBeyond Integration Server**.
5. Right-click the Environment. On the shortcut menu, point to **New**, and then click **eVision External System**.
6. Type **eVisionExtSys** and click **OK**.
7. Right-click the eVision External System. On the shortcut menu, click **Properties**.

   The **Properties** dialog box appears.
8. Add a user role called **Tutorial**.

Figure 167 Adding the Tutorial User Role



Description (User Role)

The User Role element contains the name of a security role. Select the roles that can access eVision applications that will run in this external environment

Comments (User Role)

User Role	[Tutorial]

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

10 On the **File** menu, click **Save All**.

11.5.2 Create the Deployment Profile

You now create a Deployment Profile, which maps Project components to the Environment.

To create the Deployment Profile

1 On the **View** menu, click **Project Explorer**.

2 Right-click the **eVisionTutorial** Project.

3 On the shortcut menu, point to **New**, and then click **Deployment Profile**.

The **Create Deployment Profile for eVisionTutorial** dialog box appears.

4 Ensure that the **Environment** drop-down list is set to the Environment that you just created.

5 Click **OK**.

The Deployment Editor appears.

6 Click **Automap**.

The components in the left pane automatically move to the appropriate containers in the right pane, and the **Automap Results** dialog box appears.

7 Click **Close**.

8 On the **File** menu, click **Save All**.

11.5.3 Build the EAR File

You now build an EAR file, which is an application file that can be deployed to an application server.

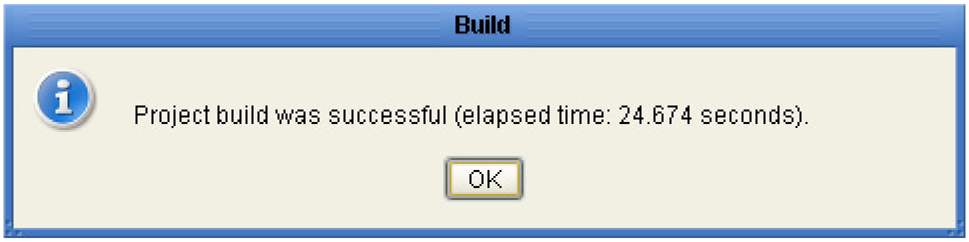
The EAR file is stored in the **Sun_JavaCAPS_install_dir\edesigner\builds** directory.

To build the EAR file

1 In the Deployment Editor, click **Build**.

The **Build** dialog box indicates that the build succeeded.

Figure 168 Build Dialog Box



Project build was successful (elapsed time: 24.674 seconds).

2 Click **OK**.

11.5.4 Deploy the EAR File

You use Enterprise Manager to deploy the EAR file to the Sun SeeBeyond Integration Server.

To deploy the EAR file

1 Start the Logical Host domain. You can use the **start-domain** script in the **Sun_JavaCAPS_install_dir\logicalhost** directory. On Windows platforms, you can also use the Domain Manager.

2 Log in to Enterprise Manager with a user name that has the **Deployment** role.

3 In the Explorer panel of Enterprise Manager, click **Deployer**.

4 In the Details panel of Enterprise Manager, click **Add Server**.

5 Ensure that the **Server Type** drop-down list is set to **Sun SeeBeyond Integration Server (version 5.1.2)**.

6 Specify the connection information for the Integration Server.

7 Click **Connect to Server**.

8 In the Explorer panel of Enterprise Manager, right-click the Integration Server and then click **Manage Integration Server Users**.

The User Management portion of the Integration Server Administration tool appears.

9 Add a user by performing the following steps:

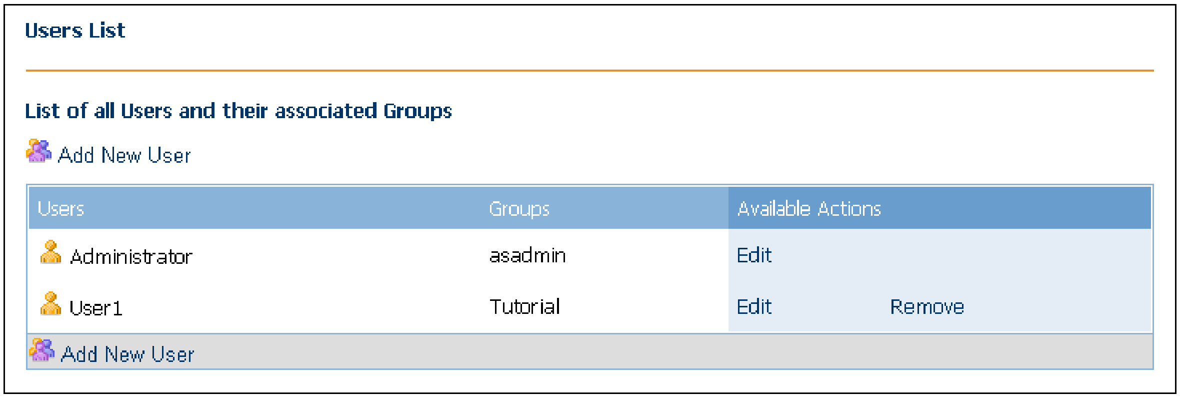
A Click **Add New User**.

B Enter information for the new user. In the **Group List** field, type **Tutorial**.

Note: *If you do not have sufficient privileges to add users, then ask the Enterprise Manager administrator to perform this task.*

C Click **Submit**.

Figure 169 New User with Tutorial Group



Users List

List of all Users and their associated Groups

Add New User

Users	Groups	Available Actions	
Administrator	asadmin	Edit	
User1	Tutorial	Edit	Remove

Add New User

D Log out of the Integration Server Administration tool.

10 In the Explorer panel of Enterprise Manager, click **Deployer**.

11 In the Details panel of Enterprise Manager, click the **Deploy Applications** tab.

12 In the **Application File** field, specify the fully qualified name of the EAR file. For example:

```
C:\JavaCAPS51\edesigner\builds\eVisionTutorialDeployment1\LogicalH
ost1\IntegrationSvr1\eVisionTutorialDeployment1.ear
```

13 Select the check box in the **Deploy** column.

14 Select the check box in the **Enable** column.

15 Click **Deploy**.

The **Results** area indicates that the EAR file deployed successfully.

11.6 Access the Web Application

Now that you have built and deployed the EAR file, you can access the web application from a web browser.

To access the web application

1. Start a web browser.
2. Enter the application URL. For example:

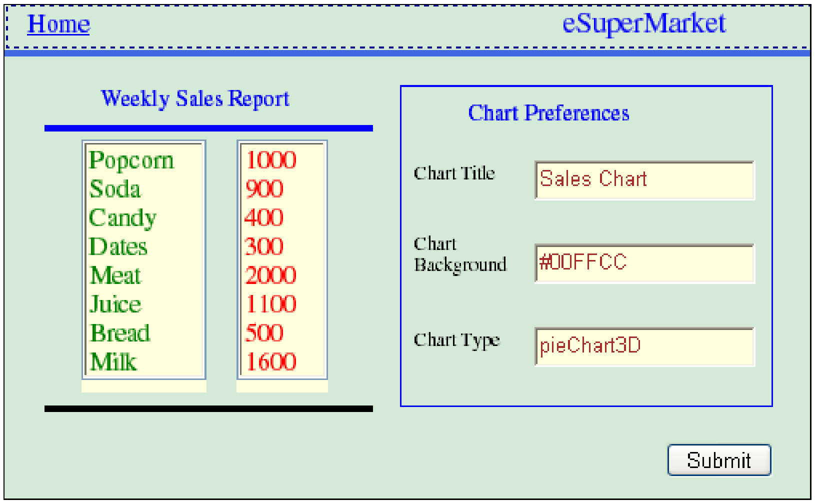
   ```
   http://server.company.com:18001/eVisionTutorial
   ```

   “Deploying Page Flows” on page 161 contains detailed information about the format of application URLs.

   The login page appears. This page corresponds to the **LoginPage** Page Layout.
3. In the **User ID** field, enter the user name that you created in “Deploy the EAR File” on page 253.
4. In the **Password** field, enter the corresponding password.
5. Click **Login**.

   The initial page appears. This page corresponds to the **PageLayout1** Page Layout.

Figure 170 Initial Page (PageLayout1)



Popcorn	1000
Soda	900
Candy	400
Dates	300
Meat	2000
Juice	1100
Bread	500
Milk	1600

Chart Preferences

Chart Title: Sales Chart

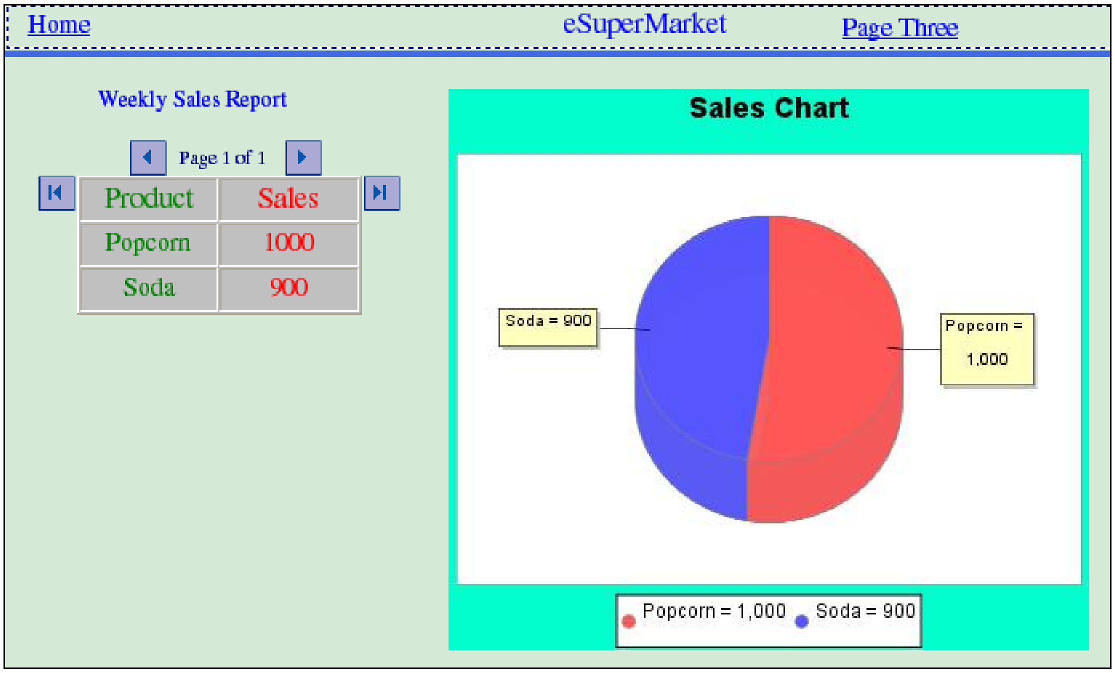
Chart Background: #00FFCC

Chart Type: pieChart3D

6. In the Weekly Sales Report area, select two products and two quantities.
7. Click **Submit**.

   The sales chart page appears. This page corresponds to the **PageLayout2** Page Layout.

Figure 171 Sales Chart Page (PageLayout2)

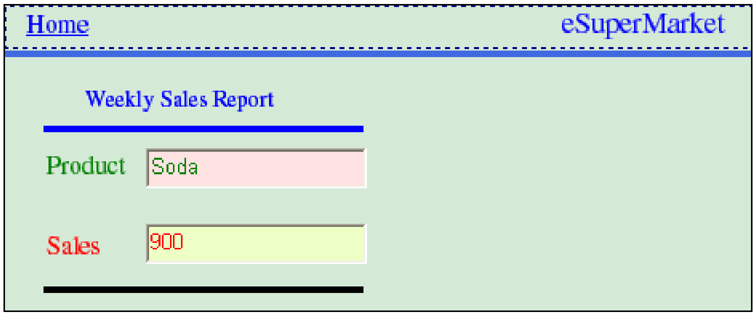


Product	Sales
Popcorn	1000
Soda	900

8 Click a segment in the pie chart.

A detail page appears. This page corresponds to the **PageLayout3** Page Layout.

Figure 172 Detail Page (PageLayout3)



Product: Soda

Sales: 900

9 Click the **Home** link.

The initial page appears.

Chapter 12

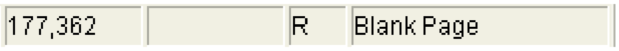
Troubleshooting

This chapter provides guidance for responding to various problems that you might encounter while using eVision Studio.

eVision Studio is not allowing me to make any changes to the Page Layout.

Check the status bar at the lower border of the design canvas. The status bar indicates whether the Page Layout is read/write or read only. You might need to check out the Page Layout from the version control system.

Figure 173 Status Bar for Page Layout in Read Only Mode



177,362 | | R | Blank Page

When I enter a value for the LName property, eVision Studio forces the initial letter to be capitalized.

This is not an error. eVision Studio requires that the first character of the **LName** property be capitalized.

When I try to install a custom object, an error message indicates that the custom.xml file cannot be found.

Ensure that the custom object files are located in the top-level directory of the .zip file.

Appendix A

Method Palette

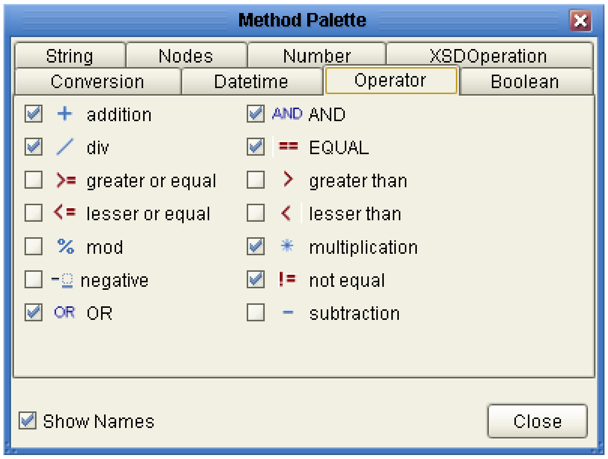
This appendix describes each method that appears in the Method Palette of the Business Rule Designer.

What's in This Appendix

A.1 Operators

Operators are the methods that allow you to manipulate data with standard mathematical operators.

Figure 174 Method Palette: Operator Tab



☑ + addition
☑ / div
□ >= greater or equal
□ <= lesser or equal
□ % mod
□ -□ negative
☑ OR OR

☑ AND AND
☑ == EQUAL
□ > greater than
□ < lesser than
☑ * multiplication
☑ != not equal
□ - subtraction

☑ Show Names

Table 31 Operator Methods

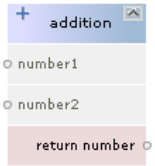
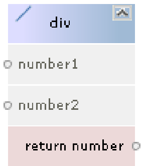
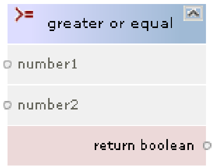
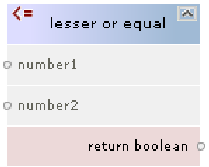
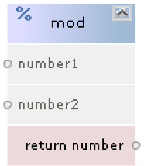
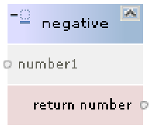
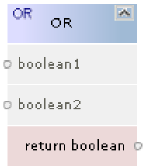
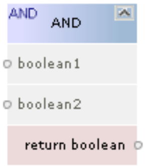
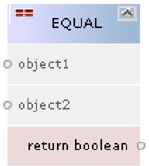
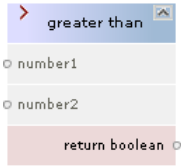
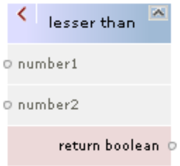
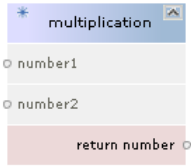
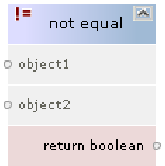
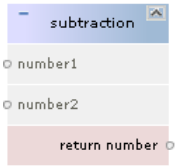
Symbol	Name	Function
	addition	Adds the value of *number1* to the value of *number2*, returns the sum.
	div	Divides the value of *number1* by the value of *number2*, returns the quotient.
 >= greater or equal number1 number2 return boolean	greater or equal	Returns Boolean true if *number1* is greater than or equal to *number2*; otherwise, returns Boolean false.
 <= lesser or equal number1 number2 return boolean	lesser or equal	Returns Boolean true if *number1* is less than or equal to *number2*; otherwise, returns Boolean false.
	mod	Used to divide two numbers and return only the remainder.
	negative	Converts the input number to negative. Result is a negative number having the same absolute value as the input number.
	OR	Returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, returns Boolean true.

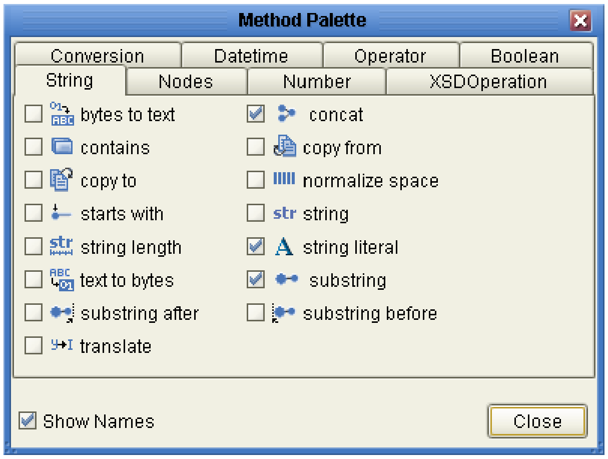
Table 31 Operator Methods (Continued)

Symbol	Name	Function
	AND	Returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, returns Boolean false.
	EQUAL	Returns Boolean true if *object1* is equal to *object2*; otherwise, returns Boolean false.
	greater than	Returns Boolean true if *number1* is greater than *number2*; otherwise, returns Boolean false.
	lesser than	Returns Boolean true if *number1* is less than *number2*; otherwise, returns Boolean false.
	multiplication	Multiplies the value of *number1* by the value of *number2*, returns the product.
	not equal	Returns Boolean true if *object1* is not equal to *object2*; otherwise, returns Boolean false.
	subtraction	Subtracts the numerical value of *number2* from the numerical value of *number1*, returns the difference.

A.2 String

The String methods allow you to manipulate string data.

Figure 175 Method Palette: String Tab



Method Palette

Conversion | Datetime | Operator | Boolean
String | Nodes | Number | XSDOperation

- □ bytes to text
- □ contains
- □ copy to
- □ starts with
- □ string length
- □ text to bytes
- □ substring after
- □ translate
- ☑ concat
- □ copy from
- □ normalize space
- □ string
- ☑ string literal
- ☑ substring
- □ substring before

☑ Show Names

Table 32 String Methods

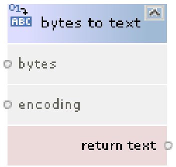
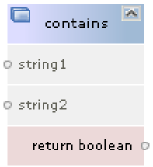
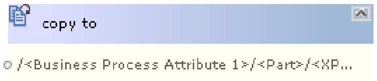
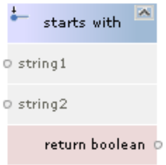
Symbol	Name	Function
	bytes to text	Decodes bytes into text using the specified encoding. If no encoding is specified, the platform's default encoding is used.
	contains	Returns true if the second string is contained within the first string, otherwise it returns false
	copy to	Allows you to type in the xpath expression for the destination of a copy operation. This is useful for entering xpath predicates. **Note:** This is for advanced users who are familiar with xpath and BPEL syntax.
	starts with	Returns true if the first string starts with the second string, otherwise it returns false

Table 32 String Methods (Continued)

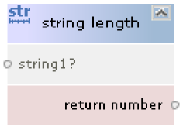
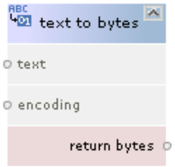
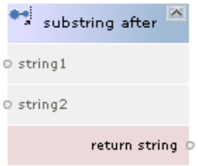
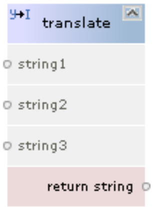
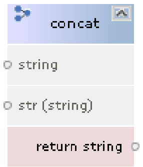
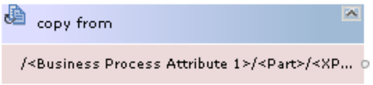
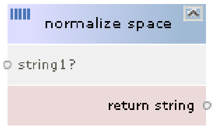
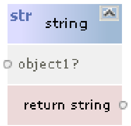
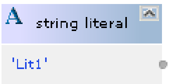
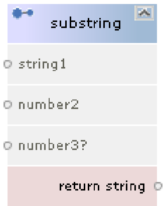
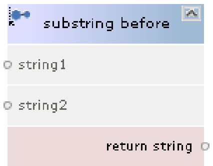
Symbol	Name	Function
	string length	Returns the number of characters in a string
	text to bytes	Encodes the input text into a sequence of bytes using the specified encoding. If no encoding is specified, the platform's default encoding is used
	substring after	Returns the part of the string in the string argument that occurs after the substring in the substring argument
	translate	Performs a character by character replacement. It looks in the value argument for characters contained in string1, and replaces each character for the one in the same position in the string2
	concat	Returns the concatenation of all its arguments **Note:** You can add mapping after mapping into the method. The method automatically adds an unmapped node as needed.
	copy from	Allows you to type in xpath expression for the source of a copy operation. This is useful for entering xpath predicates. **Note:** This is for advanced users who are familiar with xpath and BPEL syntax
	normalize space	Removes leading and trailing spaces from a string
	string	Converts the value argument to a string

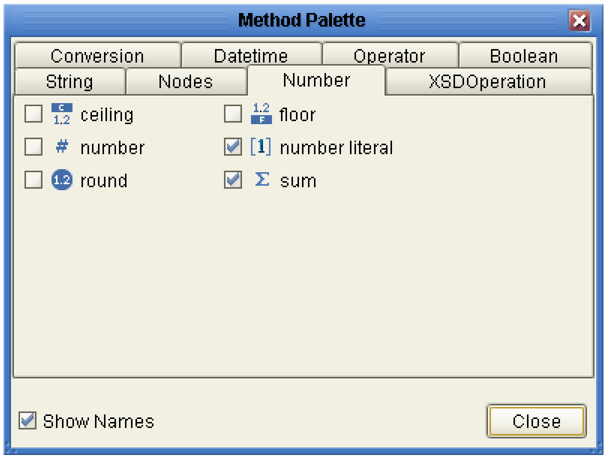
Table 32 String Methods (Continued)

Symbol	Name	Function
	string literal	A sequence of characters of fixed length and content
 substring string1 number2 number3? return string	substring	Returns a part of the string in the string argument
	substring before	Returns the part of the string in the string argument that occurs before the substring in the substring argument.

A.3 Number

The Number methods allow you to work with number data.

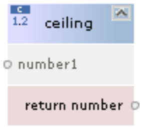
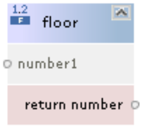
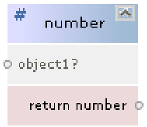
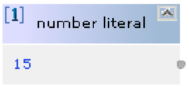
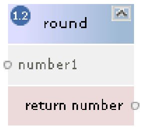
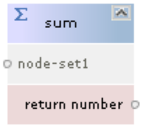
Figure 176 Method Palette: Number Tab



☐ ceiling
☐ floor
☐ number
☑ number literal
☐ round
☑ sum

☑ Show Names

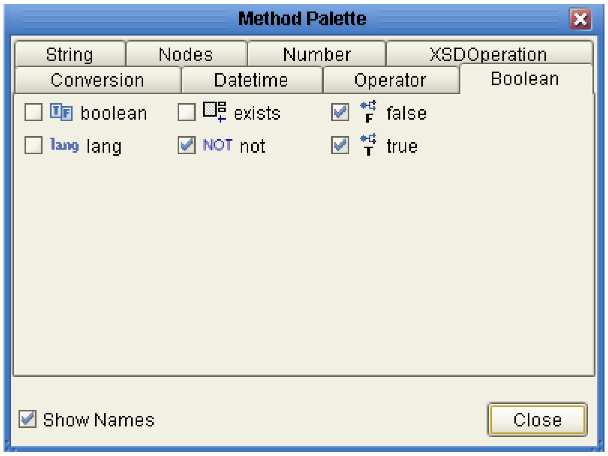
Table 33 Number Methods

Symbol	Name	Function
	ceiling	Returns the smallest integer that is not less than the number argument
	floor	Returns the largest integer that is not greater than the number argument
	number	Converts the value argument to a number
	number literal	A literal number string of fixed length and content
	round	Rounds the number argument to the nearest integer
	sum	Returns the total value of a set of numeric values in a node-set

A.4 Boolean

Boolean methods allow you to apply boolean logic to your data.

Figure 177 Method Palette: Boolean Tab



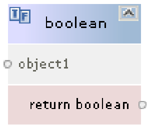
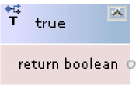
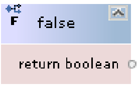
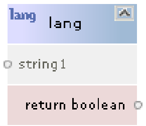
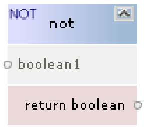
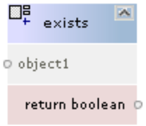
Method Palette ☒

String | Nodes | Number | XSDOperation
Conversion | Datetime | Operator | Boolean

☐ boolean ☐ exists ☑ false
☐ lang ☑ NOT not ☑ true

☑ Show Names Close

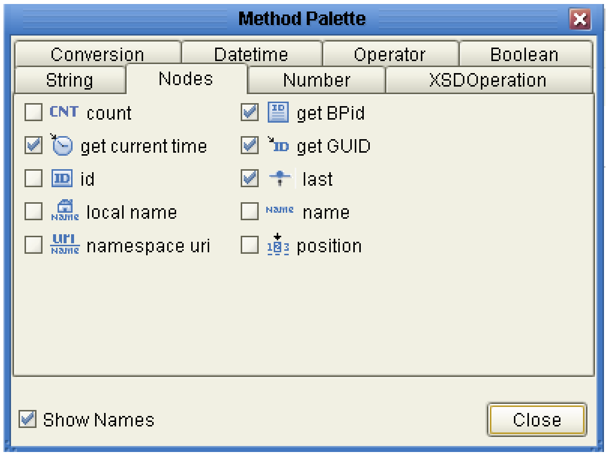
Table 34 Boolean Methods

Symbol	Name	Function
	boolean	Converts the value argument to Boolean and returns true or false.
	true	Returns true
	false	Returns false
	lang	Returns true if the language argument matches the language of the xsl:lang element, otherwise it returns false.
	not	Returns true if the condition argument is false, and false is the condition argument is true.
	exists	Checks to see if a value is present and returns a Boolean result.

A.5 Nodes

Node methods allow you to manipulate your data.

Figure 178 Method Palette: Nodes Tab



- ☐ CNT count
- ☑ get current time
- ☐ ID id
- ☐ local name
- ☐ namespace uri
- ☑ get BPid
- ☑ get GUID
- ☑ last
- ☐ name
- ☐ position

☑ Show Names

Table 35 Nodes Methods

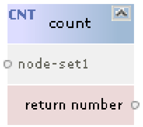
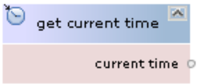
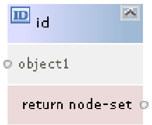
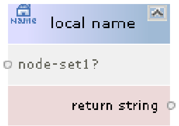
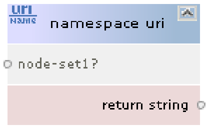
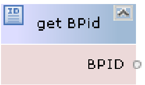
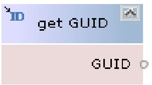
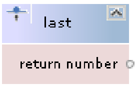
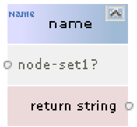
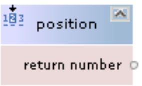
Symbol	Name	Function
	count	Returns the number of nodes in a node-set
	get current time	Gets the current time in ISO 8601 format (e.g. 2003-08-15T02:03:49.92Z).
	id	Selects elements by their unique ID
	local name	Returns the local part of a node. A node usually consists of a prefix, a colon, followed by the local name
	namespace uri	Returns the namespace URI of a specified node

Table 35 Nodes Methods (Continued)

Symbol	Name	Function
	get BPid	Gets the business process instance ID.
	get GUID	Gets a randomly generated globally unique ID.
	last	Returns the position number of the last node in the processed node list
	name	Returns the name of a node
	position	Returns the position in the node list of the node that is currently being processed

A.6 Datetime

Datetime methods allow you to manipulate date, time, and duration of data.

Figure 179 Method Palette: Datetime Tab

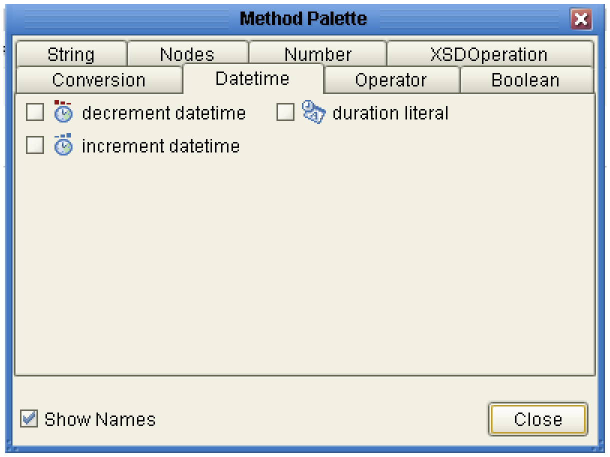
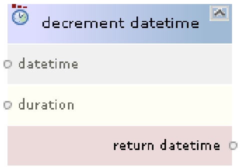
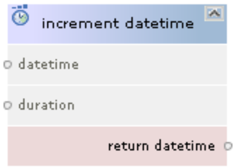
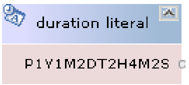


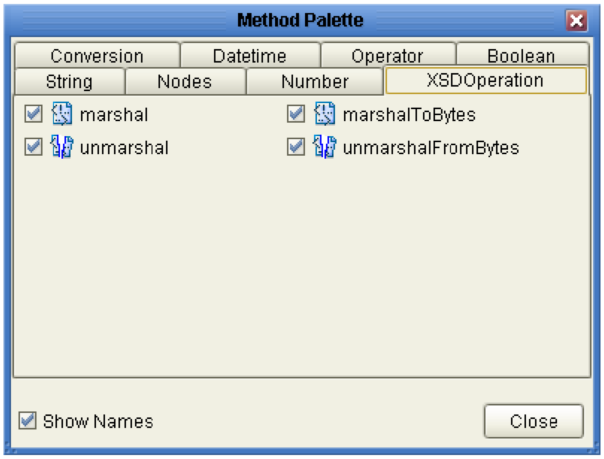
Table 36 Datetime Methods

Symbol	Name	Function
 decrement datetime datetime duration return datetime	decrement datetime	Dynamically decreases the date or time by a certain duration, such as days or hours.
 increment datetime datetime duration return datetime	increment datetime	Dynamically increases the date or time by a certain duration, such as days or hours.
	duration literal	Allows you to set an actual date or time.

A.7 XSD Operation

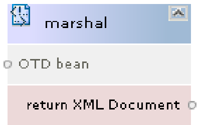
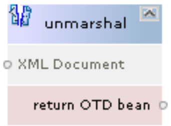
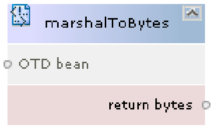
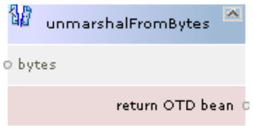
The XSD Operation methods enable you to marshal and unmarshal messages.

Figure 180 Method Palette: XSDOperation Tab



☑ marshal
☑ unmarshal
☑ marshalToBytes
☑ unmarshalFromBytes
☑ Show Names

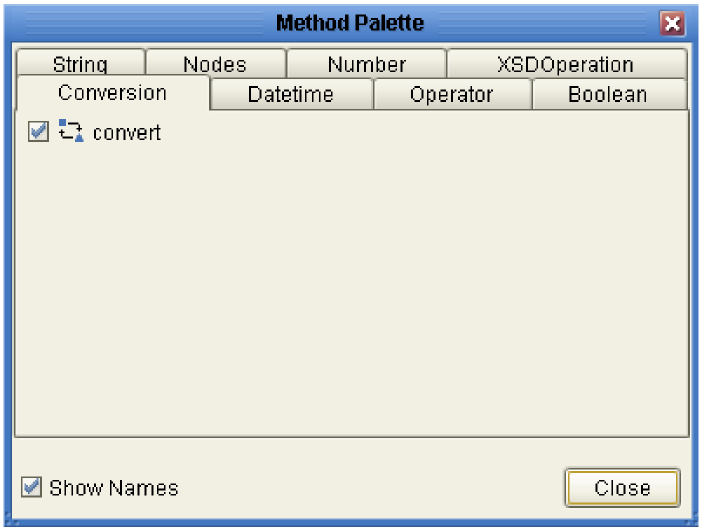
Table 37 XSDOperation Methods

Symbol	Name	Function
	marshal	Enables you to marshal messages. At runtime, the operator indicates that it needs to call the marshal function of the Object Type Definition (OTD).
	unmarshal	Enables you to unmarshal messages. At runtime, the operator indicates that it needs to call the unmarshal function of the OTD.
	marshalToBytes	Enables you to marshal an OTD bean into Bytes.
	unmarshalFromBytes	Enables you to unmarshal Bytes into an OTD bean.

A.8 Conversion

The Convert method allows you to make conversions from various data types.

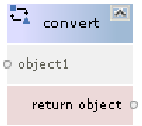
Figure 181 Method Palette: Conversion Tab



☑ convert

☑ Show Names

Table 38 Conversion Methods

Symbol	Name	Function
	convert	The convert function that takes in one input link and one output link. The convert function is limited to version 5.0.X Projects.

Appendix B

Conversational State

This appendix describes the support for conversational state in eVision Studio web applications.

What's in This Appendix

B.1 Conversational State Overview

HTTP is a stateless protocol, which means that Web applications must use some type of mechanism to maintain conversational state with clients. For example, the application might store a cookie on the user's computer.

In eVision Studio web applications, Event Based Decision elements have the ability to maintain conversational state. When an Event Based Decision element is accessed in a Page Flow, the element keeps track of which pages the user accesses. The user can leave a page and return to it as often as necessary.

This feature is particularly effective in promoting smooth page transitions when a user is moving from page to page in an unpredictable or non-sequential manner.

B.2 Example

The following Page Flow will be used to illustrate conversational state in eVision Studio. Because of the size of the Page Flow, it is shown in two figures. The right boundary of Figure 182 continues to the left boundary of Figure 183. The letters A and B and the numbers 1 through 13 are used in the explanation that follows.

Figure 182 Page Flow Example - Part 1

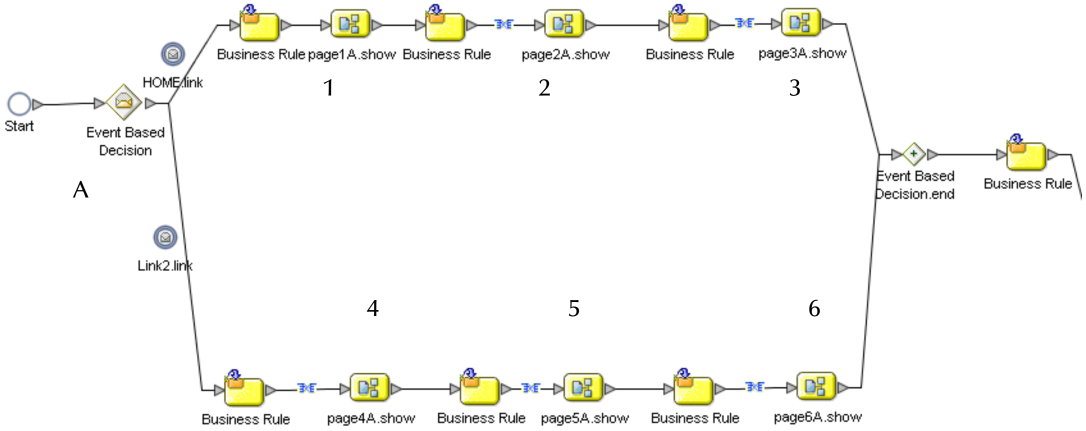
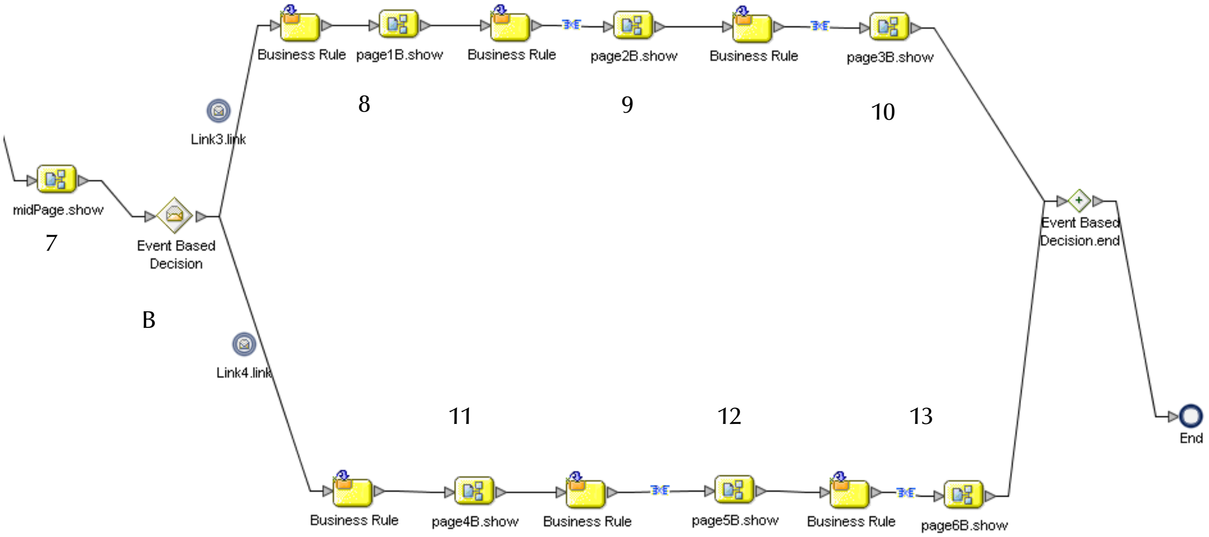


Figure 183 Page Flow Example - Part 2



The Page Flow contains two Event Based Decision elements:

- Event Based Decision A contains six pages (1 through 6).
- Event Based Decision B contains six pages (8 through 13).

Page 7 is located between the Event Based Decisions.

Once a user enters Event Based Decision A and accesses the home page, Event Based Decision A becomes “active” for the remainder of the Page Flow. If the user jumps from page 1 to page 2 and then clicks the Back button, the state of page 1 is preserved. Similarly, the user can jump from page 1 to page 2 to page 3 to page 2 to page 1.

Event Based Decision A must be active before Event Based Decision B can become active.

In addition, the user must access page 7 before Event Based Decision B can become active. For example, the user can jump from page 1 to page 2 to page 3 to page 7 to page 8. The user *cannot* jump from page 1 directly to page 8.

Once a user enters Event Based Decision B and accesses either link, Event Based Decision B becomes active for the remainder of the Page Flow. As with Event Based Decision A, the user can click the Back button and the state of previous pages are preserved. For example, the following sequence is valid:

- 1 - 2 - 3 - 7 - 8 - 9 - 10 - 9 - 8 - 7 - 3 - 2 - 1 - 2 - 3 - 7 - 11 - 12 - 13

Index