SeeBeyond ICAN Suite

eVision Studio User's Guide

Release 5.0.3

SEEBEYOND®

Version 20040303094831.

Contents

Chapter 4

Chapter 5

Chapter 7

Chapter 8

List of Figures

List of Tables

Chapter 1

Introduction

eVision Studio (eVision) is a graphical design studio for the creation of integrated Web applications. eVision provides graphical abstractions of backend data, and modeling of user/system interactions. With eVision, Web developers can rapidly create Web applications that can be deployed standalone, or as a channel in a portal without requiring any special programming expertise.

eVision is a component of the SeeBeyond® Integrated Composite Application Network (ICAN) Suite. The eVision component opens a real-time, interactive window into the ICAN Suite run-time environment. eVision allows the organization to present a single, unified view of enterprise data and applications to employees, customers, and partners.

A *Page Flow* is a series of Web pages that are laid out in a sequence to accomplish a specific group of tasks. User actions determine how the flow moves from page to page. Web applications enable the distribution of integrated Page Flows across the enterprise and allow real-time user interaction with those processes.

eVision Web applications receive and send data to ICAN Suite components, allowing users to interact with that data at run-time. Using eVision's rich set of Web development tools, the Web developer can create personalized views of business information, allowing Web application users to interact with running Page Flows in real time, while working with only the information that they need to see.

1.1 Document Purpose and Scope

The *eVision Studio User's Guide* explains how to use the eVision Studio application to create and deploy Web applications.

1.2 Organization of Information

The *eVision Studio User's Guide* includes the following information:

- A description of eVision Studio's user interface
- A description of eVision Studio's Page Layout Designer tools
- Instructions for creating sample Web pages
- Instructions for moving and resizing GUI components on the canvas
- A description of eVision Studio's Page Flow Designer tools
- Instructions for creating a Page Flow from pre-built components

1.3 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	**Blue** text	For example: See **"Online Documentation" on page 15**.

Additional Conventions

Windows Systems

For the purposes of this guide, references to "Windows" will apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

Path Name Separator

This guide uses the backslash ("\") as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

1.4 Online Documentation

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents. These documents are viewable with the Acrobat Reader application from Adobe Systems. Acrobat Reader can be downloaded from:

http://www.adobe.com

1.5 The SeeBeyond Web Site

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

http://www.seebeyond.com

Chapter 2

About eVision Studio

This chapter provides an overview of eVision Studio.

This Chapter Includes:

2.1 eVision Studio Web Applications

eVision Studio is a graphical design studio that allows the business analyst/Web developer to create Web applications. With eVision, the Web developer can rapidly create interactive Web applications, which can be deployed standalone or as a channel within a portal, without requiring advanced programming abilities.

eVision's Graphical User Interface (GUI) components gather input data from the user at run-time, and add functions and interactivity to Web pages. GUI components (called HTML and Form Objects) comprise familiar Web-centric design elements including check boxes, check box groups, text boxes, graphics containers, horizontal (separator) lines, and submit buttons. GUI components are pre-built combinations of Java classes and JSP code that represent Web interface elements. Component property sheets allow the developer to add the labeling, functionality, and appearance attributes that the components will display to users at run-time.

Through the use of familiar drag-and-drop techniques and property sheets, eVision allows the Web developer to interactively add graphics, text, and programmatic content to Web application pages. Pre-built components are dragged from convenient component palettes and positioned on the design canvas; functional and appearance attributes are added to the components in the properties window. As a Web page is designed, the developer can preview it in a browser at any time throughout the process.

Using eVision's design tools and pre-built components, the Web application developer can create personalized views of business information, so that users can influence the flow of business information through the browser.

For example, eVision Web applications can be structured to allow employees to log into a Web server, see the tasks that are assigned to them, then use the browser to complete their assignments. eVision opens a real-time, interactive window into the ICAN Suite run-time environment, presenting Web applications to users across the enterprise. eVision Web applications allow users to interact with business activities through the browser, and complete the business tasks that require real-time human analysis, judgement, and intervention.

2.2 ICAN Suite Integration

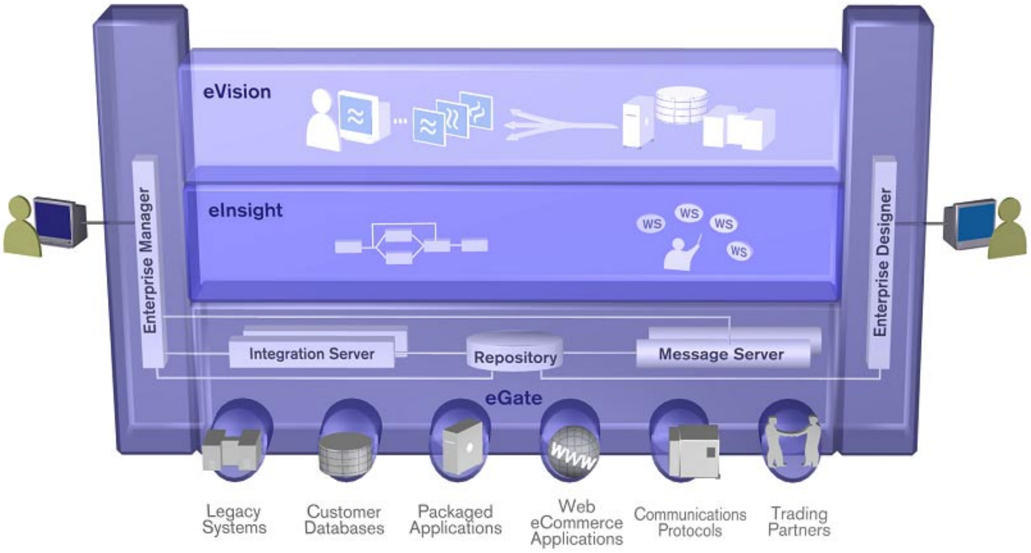
eVision is fully integrated with the ICAN Suite and runs as a component within the ICAN Suite run-time environment. The Page Layout Designer and Page Flow Designer run as components within the Enterprise Designer GUI. eVision Web applications are stored in the SeeBeyond Repository.

The ICAN Suite has a wide range of functions that it shares with all of the ICAN Suite products. eVision is able to leverage many platform level services, such as:

- Resource Management - The ICAN Suite uses a distributed and open architecture that enables components to access system resources (memory and processing power) as needed and in conjunction with other components.
- Security - The ICAN Suite provides a security module, to fulfill security needs such as authentication and authorization access to eVision functions.
- Repository storage and access - The setup, component, and configuration information for the elements of a Project, including related eVision components, are stored in the Repository.
- Deployment abilities - The ICAN Suite provides Deployment Profiles that contain all of the information necessary to activate eVision applications and associated components.
- Monitoring - Enterprise Manager provides Web-based monitoring abilities.
- Connectivity Mapping - The Connectivity Map maintains the relationships between eVision Web applications and other system components.
- Version Control - This feature allows users to maintain multiple versions of the same components and Page Flows, through a check-in and check-out process.
- Import and Export of eVision Web applications - The ability to import and export Web applications makes it possible to recreate Web applications on other systems.

Figure 1 shows an example of how eVision is integrated with the ICAN Suite.

Figure 1 eVision Studio Integration With the ICAN Suite

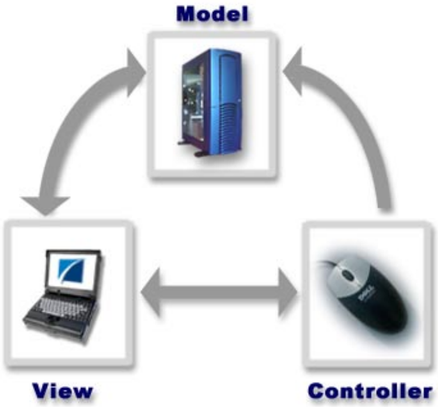


2.3 MVC Architecture

eVision applications are based on Model/View/Controller (MVC) architecture. MVC architecture 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 architecture 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.

MVC architecture consists of three types of objects, the Model, the View, and the Controller. See Figure 2.

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** (viewport) 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's application architecture fully supports the MVC paradigm. In an eVision 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 eVision **Page Flow**, contains the business logic (OTDs and Collaborations) that interacts with the back-end system applications.
- The **View** contains the JSP pages that are generated with eVision's **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 eVision Page Flow Engine

eVision *Pages* and *Page Flow*s are Web application elements that allow users to interact with backend systems to execute tasks that require human analysis and intervention. A typical example of human intervention in this context is credit approval based on a credit report. A Page Flow is typically engineered to guide the user through a page-by-page process of data viewing and task execution.

At run-time, Page Flows are evaluated and are then assigned to the proper group, user, or role. The assignee uses the eVision Web application (a Page or Page Flow) to finish the required tasks.

The **eVision Page Flow Engine** orchestrates the system responses to the execution of Web page component code. At run-time, 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 eVision Web Application Design Components

The main components of eVision Studio are the Page Layout Designer (the Web page designer's toolkit) and the Page Flow Designer (the Page Flow layout, connection, and mapping tool).

2.5.1 Page Layout Designer

The Page Layout Designer allows the Web designer to specify the details of individual pages in the application using familiar drag-and-drop techniques to place GUI components (Web page design objects) onto a blank page design canvas. The Page Layout Designer provides the user with a comprehensive collection of these pre-built GUI components, and a friendly, WYSIWYG Web page editor. **"Design Tools" on page 28** describes the Page Layout Designer components in detail.

2.5.2 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 Web page to Web page.

The Page Flow Designer allows the business analyst 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*). The business analyst can easily lay out a Web application's page flow using the Page Flow Designer's set of intuitive, graphical modeling tools. At run-time, the page flow drives the display of eVision Web pages and orchestrates the interactions with backend systems.

Through the deployment of Web pages that are developed with eVision's Page Layout and the Page Flow Designer development tools, the business analyst can enable enterprise-wide, internal and external access to Web applications across an intranet, or the Internet.

2.6 UTF-8 Support

eVision Studio provides support for the UTF-8 encoding scheme during both design time and run time. Therefore, an eVision Web application can include any character set that is supported by UTF-8.

Chapter 3

Installing eVision Studio

This chapter describes how to install eVision Studio.

3.1 System Requirements

eVision Studio is available on the following operating systems:

- Windows Server 2003, Windows XP SP1a, and Windows 2000 SP3
- HP Tru64 V5.1A with required patches
- HP-UX 11.0 and 11i with required patches and parameter changes
- IBM AIX 5.1 and 5.2 with required maintenance-level patches
- Red Hat Enterprise Linux Advanced Server 2.1
- Red Hat Linux 8 (Intel Version)
- Sun Solaris 8 and 9 with required patches

The **Readme.txt** file on the CD-ROM contains the most up-to-date operating system requirements for the supported platforms.

For more information on platform support, see the *SeeBeyond ICAN Suite Installation Guide*.

3.1.1. Database Support

If desired, you can persist eVision Page Flows using a database. eVision supports the following databases:

- Oracle 8.1.7, 9.1, and 9.2
- Sybase 12.5
- Microsoft SQL Server 2000

3.2 Installation Steps

The installation procedure for eVision Studio is similar to the installation procedure for other ICAN Suite products. You can find general product installation instructions in the *SeeBeyond ICAN Suite Installation Guide*, which is available from Enterprise Manager's Documentation page.

3.2.1. Before You Begin

- The ICAN Repository server must be running.
- The client computer that you are installing on must have Enterprise Designer already installed on it.

3.2.2. Uploading eVision Studio to the Repository

The following procedure must be performed by the Administrator user, or by a user that has been granted a role with upload privileges.

To upload eVision Studio to the Repository

1. Start Internet Explorer.
2. In the **Address** field, enter **http://*hostname*:*portnumber***

   where

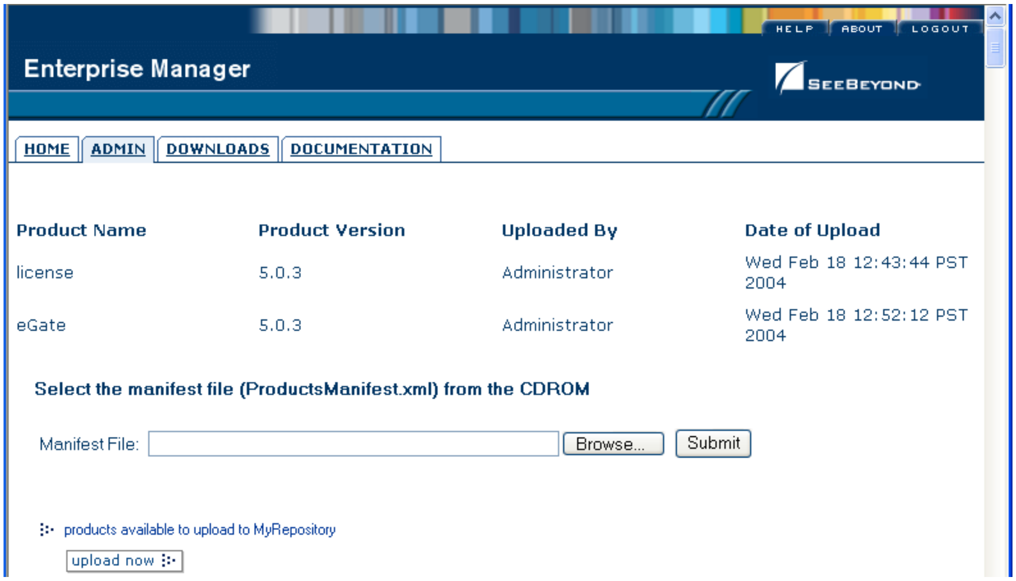
   hostname is the TCP/IP host name of the server where the Repository is installed.

   portnumber is the base port number of the Repository.

   The **SeeBeyond Customer Login** window of Enterprise Manager appears.
3. Enter your username and password.
4. Click **Login**.

   The Enterprise Manager home page appears.
5. Click the **ADMIN** tab. See Figure 3.

Figure 3 Enterprise Manager ADMIN Page



Product Name	Product Version	Uploaded By	Date of Upload
license	5.0.3	Administrator	Wed Feb 18 12:43:44 PST 2004
eGate	5.0.3	Administrator	Wed Feb 18 12:52:12 PST 2004

Select the manifest file (ProductsManifest.xml) from the CDROM

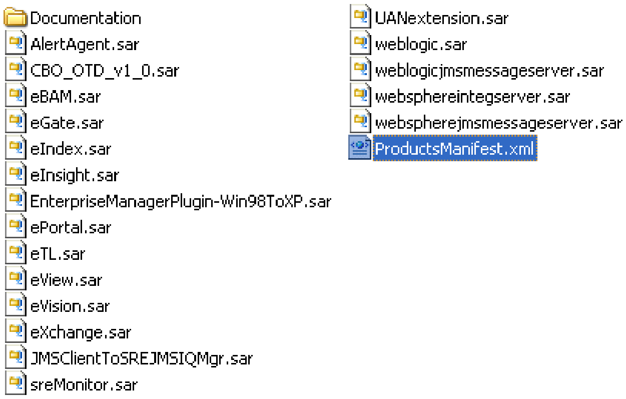
Manifest File:

products available to upload to MyRepository

6 Click **Browse**. The **Choose file** dialog box appears.

7 Select the **ProductsManifest.xml** file in the top level of the Products - Disc 1 CD-ROM. See Figure 4.

Figure 4 Selecting the ProductsManifest.xml File



Documentation
AlertAgent.sar
CBO_OTD_v1_0.sar
eBAM.sar
eGate.sar
eIndex.sar
eInsight.sar
EnterpriseManagerPlugin-Win98ToXP.sar
ePortal.sar
eTL.sar
eView.sar
eVision.sar
eXchange.sar
JMSClientToSREJMSIQMgr.sar
sreMonitor.sar
UANextension.sar
weblogic.sar
weblogicjmsmessageserver.sar
websphereintegserver.sar
websphererjmsmessageserver.sar
ProductsManifest.xml

8 Click **Open**.

You are returned to the Enterprise Manager **ADMIN** page.

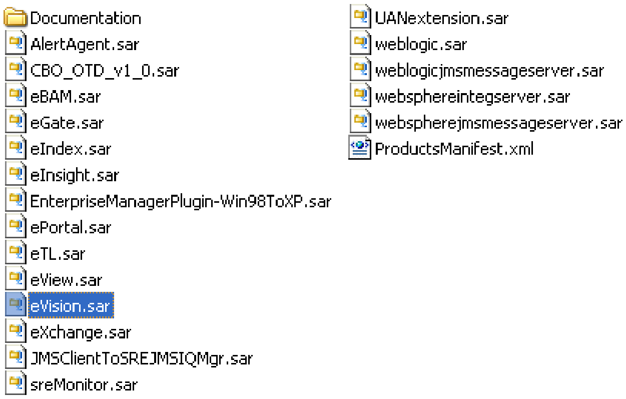
9 Click **Submit**.

The list of products available for uploading appears.

10 On the eVision row, click **Browse**. The **Choose file** dialog box appears.

11 Select the **eVision.sar** file. See Figure 5.

Figure 5 Selecting the eVision.sar File



Documentation
AlertAgent.sar
CBO_OTD_v1_0.sar
eBAM.sar
eGate.sar
eIndex.sar
eInsight.sar
EnterpriseManagerPlugin-Win98ToXP.sar
ePortal.sar
eTL.sar
eView.sar
eVision.sar
eXchange.sar
JMSClientToSREJMSIQMgr.sar
sreMonitor.sar
UANextension.sar
weblogic.sar
weblogicjmsmessageserver.sar
webspherintegserver.sar
webspherejmsmessageserver.sar
ProductsManifest.xml

12 Click **Open**.

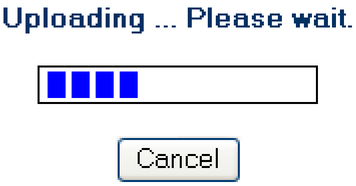
You are returned to the Enterprise Manager **ADMIN** page.

13 Click **upload now**. See Figure 6.

Figure 6 upload now Button

The progress bar indicates the status of the upload. See Figure 7.

Figure 7 Progress Bar for eVision Studio Upload



Uploading ... Please wait.

14 When the upload is finished, click **Back to component installation**.

15 To upload the eVision Studio documentation and sample files to the Repository, repeat steps 6 through 14 with the following differences:

- Use the **ProductsManifest.xml** file in the Documentation subdirectory.
- Upload the **eVisionDocs.sar** file in the Documentation subdirectory.

The documentation and sample files will be accessible from the **DOCUMENTATION** page.

3.2.3. Updating Enterprise Designer with eVision Modules

The following procedure adds the eVision modules to Enterprise Designer.

To update Enterprise Designer with eVision modules

1 On the client computer where Enterprise Designer is installed, start Enterprise Designer.

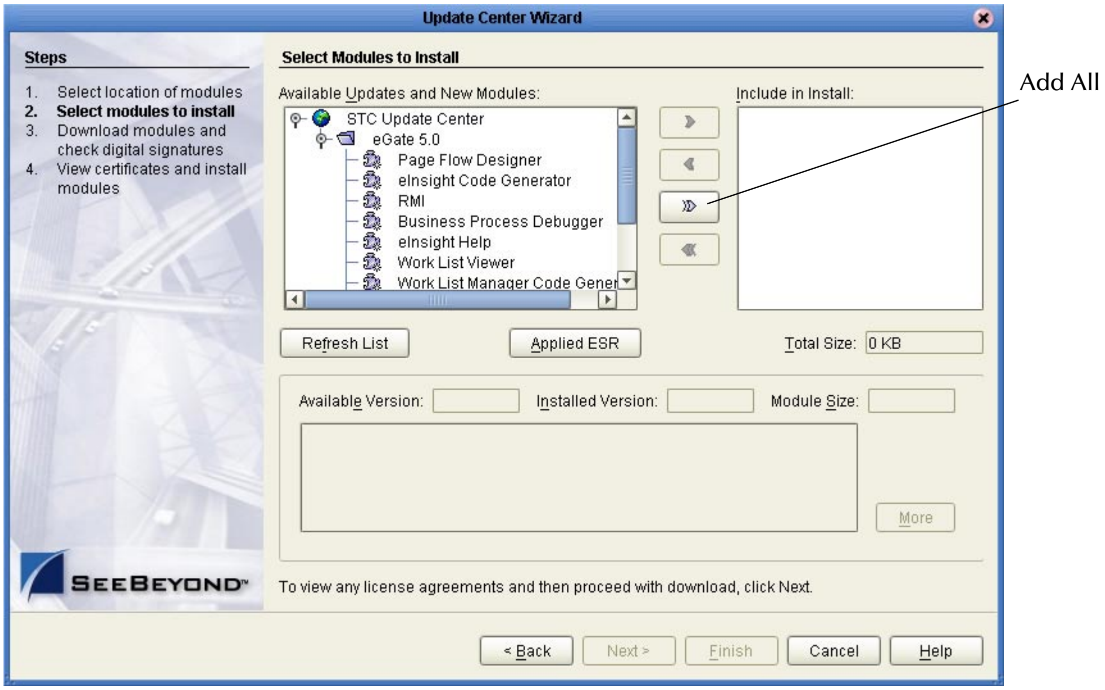
2 On the **Tools** menu, click **Update Center**.

The Update Center Wizard appears.

3 Click **Next**.

Step 2 appears. See Figure 8.

Figure 8 Update Center Wizard: Select Modules to Install



1. Select location of modules
2. **Select modules to install**
3. Download modules and check digital signatures
4. View certificates and install modules

To view any license agreements and then proceed with download, click Next.

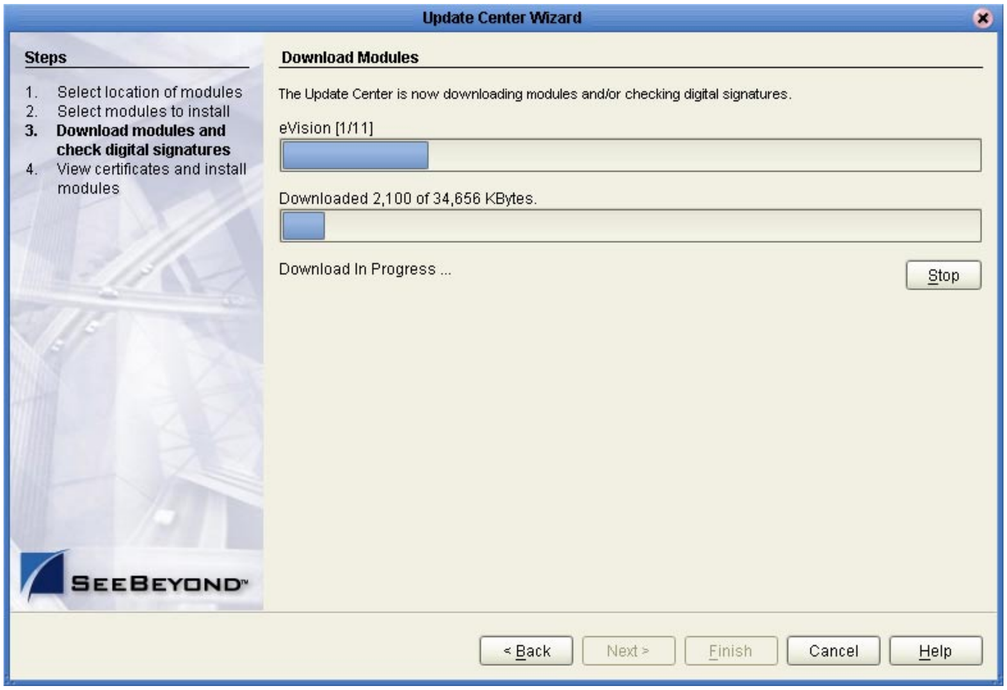
4 Click the **Add All** button. The eVision components move to the **Include in Install** box.

5 Click **Next**. The **License Agreement** window appears.

6 Click **Accept**.

The progress bars indicate the status of the download. See Figure 9.

Figure 9 Update Center Wizard: Download Modules


Update Center Wizard
Steps
1. Select location of modules
2. Select modules to install
3. Download modules and check digital signatures
4. View certificates and install modules
SEEBEYOND
Download Modules
Stop
< Back
Next >
Finish
Cancel
Help

The Update Center is now downloading modules and/or checking digital signatures.

eVision [1/11]

Downloaded 2,100 of 34,656 KBytes.

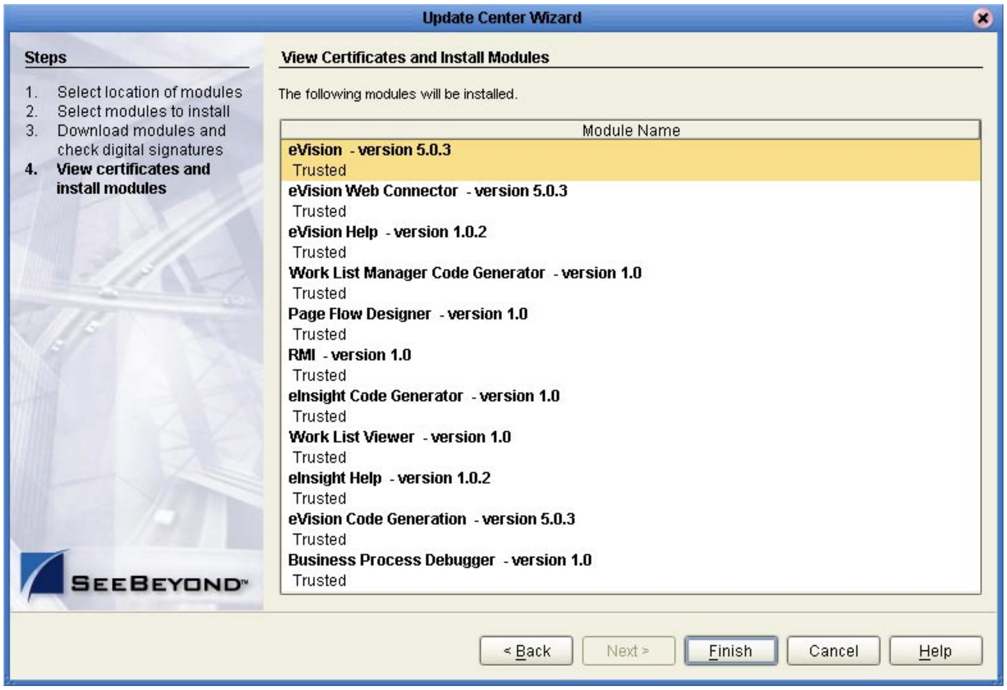
Download In Progress ...

When the download process is complete, the message **Done** appears below the progress bars.

7 Click **Next**.

8 The Update Center Wizard displays the list of certificates and installed modules. See Figure 10.

Figure 10 Update Center Wizard: View Certificates and Install Modules


Update Center Wizard
Steps
1. Select location of modules
2. Select modules to install
3. Download modules and check digital signatures
4. View certificates and install modules
SEEBEYOND
< Back
Next >
Finish
Cancel
Help

View Certificates and Install Modules

The following modules will be installed.

Module Name
eVision - version 5.0.3 Trusted
eVision Web Connector - version 5.0.3 Trusted
eVision Help - version 1.0.2 Trusted
Work List Manager Code Generator - version 1.0 Trusted
Page Flow Designer - version 1.0 Trusted
RMI - version 1.0 Trusted
eInsight Code Generator - version 1.0 Trusted
Work List Viewer - version 1.0 Trusted
eInsight Help - version 1.0.2 Trusted
eVision Code Generation - version 5.0.3 Trusted
Business Process Debugger - version 1.0 Trusted

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

10 Click **OK**.

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

Chapter 4

eVision Studio Components

This chapter focuses on eVision Studio's Page Layout Designer. Each pre-built component that you can include in a Page Layout is described. This chapter also describes how to create a Page Link.

This Chapter Includes:

4.1 Design Tools

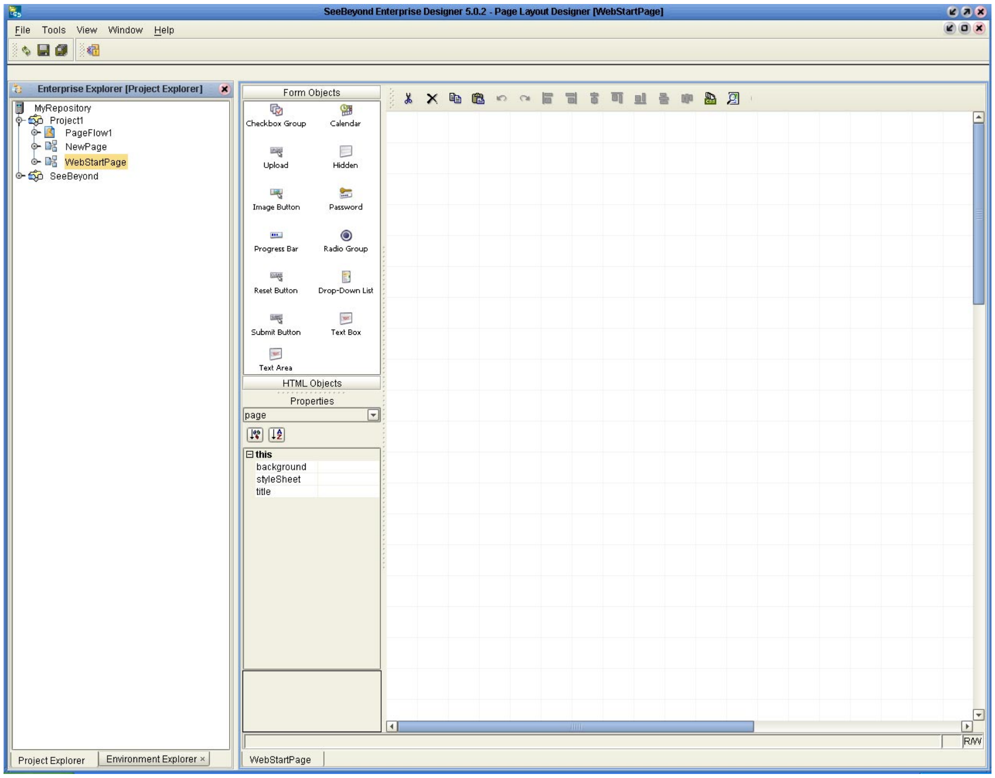
Through the use of familiar drag-and-drop techniques and text-based property sheets, eVision Studio allows you to interactively add graphics, text, and programmatic content to the Web pages in your application.

You drag pre-built components from a convenient component palette and position them on the design canvas. The component's property sheet is opened automatically. The property sheet allows you to specify attributes, such as the component name and the user-facing text to be displayed on a Web page.

As you create your Web pages, you can preview the design in your browser at any time.

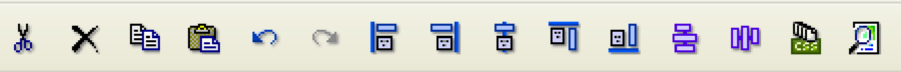
Figure 11 shows an example of the Page Layout Designer with the **Form Objects** palette active.

Figure 11 Page Layout Designer Canvas at Startup



The Page Layout Designer toolbar allows you to manipulate the objects that you place on the canvas. The toolbar is shown in Figure 12.

Figure 12 Page Layout Designer Toolbar



Each tool is described in Table 2.

Table 2 Page Layout Designer Tools

Tool	Name	Purpose
✂	Cut Component	Cuts a component from the canvas and places it on the clipboard.
✕	Delete	Deletes a component permanently.
⎘	Copy Component	Copies a component from the canvas and places it on the clipboard.
📋	Paste Component	Pastes a component from the clipboard onto the canvas.
↶	Undo	Rolls back your most recent input or interaction, and then continues backward up to the last time the layout was saved.
↷	Redo	Reverses the most recent rollback, and continues forward up to your last input or interaction.
	Left Align	Aligns two or more selected components with the left-most component in the selected group.
	Right Align	Aligns two or more selected components with the right-most component in the selected group.
	Center Align	Aligns two or more selected components with the center of the canvas (the center of the Web page).
	Top Align	Aligns two or more selected components with the top-most component in the selected group.
	Bottom Align	Aligns two or more selected components with the bottom component in the selected group.

Tool	Name	Purpose
	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.
CSS	Edit CSS	Launches the **Choose CSS to Edit** dialog box, which allows you to select an imported style sheet to edit with the **eVision Style Editor** tool. For more information, see **"Customizing a Style Sheet" on page 49**.
	Preview	Previews the Web page layout in your browser.

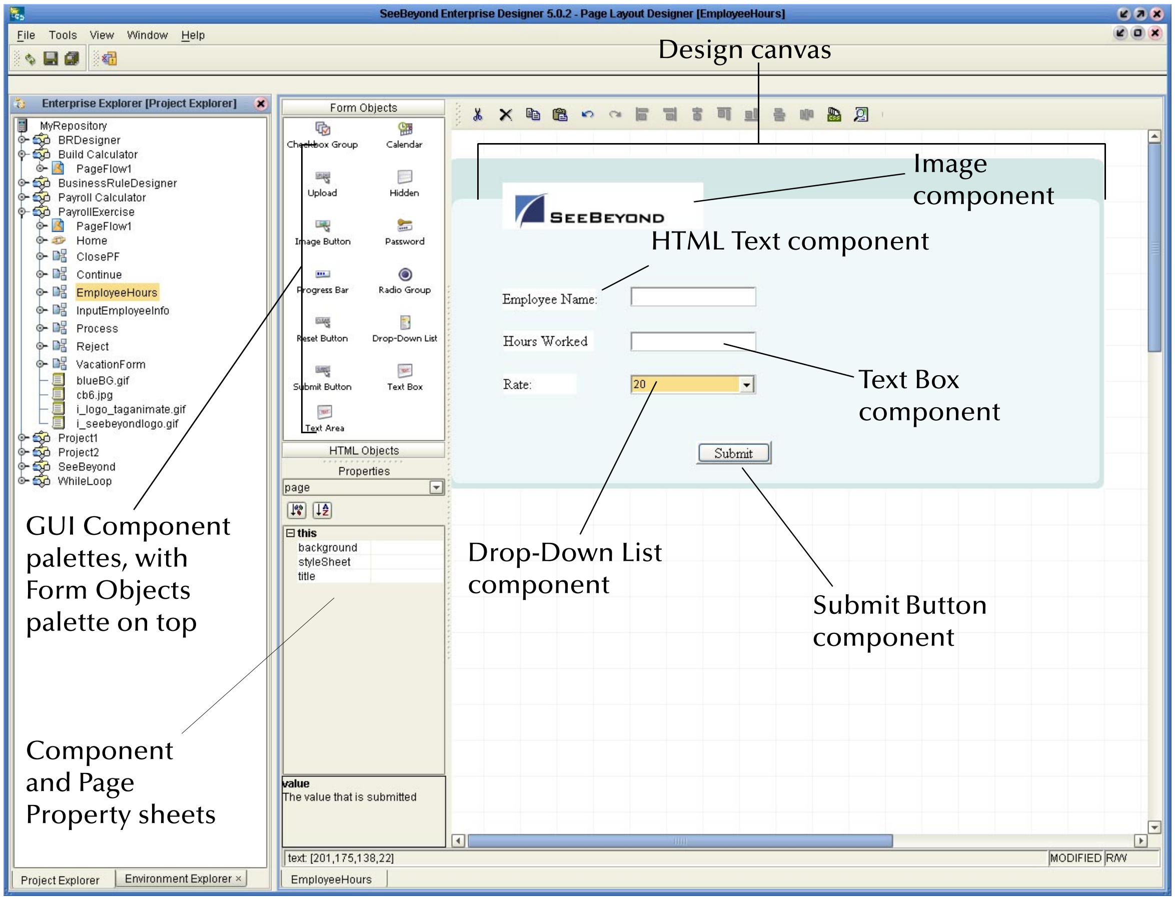
4.2 Graphical User Interface Components

Graphical User Interface (GUI) components gather input data from the user at run-time, and add functions and interactivity to Web pages. To add a GUI component to the canvas, you select a component from the **HTML Objects** or **Form Objects** palette, drag it onto the canvas, and release the mouse button. The Page Layout Designer provides familiar Web-centric design elements, such as check boxes, text boxes, drop-down lists, graphics containers, horizontal lines, and submit buttons.

GUI components are pre-built combinations of Java classes and JSP code that represent Web interface elements. When you place a GUI component on the canvas, the component's property sheet is automatically activated. In the property sheet, you add the labeling, functionality, and appearance attributes that you want the component to have when it is presented to the user in a browser.

In the Page Layout Designer, the upper left pane contains the GUI component palettes. See Figure 13.

Figure 13 Page Layout Designer GUI Components

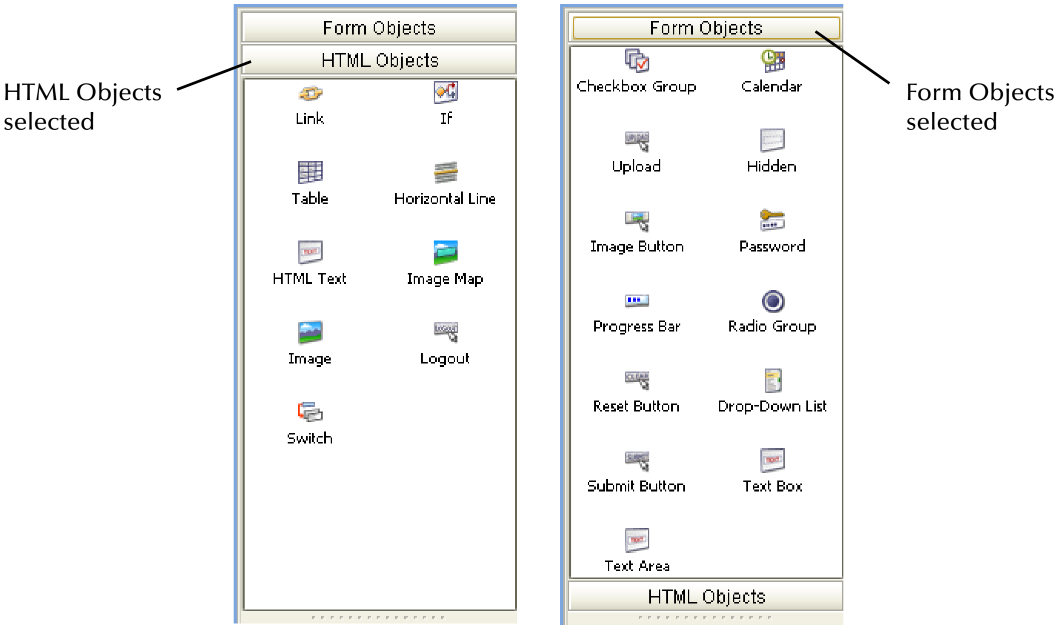


4.2.1 GUI Component Palettes

eVision Studio provides two palettes that allow you to drag and drop GUI components onto the canvas to quickly create a Web page layout from eVision's collection of pre-built objects. When you place a GUI component on the canvas, you then follow up by customizing its functional and visual presentation properties in the **Properties** window.

To access the HTML Objects palette, select the **HTML Objects** title bar. To access the Form Objects palette, select the **Form Objects** title bar. See Figure 14.

Figure 14 HTML and Form Objects Palettes



HTML Objects

The **HTML Objects** palette allows you to drag and drop HTML-based GUI components onto the Page Layout Designer canvas. HTML objects are represented graphically on a Web page and may or may not have programmatic attributes. HTML objects are described in Table 3.

Table 3 HTML Objects

Component	Name	Purpose
	Link	Creates a link to another location, another page, or an external Web site. Page links are created using the **Page Link Wizard**. For more information, see **"Linking Pages" on page 56**.
If	If	Specifies a response if a condition is true or false.
	Table	Creates a table of rows and columns. Table cells may contain any object from the HTML or Forms Objects palettes. A table can contain both *static* and *dynamic* rows. For more information about dynamic tables, see **"Dynamic Tables" on page 36**. Once you create a table, you can modify the number of rows and columns in one of two ways: by changing the **rows** and **cols** properties in the table Properties sheet, or by right-clicking the table and using the **Grow** and **Shrink** menu items.

Component	Name	Purpose
☰	Horizontal Line	Places a horizontal line on the canvas. Use the Horizontal Line object to create visual separations in your layouts. 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 will change 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. To move the line, click the line's mid-point (avoid the end points), and drag it to a new location.
HTML	HTML Text	Creates a field on the canvas to hold HTML (static) text, or can act as a placeholder for dynamic text. You use this component for labels and general information on a page.
🖼	Image Map	Allows 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. ***Note:*** *Images must reside in your Project in the Repository before you can access them.* For more information, see **"Image Maps" on page 39**.
🖼	Image	Holds a static graphic image. When you drag the icon onto the canvas, you are prompted to select the image. ***Note:*** *Images must reside in your Project in the Repository before you can access them.* For more information, see **"Importing Images" on page 44**.
Logout	Logout	Allows the user to exit the eVision Web application. Clicking this button returns the user to the beginning of the Page Flow. For more information, see **Logout Buttons** on page 41.

Component	Name	Purpose
	Switch	Allows you to add conditional logic to a Web page. Multiple switches may be specified to conditionally control the display of a single Web page. eVision passes the desired display conditions to the Web page at run-time.

Form Objects

Form Objects are pre-built combinations of Java classes and JSP code representing Web interface entities. These objects allow users to communicate with Page Flows in the run-time environment. Form objects are described in Table 4.

Table 4 Form Objects

Component	Name	Purpose
	Checkbox Group	Creates a group of check boxes. Indicates inclusive user selection at run-time. You can create multiple check boxes within a group. For more information, see **Check Box Groups** on page 41.
	Calendar	Adds a calendar under a drop-down arrow. On a Web page, selecting a date on the calendar object populates an attached text box with the selected date.
UPLOAD	Upload	Allows the user to upload files and data to an application.
	Hidden	Creates a hidden text field (a text field that is hidden from users at run-time), which can be used to pass session information to another page without being visible to users.
	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.
	Password	Creates a password input box with asterisks that mask the password.
	Progress Bar	Creates a progress bar that allows the user to monitor the progress of a particular operation on a Web page.

Component	Name	Purpose
◉	Radio Group	Switches an attribute or condition on or off. You can create multiple buttons in a group. ***Note:*** *The Value property cannot be empty. This object requires that a value be assigned in order to pass the true/false condition.* For more information, see **Radio Groups** on page 42.
CLEAR	Reset Button	Allows the user to cancel an operation or reset values to a default condition.
▤	Drop-Down List	Allows the user to select an entry from a drop-down list. You can create multiple entries on a list. For more information, see **Drop-Down Lists** on page 43.
SUBMIT	Submit Button	Allows the user to launch an operation or submit text to the application. Passes accumulated values to the ICAN system.
TEXT	Text Box	Allows the user to type text that will be displayed on the Web page surrounded by a bounding box.
TEXT	Text Area	Allows the user to type an extensive body of text and place it anywhere on the Web page without disturbing formatting.

4.2.2 Dynamic Tables

Dynamic tables are created like static tables but with one or more rows designated as “dynamic.” In a dynamic table, the total number of rows, and the row content are undefined. At run-time, table rows and content are dynamically generated by a repeating element in a Page Flow.

To Define a Dynamic Row in a Table

1. Select a cell in the table. The cell will be highlighted in blue.
2. Right-click the cell.
3. On the context menu, select **Set As Dynamic Row**.

   The row will be marked as being dynamic.

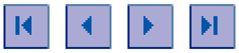
Adding Pagination to a Dynamic Table

You can add a “page-forward, page-back”, “page-first, page-last” function to a row or rows in a dynamic table. The Pagination feature allows table cell data to be incrementally displayed. 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 allow 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 provides the following default image files for the pagination buttons: **firstImg.gif**, **prevImg.gif**, **nextImg.gif**, and **lastImg.gif**. See Figure 15.

Figure 15 Default Pagination Buttons

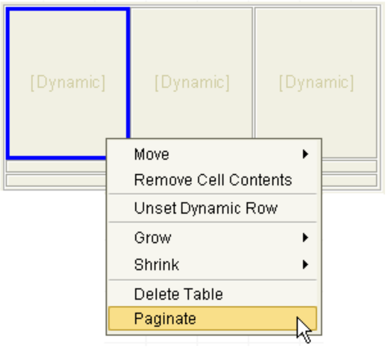


If you want to use your own image files, you must first import them into the Project. See **"Importing Images" on page 44**. You then specify the replacement image files during the following procedure.

To Add Pagination to a Row in a Dynamic Table

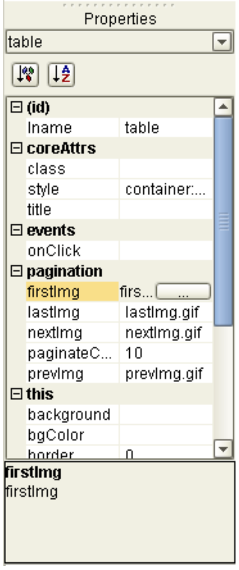
1. Right-click the dynamic row.
2. On the context menu, click **Paginate**. See Figure 16.

Figure 16 Paginate Menu Option



3. Click and drag a bounding box around the table.
4. In the table **Properties** sheet, the **paginateCount** property specifies how many rows the table will accept at a time. The default value is 10. If desired, increase or decrease the value.
5. In the table **Properties** sheet, the **firstImg**, **lastImg**, **nextImg**, and **prevImg** properties are set to the default image files for the pagination buttons. See Figure 17.

Figure 17 Pagination Properties



⊟ **(id)**	
lname	table
⊟ **coreAttrs**	
class	
style	container:...
title	
⊟ **events**	
onClick	
⊟ **pagination**	
firstImg	firs... ...
lastImg	lastImg.gif
nextImg	nextImg.gif
paginateC...	10
prevImg	prevImg.gif
⊟ **this**	
background	
bgColor	
border	0

firstImg
firstImg

If you want to use your own image files, then click a property value, navigate to the image file, and click **Open**.

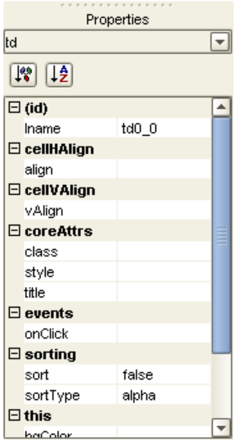
Adding Sorting to a Dynamic Table

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. Click a column cell.
2. In the **Properties** sheet, make sure that the **td** properties are displayed. See Figure 18.

Figure 18 td Properties



⊟ **(id)**	
lname	td0_0
⊟ **cellHAlign**	
align	
⊟ **cellVAlign**	
vAlign	
⊟ **coreAttrs**	
class	
style	
title	
⊟ **events**	
onClick	
⊟ **sorting**	
sort	false
sortType	alpha
⊟ **this**	
bgColor	

3. Set the value of the **sort** property to **true**.

4. Specify how the column should be sorted by selecting a value for the **sortType** property. If you want to consider case when sorting alphabetic values, select the **alpha** value; otherwise, select the **alphaIgnoreCase** value.
5. Save the Page.

4.2.3 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.

Important: *Before you begin, you must import the image into the Project. For more information, see* **"Importing Images" on page 44**.

When you work with image maps, the following tools are added to the Page Layout Designer toolbar:

- The **Select Link Area** tool highlights the link area.
- The **Select Image Map** tool highlights the image.

Only one of these tools is enabled at a time.

To Create an Image Map

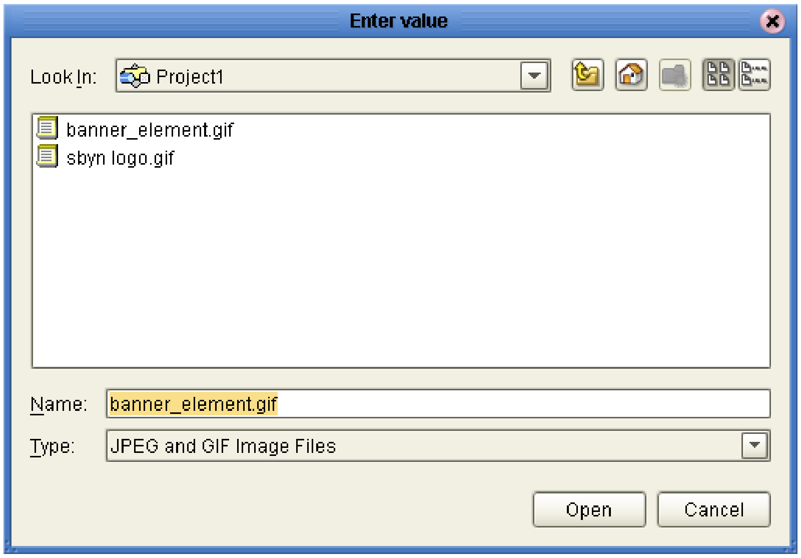
1. From the **HTML Objects** palette, drag the **Image Map** icon onto the canvas.

   When the icon transfers onto the canvas, it changes into a dashed-line rectangle.

2. Drop the rectangle onto the canvas.

   The **Enter value** dialog box appears. See Figure 19.

Figure 19 Selecting the Image for an Image Map



Look In: Project1

Name: banner_element.gif

Type: JPEG and GIF Image Files

3 Select the image for the image map.

4 Click **Open**.

The image appears on the canvas. The gold outline represents the outside boundaries.

5 In the **Properties** sheet, define the value for **lname** (the component's logical name) by doing the following:

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

B In the right column, delete the default value and enter a descriptive name for the component (for example, **logoImageMap**).

lname is the internal, logical name for the component, which identifies the component in the Page Flow.

6 If you want to resize the image container, click and drag a highlighted boundary element. The shape of the bidirectional arrow cursor indicates which way the container will be resized.

If you want to move the image container, click the container in the center (away from the edges) and drag it to the new location.

7 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 rectangle appears. This is the *link area*.

8 In the **Properties** sheet, 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**).

9 If you want to resize the link area, click and *slowly* drag a highlighted boundary element.

If you want to move the link area, click the link area in the center (away from the edges) and drag it to the new location.

You can also resize and move the link area by manually entering the coordinates in the **Properties** sheet. In the left column, click the **coords** property. In the right column, change the value.

If the link area that you are trying to size to is too small, right-click the image and choose **Set Link Area to Image Size**.

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

To Test an Image Map

1 On the Page Layout Designer toolbar, click the **Preview** icon.

In the preview image, when you pass the pointer over the mapped area, the cursor changes to the “hand” icon.

2 Click the defined image area to execute the link code.

4.2.4 Logout Buttons

The **Logout** button allows the user to exit the eVision Web application. Clicking this button returns the user to the beginning of the Page Flow. You can place a **Logout** button on any page.

To Create a Logout Button

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

The **Logout** button appears. The gold outline represents the outside boundaries of the button.

2 In the **Properties** sheet, change the default value of the **lname** property (the component’s logical name).

3 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 Images** on page 44.

B In the **Properties** sheet, 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**.

4.2.5 Check Box Groups

The **Checkbox Group** component allows you to create a group of check boxes. You specify the label and value for each check box.

To Create a Checkbox Group

1 Drag the **Checkbox Group** component onto the canvas.

2 Right-click the component.

3 On the context menu, select **Edit Options**.

The **Edit Options** dialog box appears.

4 Select a **Label** field and type a label.

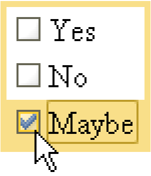
The label will be displayed next to the check box at run-time.

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

The value will be submitted when the user checks the box at run-time.

6 To add additional check boxes to the group, click **Add** and repeat the previous steps. See Figure 20.

Figure 20 Editing Checkbox Group Options



□ Yes
□ No
☑ Maybe

7 Click **OK**.

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

4.2.6 Radio Groups

The **Radio Group** component allows you to create a group of radio buttons. You specify the label and value for each radio button.

To Create a Radio Group

1 Drag the **Radio Group** component onto the canvas.

2 Right-click the component.

3 On the context menu, select **Edit Options**.

The **Edit Options** dialog box appears.

4 Select a **Label** field and type a label.

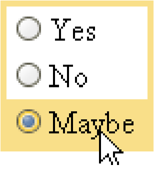
The label will be displayed next to the button at run-time.

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

This value will be submitted when the user clicks the button at run-time.

6 To create additional buttons in the group, click **Add** and repeat the previous steps. See Figure 21.

Figure 21 Editing Radio Group Options



○ Yes
○ No
● Maybe

7 Click **OK**.

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

4.2.7 Drop-Down Lists

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

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

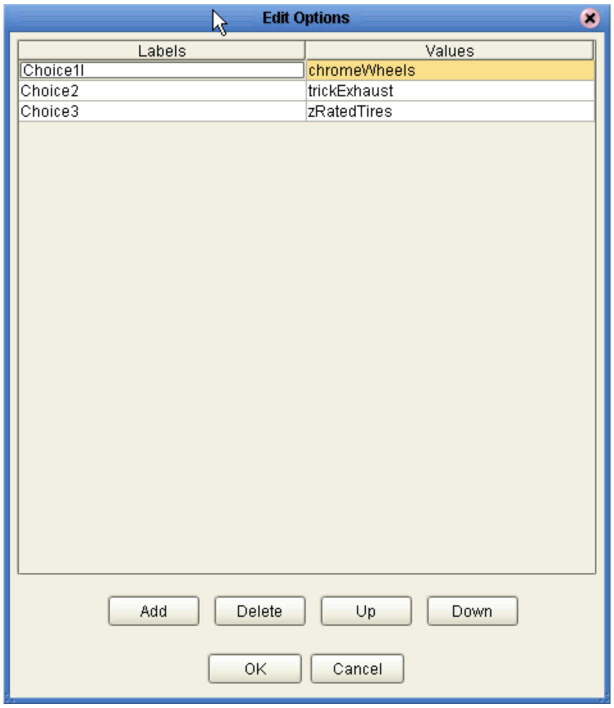
1. Drag the **Drop-Down List** component onto the canvas.
2. Right-click the component.
3. On the context menu, select **Edit Options**.

   The **Edit Options** dialog box appears.
4. Select a **Label** field and type a label.

   The label will be displayed as an option to users at run-time.
5. In the **Value** field, type a value.

   The value will be submitted when the user clicks the label at run-time.
6. To create additional **Drop-Down List** entries, click **Add** and repeat the previous steps. See Figure 22.

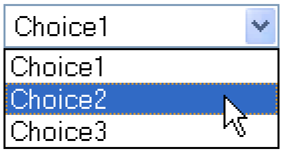
Figure 22 Edit Options Dialog Box



Labels	Values
Choice1	chromeWheels
Choice2	trickExhaust
Choice3	zRatedTires

7. Click **OK**.
8. To see how the list will be presented to users at run-time, click the **Preview** icon. See Figure 23.

Figure 23 Drop-Down List Preview



9 To make the **Drop-Down List** component dynamic, do the following:

A Right-click the component.

B On the context menu, select **Make Dynamic**.

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

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

4.2.8 Importing Images

To make graphic images accessible to your Web page, you must first import them into your Project from their location on disk.

To Import an Image

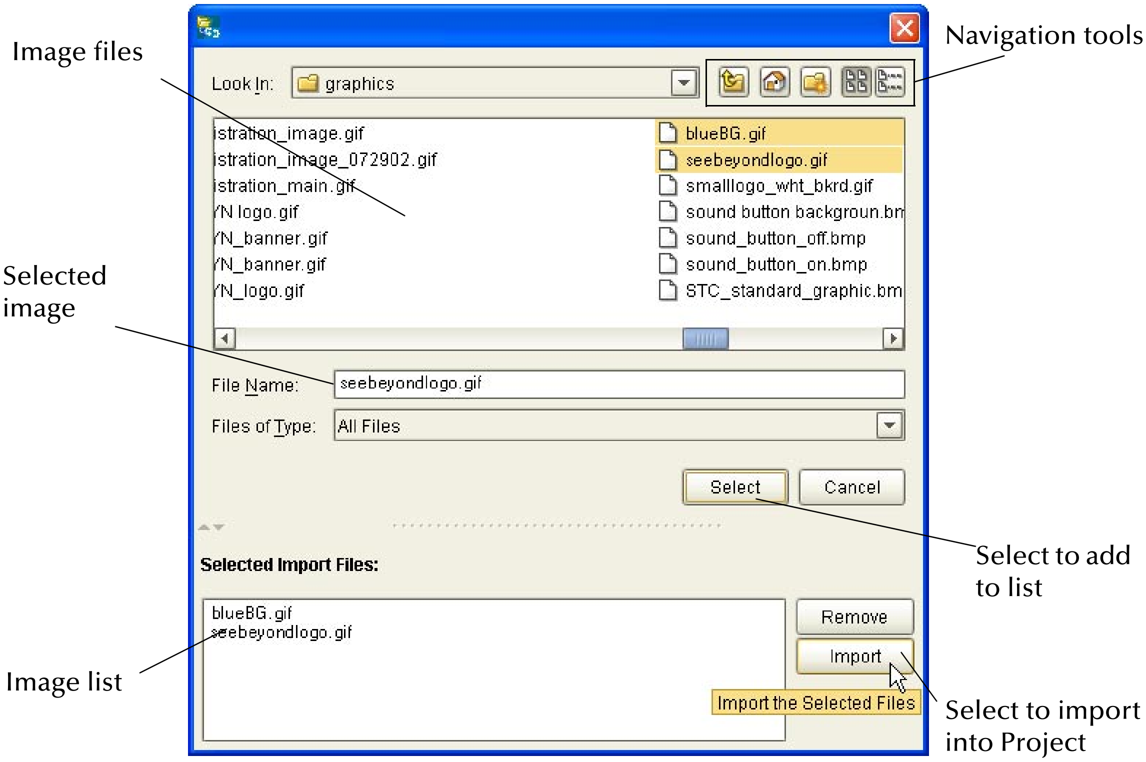
1 In the Project Explorer, right-click the **Project** icon.

2 On the context menu, select **New**, and then select **File**.

3 In the dialog box that appears, navigate to the directory that contains the file or files that you want to import. See Figure 24.

The image files can reside anywhere on your machine or a network.

Figure 24 Importing Images into a Project



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

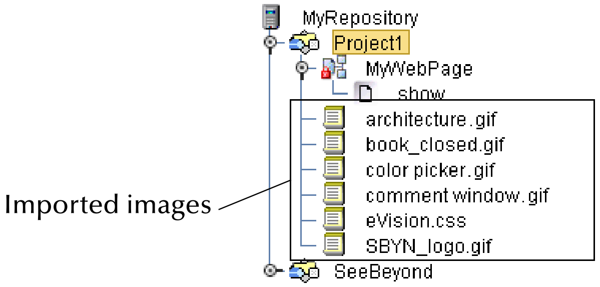
The file names appear in the **Selected Import Files** box.

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 images are displayed under your Project in the Project Explorer tree. See Figure 25.

Figure 25 Imported Images in Project Explorer Tree

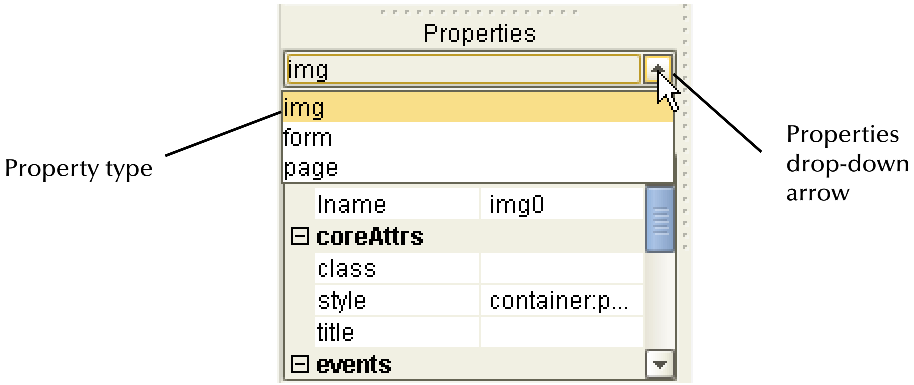


4.3 Defining GUI Component Properties

When you drop a GUI component onto the canvas, the component's property sheet is automatically displayed. Component Properties sheets allow you to customize the HTML (static) and FORM (dynamic) components.

When you click an HTML or Form Object, its properties are automatically displayed in the **Properties** window. If you want to switch from one property type to another without clicking an object on the canvas, you can click the **Properties** drop-down arrow and click the property type that you want to access from the drop-down menu. A property type is typically Page properties, image properties (img), or HTML/Form Object properties. See Figure 26.

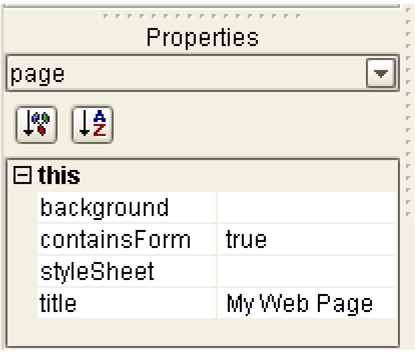
Figure 26 Properties Drop-Down Menu



lname	img0
⊟ coreAttrs	
class	
style	container:p...
title	
⊟ events	

For example, when you click **page** from the **Properties** drop-down menu, the **page** properties sheet pops to the front. See Figure 27.

Figure 27 New Page Properties



⊟ this	
background	
containsForm	true
styleSheet	
title	My Web Page

Each component has its own, unique set of user-definable properties. Some properties are required; for example, you must typically provide a label (a logical name) for a component, and at least one other property, such as a text string to be presented to Web application users at run-time.

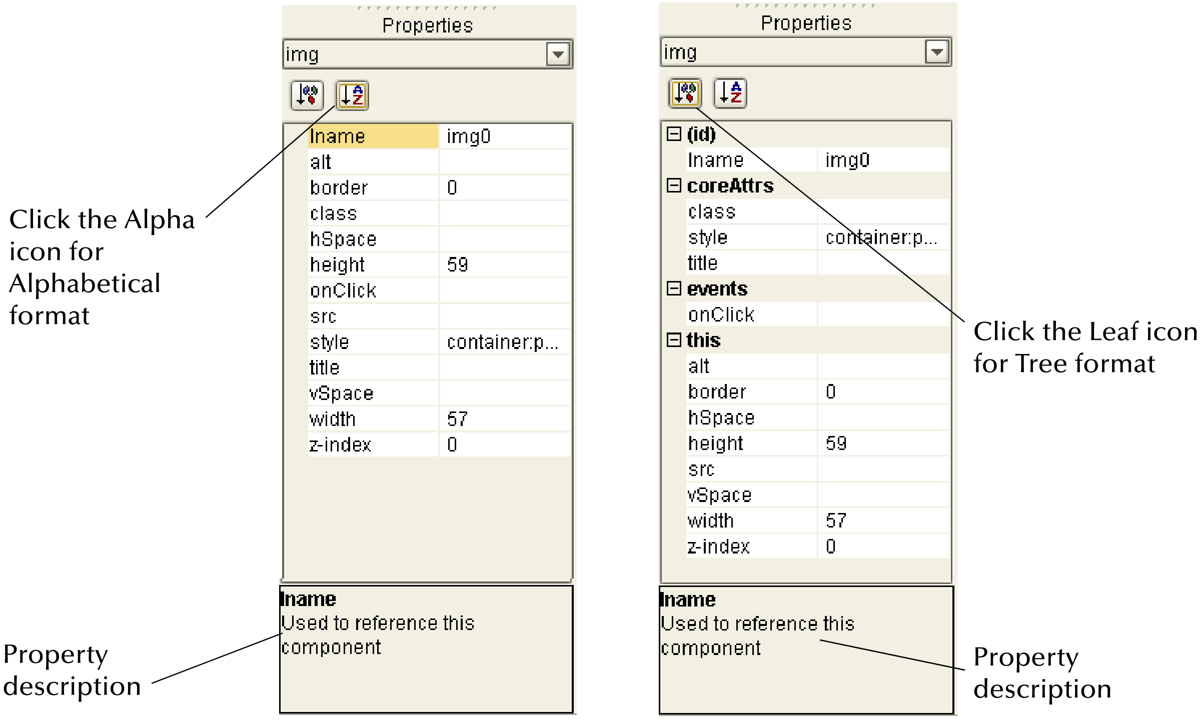
Important: *Be sure to change the default value of the* ***lname*** *property to a more descriptive value. The Page Flow Designer uses the* ***lname*** *property as the identifier for components. Using a descriptive value will make it easier for you to identify the component in the Page Flow Designer.*

A description of each property is provided in two ways:

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

You can specify whether to display the property list in alphabetical or Explorer (tree) formats. For alphabetical format, click the **Alpha** icon. For tree format, click the **Leaf** icon. See Figure 28.

Figure 28 Swapping Component Properties Formats


Properties
img
Click the Alpha icon for Alphabetical format
Property description
Properties
img
Click the Leaf icon for Tree format
Property description

lname	img0
alt	
border	0
class	
hSpace	
height	59
onClick	
src	
style	container:p...
title	
vSpace	
width	57
z-index	0

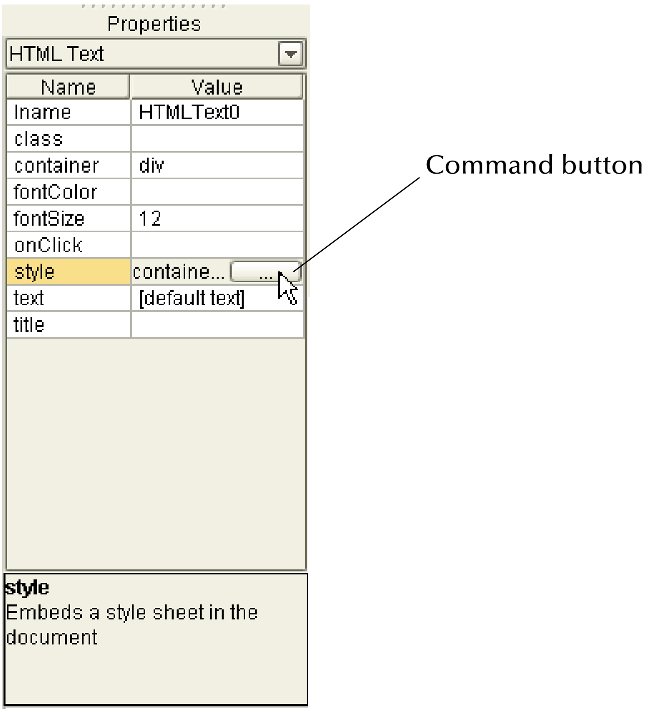
lname
Used to reference this component

⊟ **(id)**	
lname	img0
⊟ **coreAttrs**	
class	
style	container:p...
title	
⊟ **events**	
onClick	
⊟ **this**	
alt	
border	0
hSpace	
height	59
src	
vSpace	
width	57
z-index	0

lname
Used to reference this component

To define a property, select any field and type directly into the highlighted field, or click the **Command** button (**...**) to display the **Enter value for (“text/src/style”)** dialog box and type the desired value into the text input area in the dialog box text field.

Figure 29 Command Button

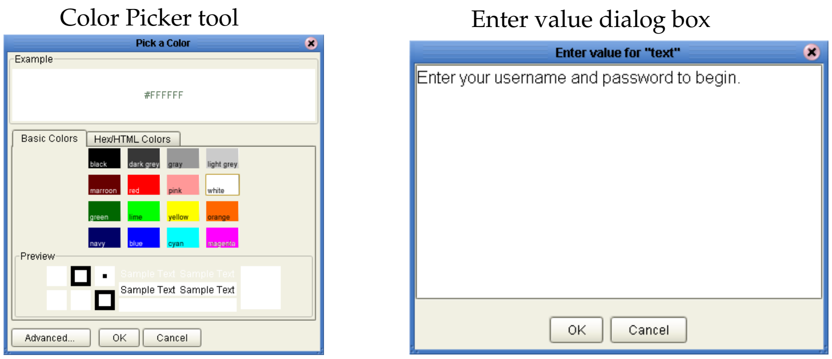


Name	Value
lname	HTMLText0
class	
container	div
fontColor	
fontSize	12
onClick	
style	containe...
text	[default text]
title	

style
Embeds a style sheet in the document

The **Command** button indicates that another window, tool, or dialog box will be triggered when the button is clicked. See Figure 30.

Figure 30 Interface Elements Displayed With the Command Button


Color Picker tool

Enter value dialog box

Enter your username and password to begin.

4.4 Moving Overlapping Components in the Z-Direction

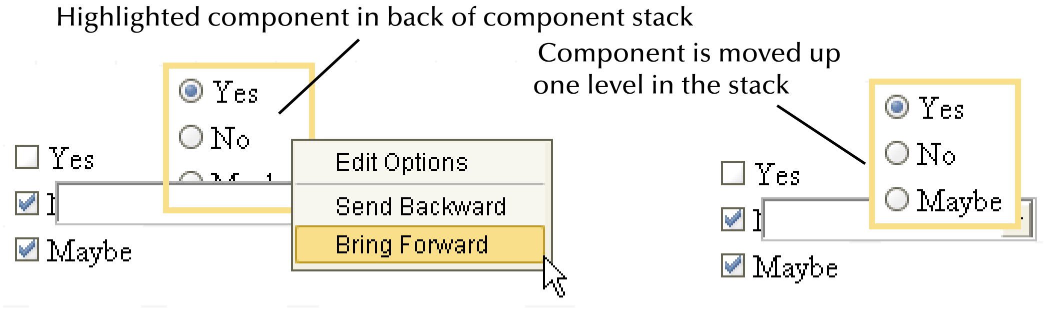
The standard 2D graphical orientation consists of two values: **x** (horizontal) and **y** (vertical). In 2D, a third value (**z**) implies another planar orientation, which is toward or away from the viewer (backward/forward). The z-value allows you to place objects under or on top of each other using an integer to specify the “plane.”

Components can often overlap. Sometimes this is desirable; other times, it is not. You can move components in the z-direction one level at a time, forward or back, using the context menu.

To move a component in the z-direction, one level at a time, do the following:

1. Highlight the component you want to move.
2. Place the cursor inside the component's highlight boundary and right-click the mouse.
3. On the context menu, select **Bring Forward** or **Send Backward** as appropriate. See Figure 31.

Figure 31 Moving Components in the Z-direction



Highlighted component in back of component stack

Component is moved up one level in the stack

To modify the z-value of a component, right-click the component, and in the context menu, do the following:

- To move the component back (down) one level, click **Send Backward**.
- To move the component forward (up) one level, click **Bring Forward**.

You can also move an object forward or backward in the stack order by entering a negative or positive integer in the **z-index** property on the property sheet. Initially, all objects are dropped onto the same layer, where the default plane is "**0**."

- To move an image down (back) one or more levels, click the **z-index** property on the property sheet and type a negative integer; for example: **-9**.
- To move an image up (forward) one or more levels, click the **z-index** property on the property sheet and type a positive integer; for example **7**. Do not include a plus sign.

4.5 Customizing a Style Sheet

Style sheets control the fonts and formatting of a Web page. For example, in a style sheet, a font and its characteristics such as color and size are assigned to an HTML tag (for example, a paragraph or heading). When the tag is used, the font definition is employed and the font is displayed on the Web page according to the style elements assigned to it.

eVision Studio provides two style sheets for immediate use: **eVision-default.css** and **eVision.css**. To apply a style sheet, see **"To Apply a Style Sheet to Your Web Application" on page 53**.

Before you start designing a page, you may want to import and apply your own customized style sheet. To import a style sheet, see **"Importing a Style Sheet Into Your Project" on page 53**.

If you do not have a preferred style sheet that you want to apply, you may want to create an updated style sheet from the existing source files to manage the page format. You can create a custom style sheet done in two ways:

- Edit one of the default style sheets.
- Import a style sheet from an external source and modify it.

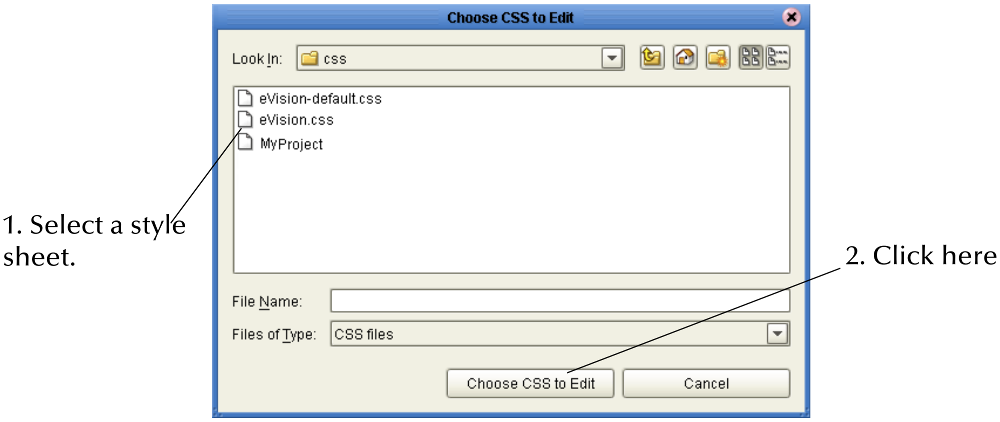
4.5.1 Creating a New Style Sheet from eVision.css

To create a custom style sheet, you can start by modifying one of the default style sheets and use it as a starter template.

To create a new style sheet from the **eVision.css** style sheet, do the following (recommended):

1. Click the **Edit CSS** icon on the Page Layout Designer toolbar. The **Choose CSS to Edit** dialog box appears.
2. Click **eVision.css**.
3. Click **Choose CSS to Edit**. See Figure 32.

Figure 32 Choosing a Style Sheet to Edit



Look In: css

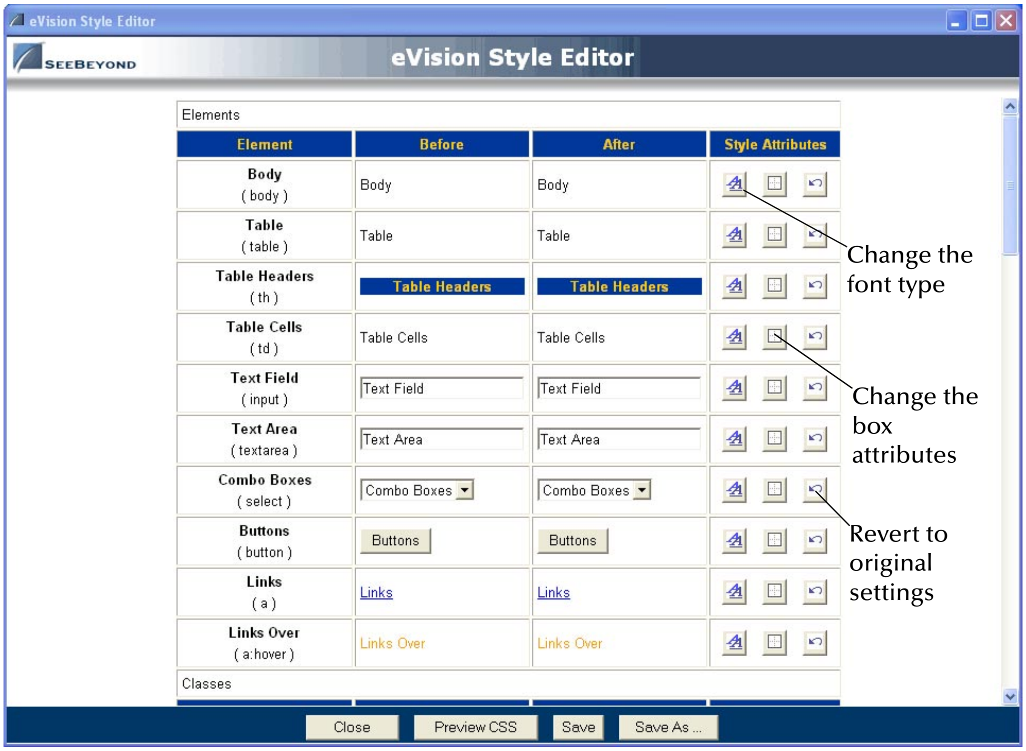
Files of Type: CSS files

1. Select a style sheet.

2. Click here

The **eVision Style Editor** appears. This editor allows you to modify a host of properties that will be used to manage the format of a Web page. See Figure 33.

Figure 33 eVision Style Editor



Elements

Element	Before	After	Style Attributes
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	
Links Over (a:hover)	Links Over	Links Over	

Classes

Change the font type

Change the box attributes

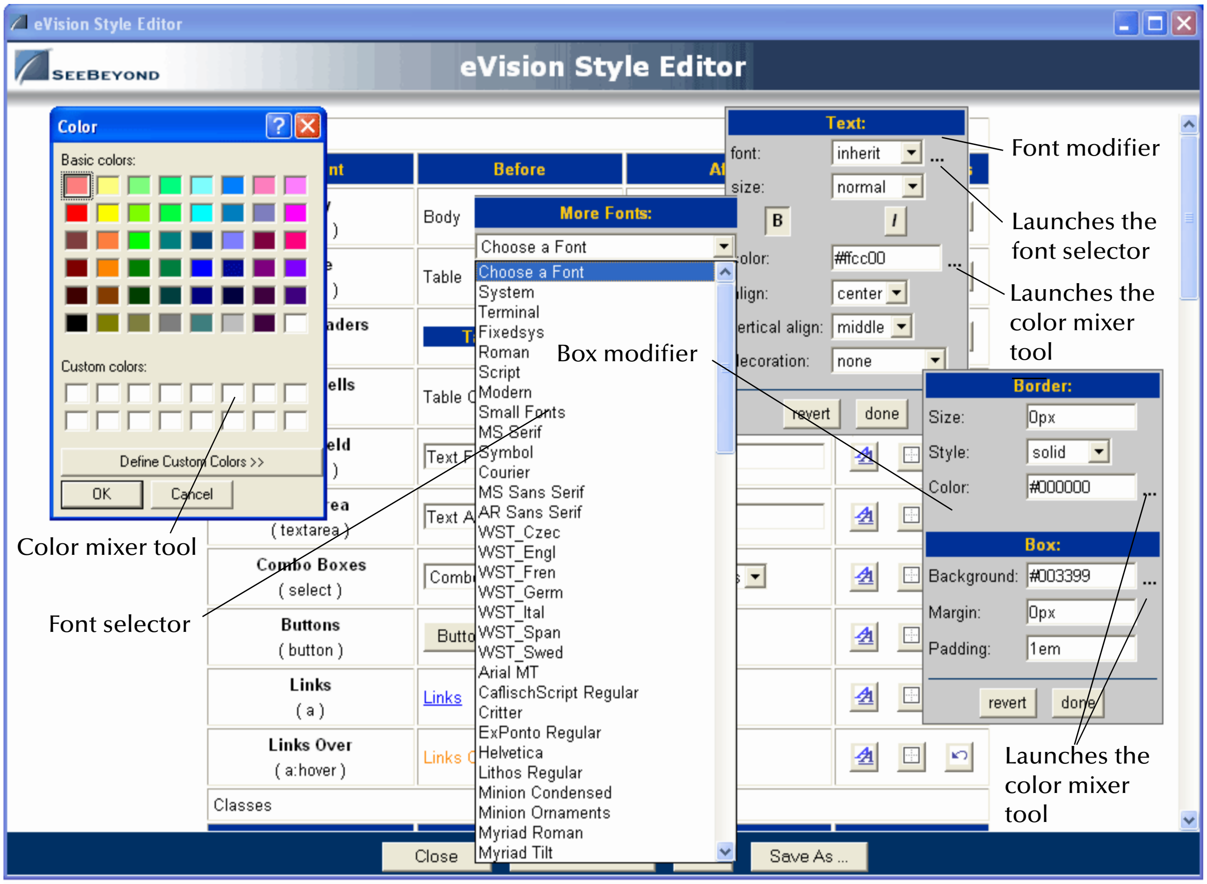
Revert to original settings

The **eVision Style Editor** contains a list of style sheet elements that you can modify to control the format of your Web page. Modifications that you can make to style sheet elements include:

- **Font**—type, size (percent), and color; bold and/or italic
- **Font decoration**—underscore, overscore, line-through
- **Horizontal element alignment**— left, right, center, justify
- **Vertical element alignment**—top, bottom, or center
- **Border attributes**—size, color, and line style including groove, ridge, inset, or outset
- **Box attributes**—background color, margin and padding in pixels

For a composite view of the available **eVision Style Editor** tools, see Figure 34.

Figure 34 eVision Style Editor Tools



Text:
- font: inherit
- size: normal
- color: #ffcc00
- align: center
- vertical align: middle
- decoration: none

Border:
- Size: 0px
- Style: solid
- Color: #000000

Box:
- Background: #003399
- Margin: 0px
- Padding: 1em

The **eVision Style Editor** retains the original specifications for the elements you have changed. The original element configuration is shown under the “**Before**” heading. A real-time rendering of the elements you have modified is displayed under the “**After**” heading.

- To change back to the original configuration (reverse your changes), click **Revert**.
- To view the changes you have made, click **Preview CSS**, and then click **Back to Editor**.
- To save your changes and overwrite the template, click **Save CSS**.
- To save the style sheet under a new name (recommended), click **Save As** and type a unique name in the dialog box that appears.

4.6 Importing a Style Sheet Into Your Project

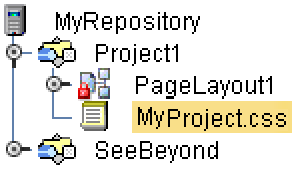
When you are satisfied with the element configuration in your style sheet, you must import it into your Project before it can be applied.

To Import a Custom Style Sheet

1. In the Project Explorer, right-click the Project.
2. On the context menu, click **New**, and then select **File**.
3. Navigate to the <*ICAN-root*>/**edesigner/usrdir/modules/ext/stc/evision_core/tigris/css** directory.
4. Select one or more style sheets and click **Select**.
5. Click **Import**.

   The style sheet(s) will appear in the Project Explorer. See Figure 35.

Figure 35 Style Sheet in Project Explorer

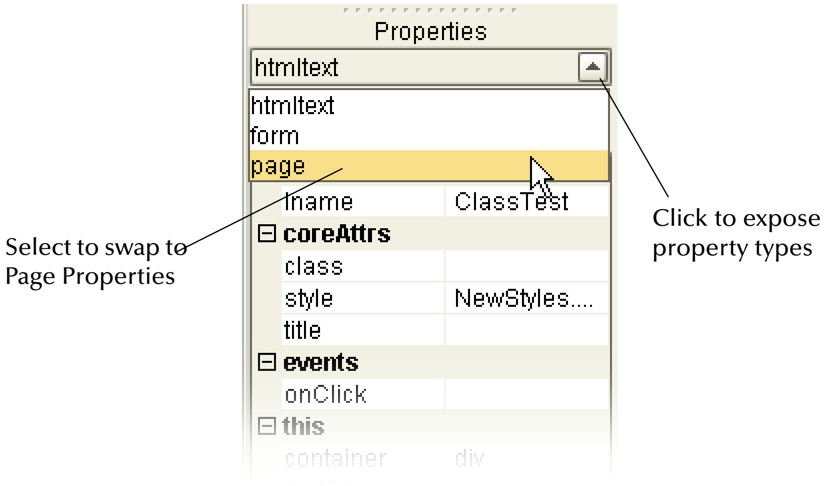


The selected style sheets are now a part of your Project and can be applied your application's Web pages.

To Apply a Style Sheet to Your Web Application

1. In the **Properties** window, click the drop-down arrow.
2. On the drop-down list, click **page**. See Figure 36.

Figure 36 Accessing the Page Properties



name ClassTest
class
style NewStyles....
title
onClick
container div

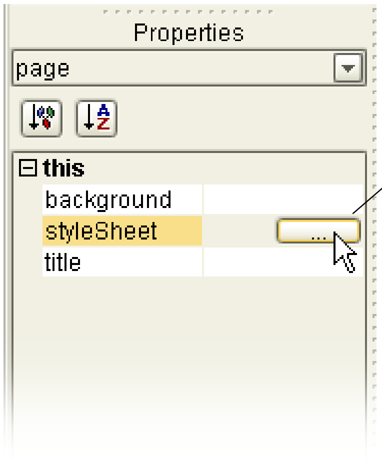
Select to swap to Page Properties

Click to expose property types

3 In the left column, select the **styleSheet** property.

4 In the right column, click the **Command** button (**...**). See Figure 37.

Figure 37 styleSheet Property

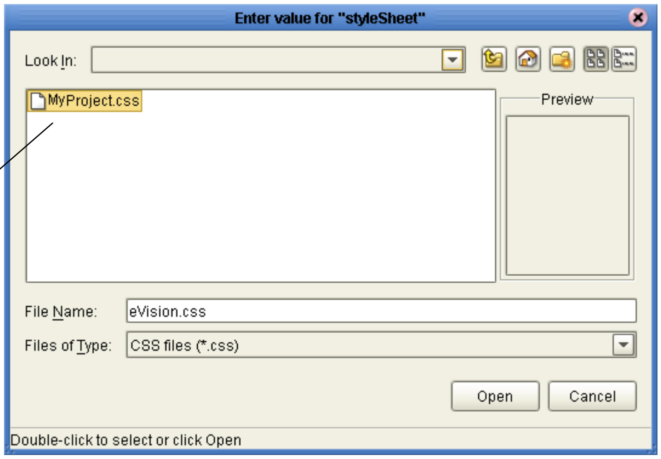


Click to navigate to the style sheet to apply to the page

5 In the **Enter value for "styleSheet"** dialog box, navigate to the Project folder.

6 Select the style sheet that you want to apply, and then click **Open**. See Figure 38.

Figure 38 Selecting a Style Sheet to Apply



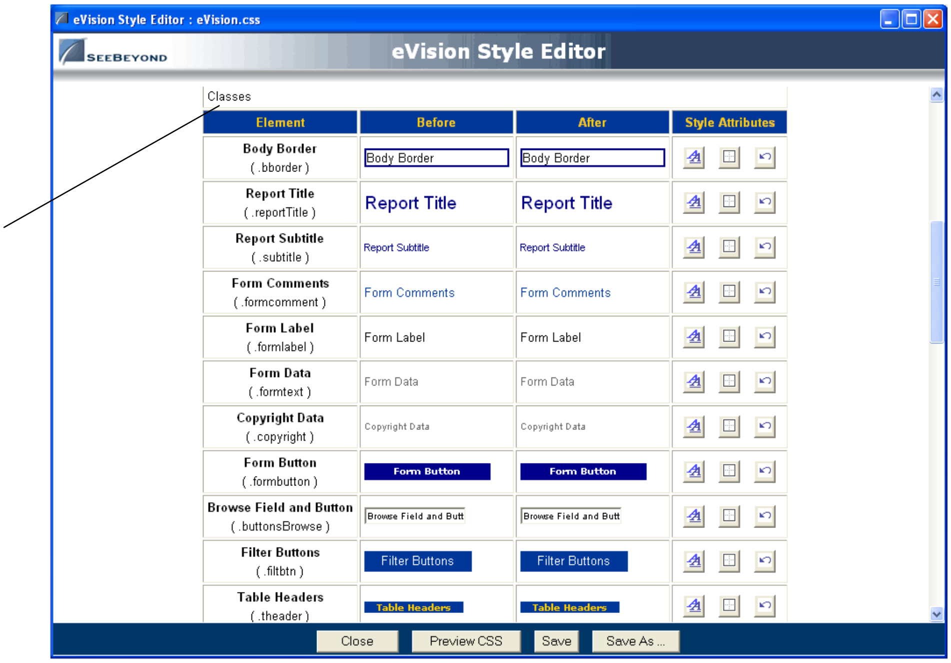
Select imported style sheet

4.6.1 Applying Classes

You can also apply *classes* of specialized style elements. An *element class* is a special set of element attributes that can be applied to individual objects without disrupting the primary style sheet.

The style sheet that is packaged with eVision, **eVision.css**, provides a number of classes that are already defined and named. The default classes are accessible via the **Style Editor** under the **Classes** heading. To see the class section in the editor, use the scroll bar and scroll down until the **Classes** heading appears. See Figure 39.

Figure 39 Style Sheet with Classes



Classes heading

Classes

Element	Before	After	Style Attributes
Body Border (.bborder)	Body Border	Body Border	
Report Title (.reportTitle)	Report Title	Report Title	
Report Subtitle (.subtitle)	Report Subtitle	Report Subtitle	
Form Comments (.formcomment)	Form Comments	Form Comments	
Form Label (.formlabel)	Form Label	Form Label	
Form Data (.formtext)	Form Data	Form Data	
Copyright Data (.copyright)	Copyright Data	Copyright Data	
Form Button (.formbutton)	Form Button	Form Button	
Browse Field and Button (.buttonsBrowse)	Browse Field and Butt	Browse Field and Butt	
Filter Buttons (.filtbtn)	Filter Buttons	Filter Buttons	
Table Headers (.theader)	Table Headers	Table Headers	

You can modify classes when you create or modify a style sheet. Classes cannot be applied until the style sheet is imported into the Project. (See **"Importing a Style Sheet Into Your Project" on page 53**.) After importing, follow the instructions in **"Applying a Class to an Object"**.

Applying a Class to an Object

To apply a class to an HTML or Form object, do the following:

1. On the design canvas, select the HTML or Form object to which you want to add the attributes of a class.

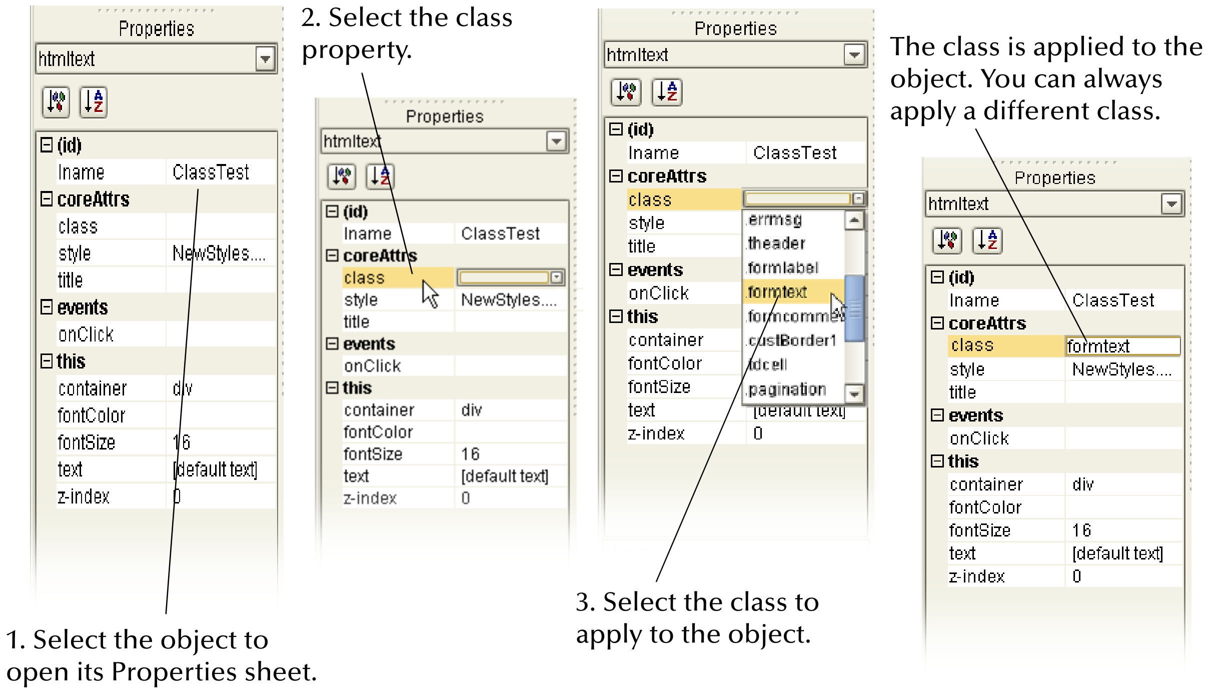
   The **Properties** sheet displays the object's properties.

2. In the **Properties** sheet, select the **class** property.

   A drop-down menu lists the classes that are available in the style sheet.

3. Select a class name from the list to apply the class to the object. See Figure 40.

Figure 40 Mapping a Class to a Component



1. Select the object to open its Properties sheet.

Properties	htmltext
(id)	
lname	ClassTest
coreAttrs	
class	
style	NewStyles....
title	
events	
onClick	
this	
container	div
fontColor	
fontSize	16
text	[default text]
z-index	0

2. Select the class property.

Properties	htmltext
(id)	
lname	ClassTest
coreAttrs	
class	
style	NewStyles....
title	
events	
onClick	
this	
container	div
fontColor	
fontSize	16
text	[default text]
z-index	0

3. Select the class to apply to the object.

Properties	htmltext
(id)	
lname	ClassTest
coreAttrs	
class	ermsg theader formlabel formtext formcomme custBorder1 tdcell pagination
style	
title	
events	
onClick	
this	
container	
fontColor	
fontSize	
text	[default text]
z-index	0

The class is applied to the object. You can always apply a different class.

Properties	htmltext
(id)	
lname	ClassTest
coreAttrs	
class	formtext
style	NewStyles....
title	
events	
onClick	
this	
container	div
fontColor	
fontSize	16
text	[default text]
z-index	0

4.7 Linking Pages

In eVision Web applications, links that point to internal application resources (other pages, applications, and so on), including "Home" page links, are created with the Page Link Wizard. The Page Link Wizard allows you to:

- Identify the start (Home) page of the Page Flow
- Create text links that connect Web pages and other resources to the Page Flow
- Specify parameters to pass to other objects on destination pages

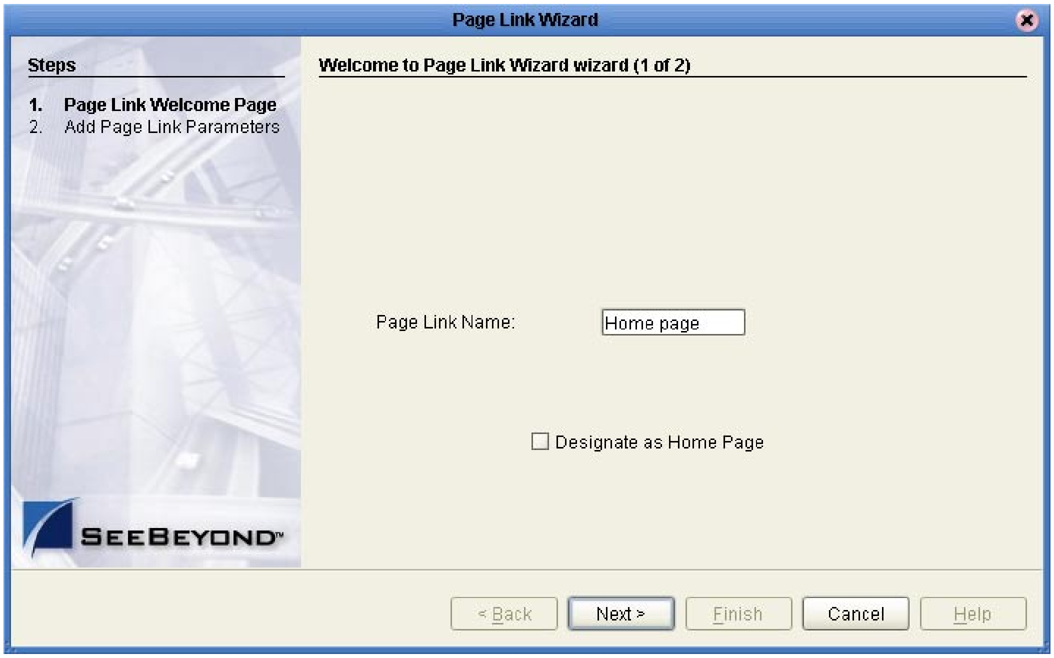
The Page Link Wizard is used to create links within the same Web application, which pass an event from a source page to a destination page. An event is any user interaction that triggers an application response; for example, selecting a radio button, or entering text in a text field.

To Create an Application Page Link

1. In the Project Explorer, right-click the Project.
2. On the context menu, click **New**, and then select **Page Link**.

   The **Page Link Wizard** appears. See Figure 41.

Figure 41 Page 1 of 2 in the Page Link Wizard



Page Link Wizard

Welcome to Page Link Wizard wizard (1 of 2)

Page Link Name: Home page

☐ Designate as Home Page

3 In the **Page Link Name** field, type a descriptive name for the link.

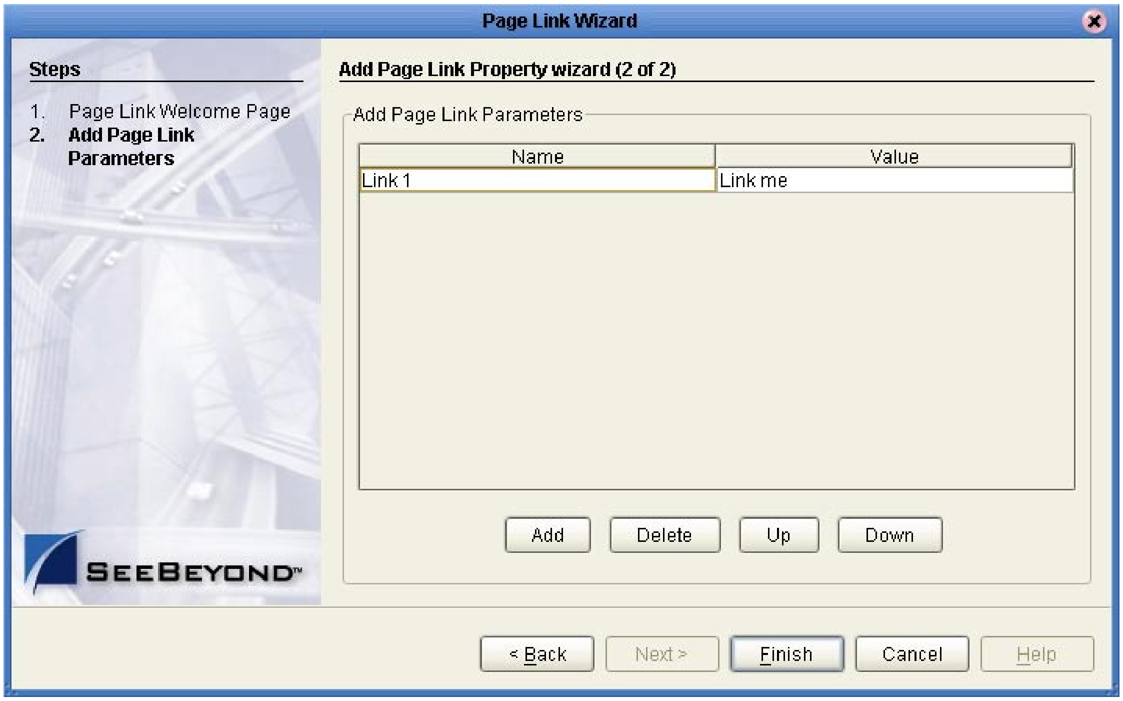
This is the internal application link name, used by the system to identify the link.

4 If you want the link to be defined as the link to the Page Flow's Home page, check **Designate as Home Page**.

The Home page designation specifies that the page will be the first page that users see when the Web application is accessed via its URL. In a Page Flow, only one page can be designated as the Home page.

5 Click **Next**. Step 2 of the wizard appears. See Figure 42.

Figure 42 Page 2 of 2 in the Page Link Wizard



Name	Value
Link 1	Link me

6 Add Page Link parameters.

A Page Link parameter is any data type that is passed via the link to an HTML or Form Object on the linked destination page. You can pass a single parameter or multiple sets of parameters.

On the **Add Page Link Parameters** page:

- **Name** is the parameter's name (the internal system name used when specifying a target for the link).
- **Value** is the data to be passed via the link to a target object on the destination page.

Note: *A value must be assigned to the page link, even if it is dynamically assigned at runtime. Do not leave the Value field blank.*

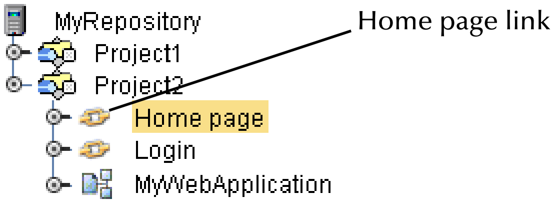
7 Modify the Page Link parameter fields, if necessary:

- To add a Name/Value parameter field to the stack, click **Add**.
- To delete a Name/Value parameter field from the stack, click the field and then click **Delete**.
- To navigate up one level in a stack of parameter fields, click **Up**.
- To navigate down one level, click **Down**.

8 To save the Page Link, click **Finish**.

The Link is added to your Project in the Project Explorer tree. See Figure 43.

Figure 43 Links in Enterprise Explorer



You can drag a page link from the Explorer tree and drop it directly onto a page in a Page Flow. In a Page Flow, page links can be used as follows:

- To link pages and Activities within the process
- As a Receive operation
- As an Activity in an Event-based decision

For information on using Page Links in a Page Flow, see **“Creating a Page Flow” on page 81**.

Chapter 5

Using the Page Layout Designer

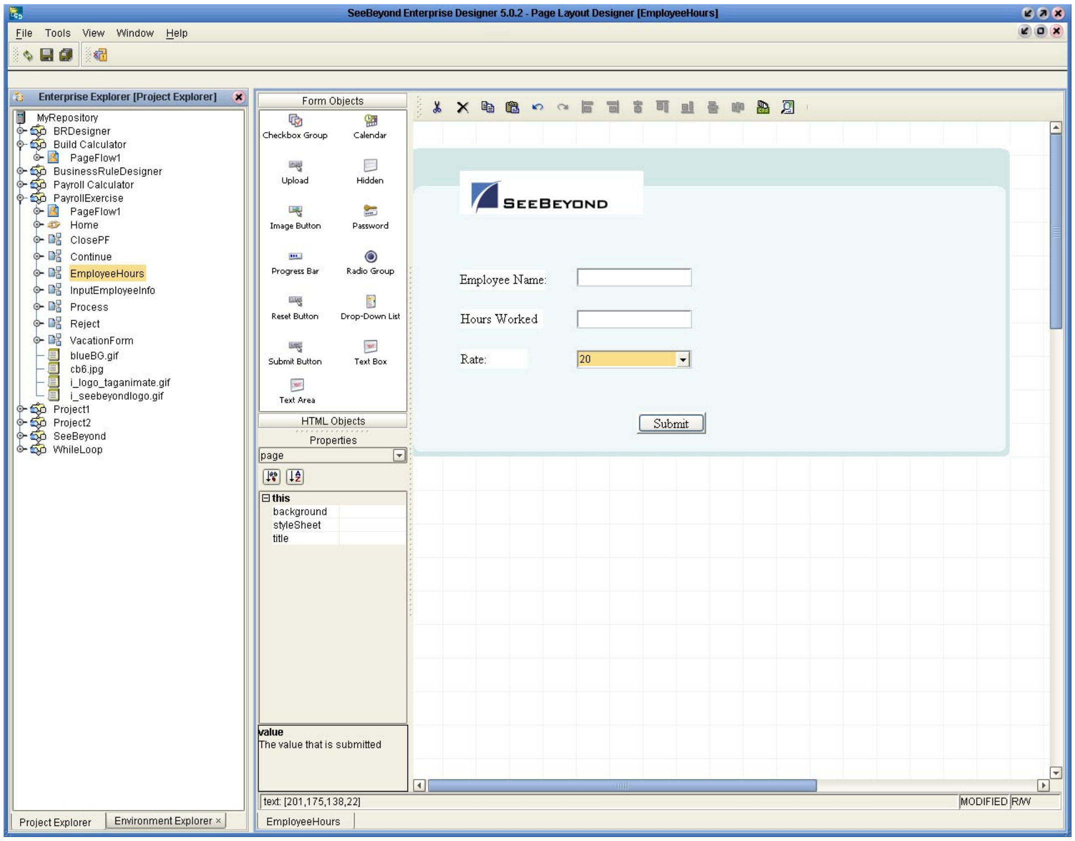
This chapter guides you through the process of starting a new page, placing objects on the canvas, using the property sheets, and changing the size and location of objects.

This Chapter Includes:

5.1 Laying Out a Web Page

In the following exercise, you build an input page for a small payroll calculation system. The purpose of the exercise is to create a finished Web page that you can preview in your browser. In the process of constructing the page, you will gain functional familiarity with eVision Studio's Page Layout Designer tools. For an example of the finished page, see Figure 44.

Figure 44 The Page Layout Designer Web Page Project



Employee Name:

Hours Worked

Rate: 20

value
The value that is submitted

text: [201,175,138,22]

MODIFIED R/W

5.1.1 Downloading the Image Files

The sample project files are provided in .zip file format and were installed on the Repository server when you uploaded the eVision Studio documentation **.sar** file. The first step in building the sample page is to download and import two sample image files from this file set.

The Repository server must be running before you can use Enterprise Manager or Enterprise Designer. The first step of any procedure is to make sure the server is running, and if not, to start it. The *SeeBeyond ICAN Suite Installation Guide* describes how to start the Repository server.

To Download the Project Archive Files

Before you can import the sample image files, you must download the project archive files onto your computer.

1. Start Internet Explorer.
2. In the **Address** field, enter **http://*hostname*:*portnumber***

   where

   hostname is the TCP/IP host name of the server where the Repository is installed.

   portnumber is the base port number of the Repository.

3. Enter your username and password.

   The **SeeBeyond Customer Login** window of Enterprise Manager appears.

4. Click **Login**.

   The Enterprise Manager home page appears.

5. Click the **DOCUMENTATION** tab.

   The Documentation page provides access to the ICAN Suite technical documentation.

6. In the left frame of the Documentation page, click **eVision Studio**.

Note: *Documentation is uploaded in a process that is separate from the product uploads. If you do not see an eVision Studio link in the left frame, then the eVision Studio Documentation* ***.sar*** *file must be uploaded. See* **Chapter 3**, **"Installing eVision Studio" on page 21**.

7. In the right frame of the Documentation page, click **Download Sample**.

8. Using an archive utility (such as WinZip), extract the sample files to a folder on your computer. Remember where you saved the files.

   The extraction process places the following files into the designated folder:

   - **blueBG.gif**
   - **i_seebeyondlogo.gif**
   - **eVisionSampleComponents.zip**
   - **eVisionTutorial_sample.zip**

5.1.2 Starting a New Page Layout

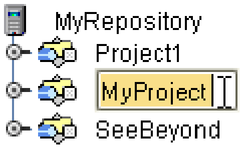
1. Start Enterprise Designer.

2. In the Project Explorer, right-click the Repository.

3. On the context menu, select **New Project**.

   A Project icon is added to the **Enterprise Explorer** tree. The new branch is called **Project*n***, where ***n*** is the next sequential default Project number. The keyboard focus is inside the **Project*n*** field. See Figure 45.

Figure 45 New Project in Project Explorer



4. Without moving the cursor, type a name for the new Project, for example, **MyProject**.

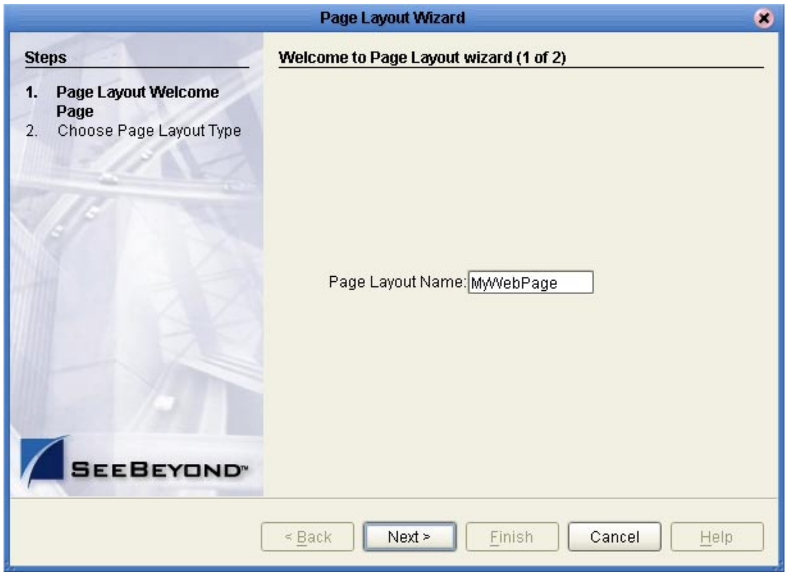
5 Press **Enter**.

6 Right-click the **Project** icon for your new Project.

7 On the context menu, click **New**, and then select **Page Layout**.

The Page Layout Wizard appears. See Figure 46.

Figure 46 Page Layout Wizard - Step 1



Page Layout Wizard

Steps

1. **Page Layout Welcome Page**
2. Choose Page Layout Type

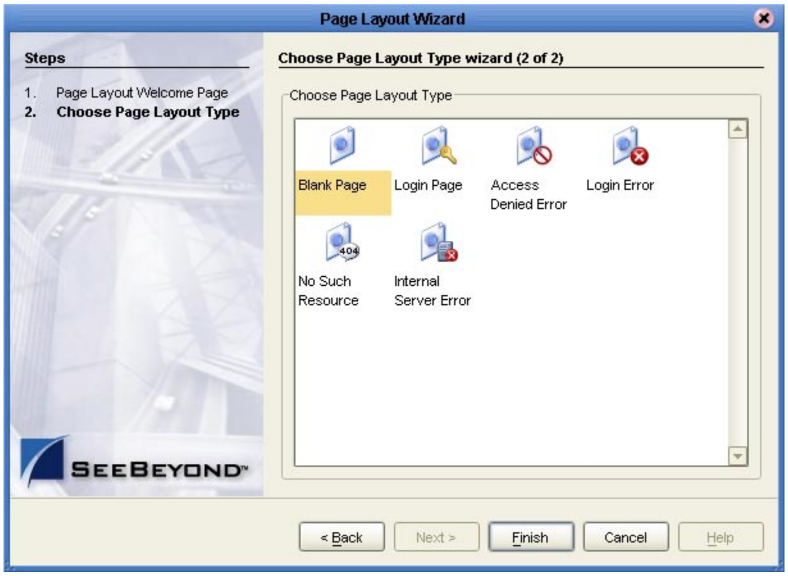
Welcome to Page Layout wizard (1 of 2)

Page Layout Name: MyWebPage

8 In the **Page Layout Name** box, type a unique name for the new Page Layout, for example, **MyWebPage**.

9 Click **Next**.

Figure 47 Page Layout Wizard - Step 2



Page Layout Wizard

Steps

1. Page Layout Welcome Page
2. **Choose Page Layout Type**

Choose Page Layout Type wizard (2 of 2)

Choose Page Layout Type

Step 2 of the Page Layout Wizard allows you to select any of the preconfigured Web pages based on a particular Project's requirements. For this Project, you start with a blank page.

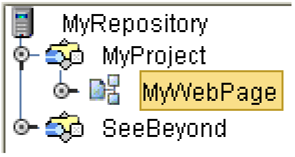
Note: *Chapter 8, "**Authentication and Error Handling**" describes the Login Page, Access Denied Error, Login Error, No Such Resource, and Internal Server Error templates.*

10 Click the **Blank Page** icon.

11 Click **Finish**.

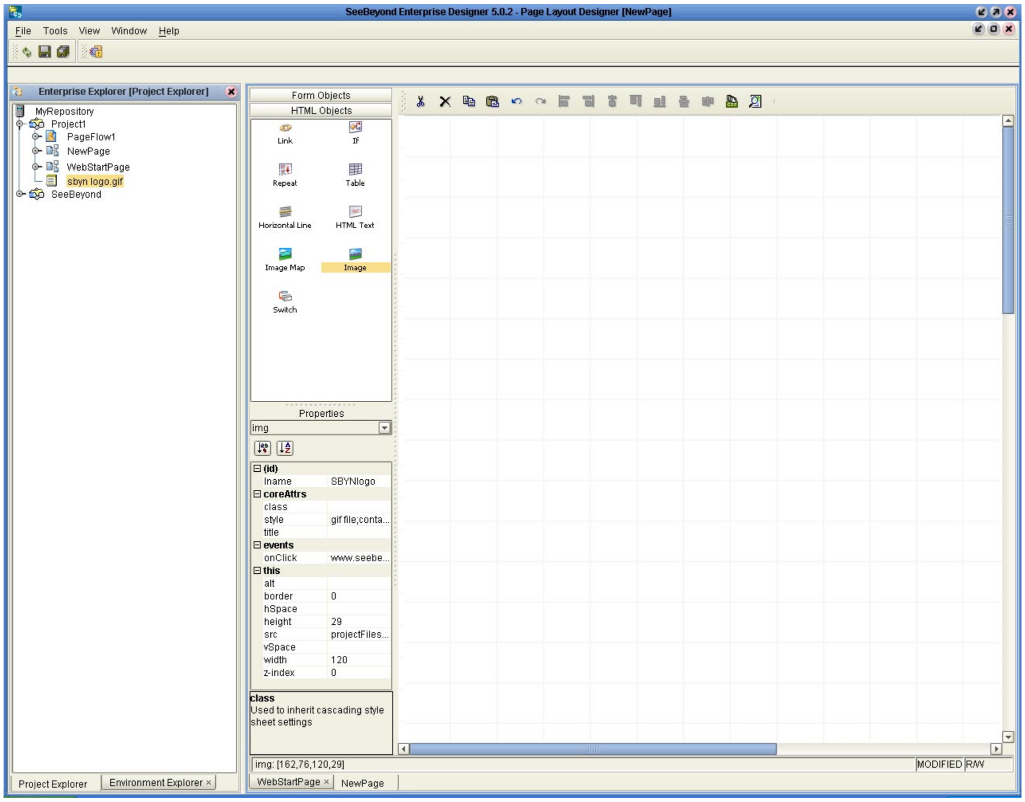
A new Page Layout icon is added to the Project Explorer tree. See Figure 48.

Figure 48 Page Layout Icon in Project Explorer



The name that you provided in the Page Layout Wizard is appended to the eVision application icon. The eVision Studio GUI appears in the work area of Enterprise Designer with a blank canvas. See Figure 49.

Figure 49 eVision GUI with a Blank Canvas



⊟ **(id)**	
Iname	SBYNlogo
⊟ **coreAttrs**	
class	
style	gif file;conta...
title	
⊟ **events**	
onClick	www.seebe...
⊟ **this**	
alt	
border	0
hSpace	
height	29
src	projectFiles...
vSpace	
width	120
z-index	0

class
Used to inherit cascading style sheet settings

img: [162,76,120,29]

You have just created a blank Web page and started the Page Layout Designer. You will now use the Page Layout Designer tools and pre-built GUI components to create a Web page that receives and passes user input.

Note: *Although it is not imperative now, before continuing the page layout procedure, you should import a Cascading Style Sheet (CSS). For instructions, see* **"Importing a Style Sheet Into Your Project" on page 53**.

5.1.3 Importing the Image Files

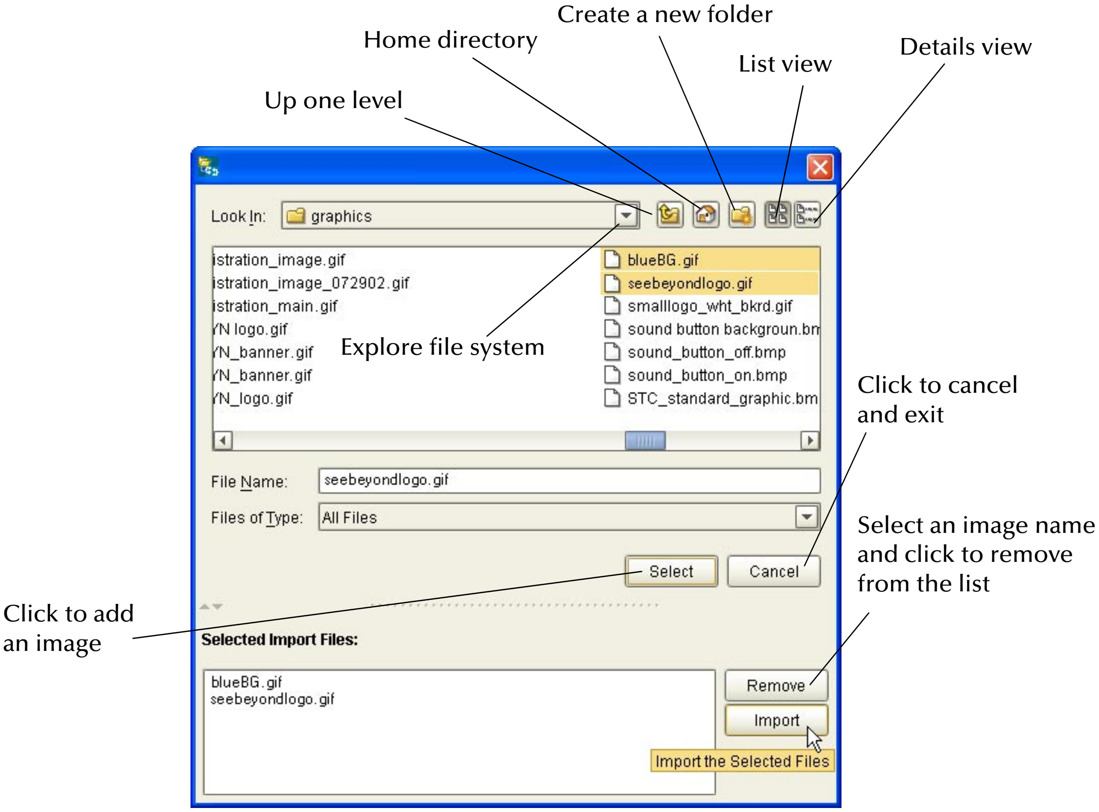
The sample Web page requires that you import two graphic images for use in the page layout: a background image and a SeeBeyond logo. Before graphic images can be used in an eVision Web page, you must import them into the Project.

To Import the Image Files

1. In the Project Explorer, right-click **MyProject**.
2. On the context menu, click **New**, and then select **File**.
3. Navigate to the folder containing the image files that you extracted from the sample project archive file. See Figure 50.

   Graphic images can reside in any directory that is accessible to your computer, locally or on a network.
4. Select **blueBG.gif** and click **Select**. The file name appears in the **Selected Import Files** box.
5. Click **i_seebeyondlogo.gif** and click **Select**. The file name appears in the **Selected Import Files** box.
6. Click **Import**.

Figure 50 Selecting the Image Files



Look In: graphics

istration_image.gif
istration_image_072902.gif
istration_main.gif
YN logo.gif
YN_banner.gif
YN_banner.gif
YN_logo.gif
blueBG.gif
seebeyondlogo.gif
smalllogo_wht_bkrd.gif
sound button backgroun.bm
sound_button_off.bmp
sound_button_on.bmp
STC_standard_graphic.bm

File Name: seebeyondlogo.gif

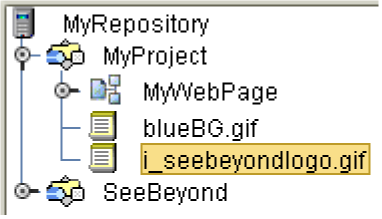
Files of Type: All Files

Selected Import Files:

blueBG.gif
seebeyondlogo.gif

A named icon for each image will be nested under the Project in the Project Explorer tree. See Figure 51.

Figure 51 Image Files in Project Explorer



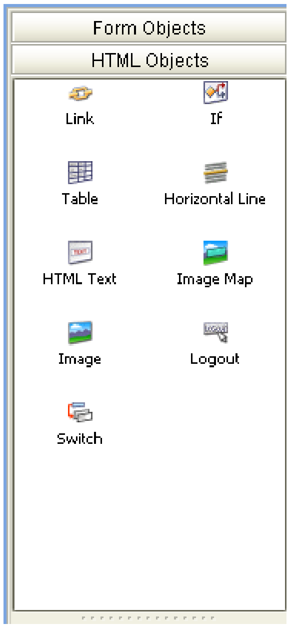
MyRepository
MyProject
MyWebPage
blueBG.gif
i_seebeyondlogo.gif
SeeBeyond

The images are now part of your Project.

5.1.4 Placing GUI Components on the Canvas

This section describes how to place GUI components on the canvas using the Component selection palettes. Before you begin, make sure that the **HTML Objects** palette is in front. See Figure 52.

Figure 52 HTML Objects Palette

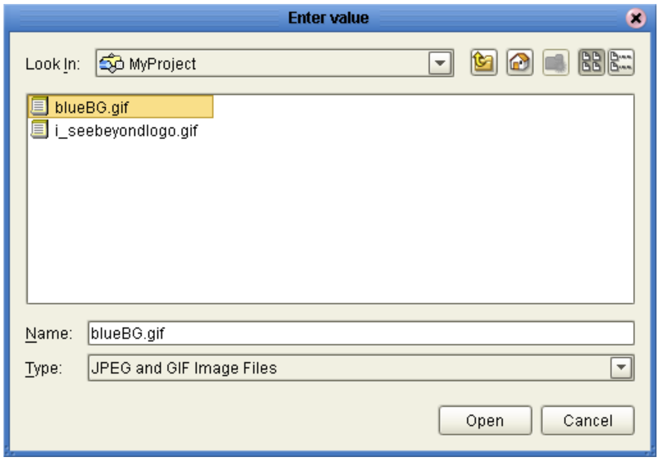


Create the Background Layer Component

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

   The **Enter value** dialog box appears. See Figure 53.

Figure 53 Enter value Dialog Box



Enter value

Look In: MyProject

blueBG.gif
i_seebeyondlogo.gif

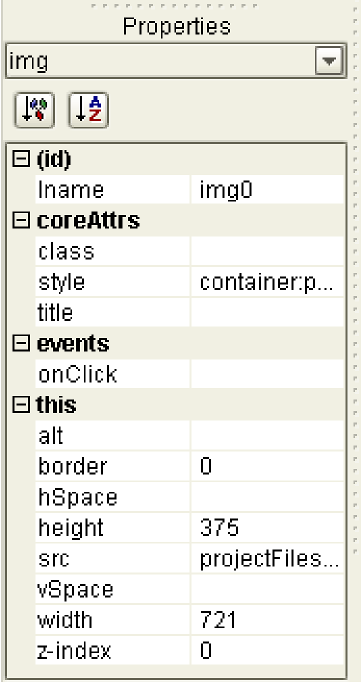
Name: blueBG.gif

Type: JPEG and GIF Image Files

2 Select the **blueBG.gif** file and click **Open**.

The **Image** component appears with a gold outline. The **Properties** sheet for the **Image** component is automatically displayed. See Figure 54.

Figure 54 Properties Sheet for the Image Component



Properties

img

⊟ **(id)**	
lname	img0
⊟ **coreAttrs**	
class	
style	container:p...
title	
⊟ **events**	
onClick	
⊟ **this**	
alt	
border	0
hSpace	
height	375
src	projectFiles...
vSpace	
width	721
z-index	0

3 In the **Properties** sheet, define the value for **lname** by doing the following:

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

B In the right column, delete the default value and type **Background**.

lname is the internal, logical system name for the component, used to identify the component in a Page Flow.

4 In the **Properties** sheet, define the value for **z-index** by doing the following:

A In the left column, select the **z-index** property.

B In the right column, delete the default value and type **-1**.

z-index specifies the placement of the object in the stack order. Setting the **z-index** to -1 enables you to place components on top of the background layer and have the components be immediately visible. For more information, see **"Moving Overlapping Components in the Z-Direction" on page 48**.

5 If you want to resize the component, click and drag a highlighted boundary element. The shape of the bidirectional arrow cursor indicates which way the component will be resized.

If you want to move the component, click the component in the center (away from the edges) and drag it to the new location.

Create the Page Banner Component

1. From the **HTML Objects** palette, drag the **Image** icon onto the canvas and inside the boundaries of the background image.

   The **Enter value** dialog box appears.

2. Select the **i_seebeyondlogo.gif** file and click **Open**.

   The **Properties** sheet for the **Image** component is automatically displayed.

3. In the **Properties** sheet, define the value for **lname** (the component name) by doing the following:

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

   B In the right column, delete the default value and type **logo**.

4. If you want to move the component, click the component in the center (away from the edges) and drag it to the new location.

Create the Employee Name Label Component

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

2. Place the component left-aligned with the **logo** component.

   The **Properties** sheet for the **HTML Text** component is automatically displayed.

3. In the **Properties** sheet, define the value for **lname** by doing the following:

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

   B In the right column, delete the default value and type **EmpName**.

4. In the **Properties** sheet, define the value for **text** by doing the following:

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

   B In the right column, click the **[default text]** field. The **Enter value for "text"** window appears.

   C Delete the **[default text]** string and type **Employee Name**.

   D Click **OK**.

   The text will be displayed to users on the Web page as a field label that prompts them to enter their name into the associated field on the page.

5. Initially, on the canvas, the text string may appear truncated. To see the entire text string:

   A Select the component.

   B Press and hold the Shift key.

   C Press the right arrow key until the entire text string is visible.

Create the Hours Worked Label Component

1. From the **HTML Objects** palette, drag the **HTML Text** icon onto the canvas.
2. Place the component under and left-aligned with the **EmpName** component.

   The **Properties** sheet for the **HTML Text** component is automatically displayed.
3. In the **Properties** sheet, define the value for **lname** by doing the following:

   A. In the left column, click the **lname** property.

   B. In the right column, delete the default value and type **HoursWorked**.
4. In the **Properties** sheet, define the value for **text** by doing the following:

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

   B. In the right column, click the **[default text]** field. The **Enter value for "text"** window appears.

   C. Delete the **[default text]** string and type **Hours Worked**.

   D. Click **OK**.

   The text will be displayed to users on the Web page as a field label that prompts them to enter the number of hours worked into the associated field on the page.
5. Initially, on the canvas, the text string may appear truncated. To see the entire text string:

   A. Select the component.

   B. Press and hold the Shift key.

   C. Press the right arrow key until the entire text string is visible.

Create the Rate Label Component

1. From the **HTML Objects** palette, drag the **HTML Text** icon onto the canvas.
2. Place the component under and left-aligned with the **HoursWorked** component.

   The **Properties** sheet for the **HTML Text** component is automatically displayed.
3. In the **Properties** sheet, define the value for **lname** by doing the following:

   A. In the left column, click the **lname** property.

   B. In the right column, delete the default value and type **HourlyRate**.
4. In the **Properties** sheet, define the value for **text** by doing the following:

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

   B. In the right column, click the **[default text]** field. The **Enter value for "text"** window appears.

   C. Delete the **[default text]** string and type **Rate**.

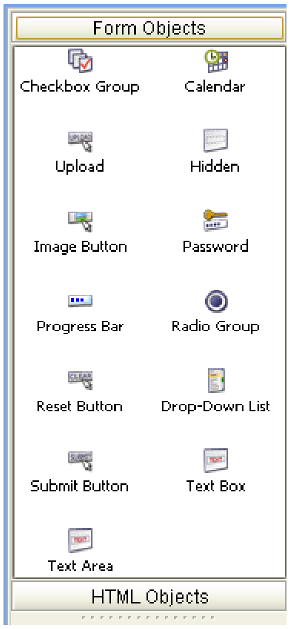
   D. Click **OK**.

   The text will be displayed to users on the Web page as a field label that prompts them to select their hourly pay rate from the associated drop-down list.

Swap the HTML Objects and Form Objects Palettes

To bring the **Form Objects** palette to the front, click the **Form Objects** title bar. See Figure 55.

Figure 55 Form Objects Palette



Create the Employee Name Input Field Component

1 From the **Form Objects** palette, drag the **Text Box** icon onto the canvas.

2 Align the component horizontally with and to the right of the **EmpName** component. Make sure that the components' vertical edges do not overlap.

The **Properties** sheet for the **Text Box** component is automatically displayed.

3 In the **Properties** sheet, define the value for **lname** by doing the following:

A In the left column, click the **lname** property.

B In the right column, delete the default value and type **EmpNameInput**.

4 In the **Properties** sheet, define the value for **value** by doing the following:

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

B In the right column, delete the **[default text]** string. (You want this input field to be initially blank, to receive input from users.)

This component will be displayed to users as an empty input field to the right of the **Employee Name** label.

Create the Hours Worked Input Field Component

1. From the **Form Objects** palette, drag the **Text Box** icon onto the canvas.
2. Align the component horizontally with and to the right of the **HoursWorked** component. Make sure that the components' vertical edges do not overlap.

   The **Properties** sheet for the **Text Box** component is automatically displayed.
3. In the **Properties** sheet, define the value for **lname** by doing the following:

   A In the left column, click the **lname** property.

   B In the right column, delete the default value and type **HoursWorkedInput**.
4. In the **Properties** sheet, define the value for **value** by doing the following:

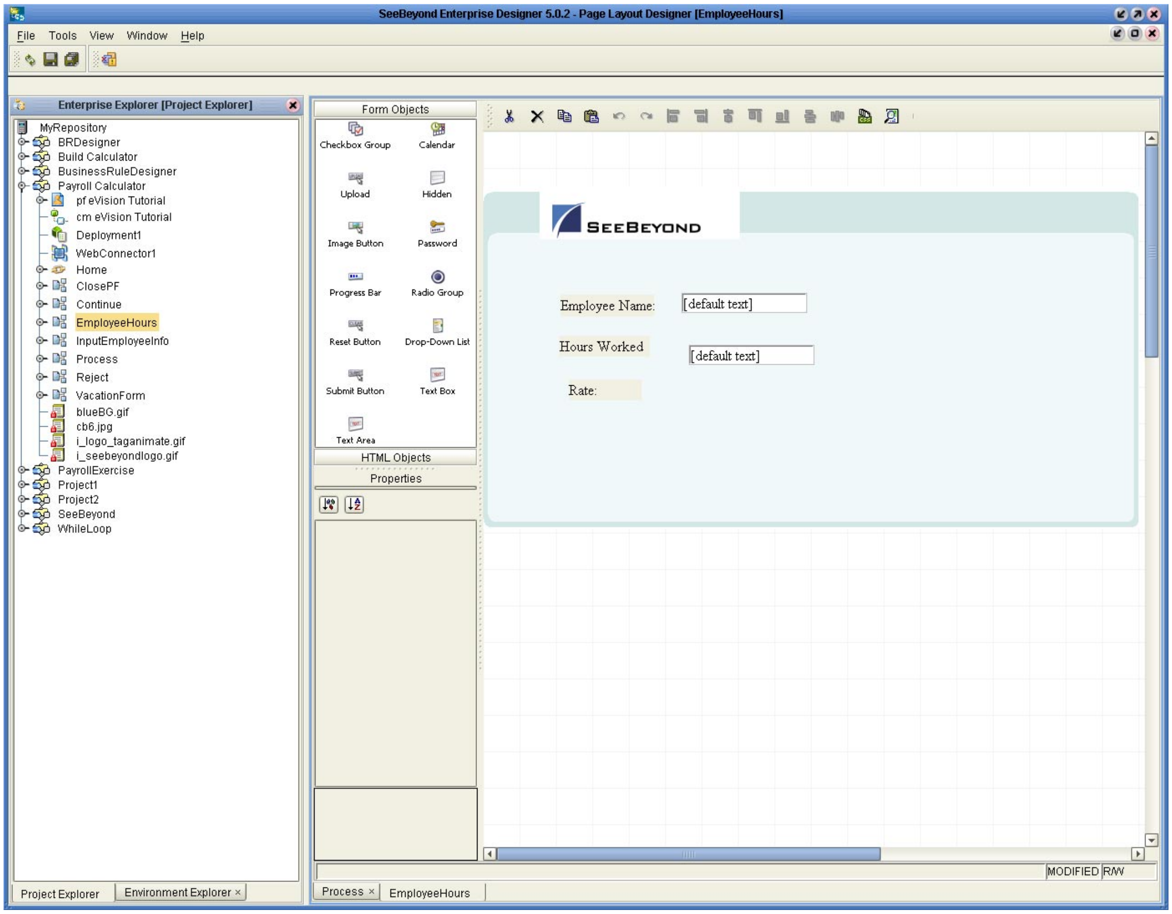
   A In the left column, select the **value** property.

   B In the right column, delete the **[default text]** string. (You want this input field to be initially blank, to receive input from users.)

   This component will be displayed to users as an empty input field to the right of the **Hours Worked** label.

Check your progress. The unfinished canvas should look like the example in Figure 56.

Figure 56 Incomplete Page



Employee Name: [default text]

Hours Worked [default text]

Rate:

Create the Rate Drop-Down List Component

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

2 Align the component horizontally with and to the right of the **HourlyRate** component. Make sure that the components' vertical edges do not overlap.

The **Properties** sheet for the **Drop-Down List** component is automatically displayed.

3 In the **Properties** sheet, define the value for **lname** by doing the following:

A In the left column, click the **lname** property.

B In the right column, delete the default value and type **HourlyRateDropDown**.

4 On the canvas, right-click the **Drop-Down List** component.

5 On the context menu, select **Edit Options**.

6 Under the **Labels** heading, click the empty input field and type **20**. This is the default pay rate that will be displayed to users in the drop-down list on the Web page. See Figure 57.

7 Under the **Values** heading, click the empty input field and type **20.** This is the label for the drop-down list entry.

8 To add three additional input rows:

A Click **Add** three times in succession.

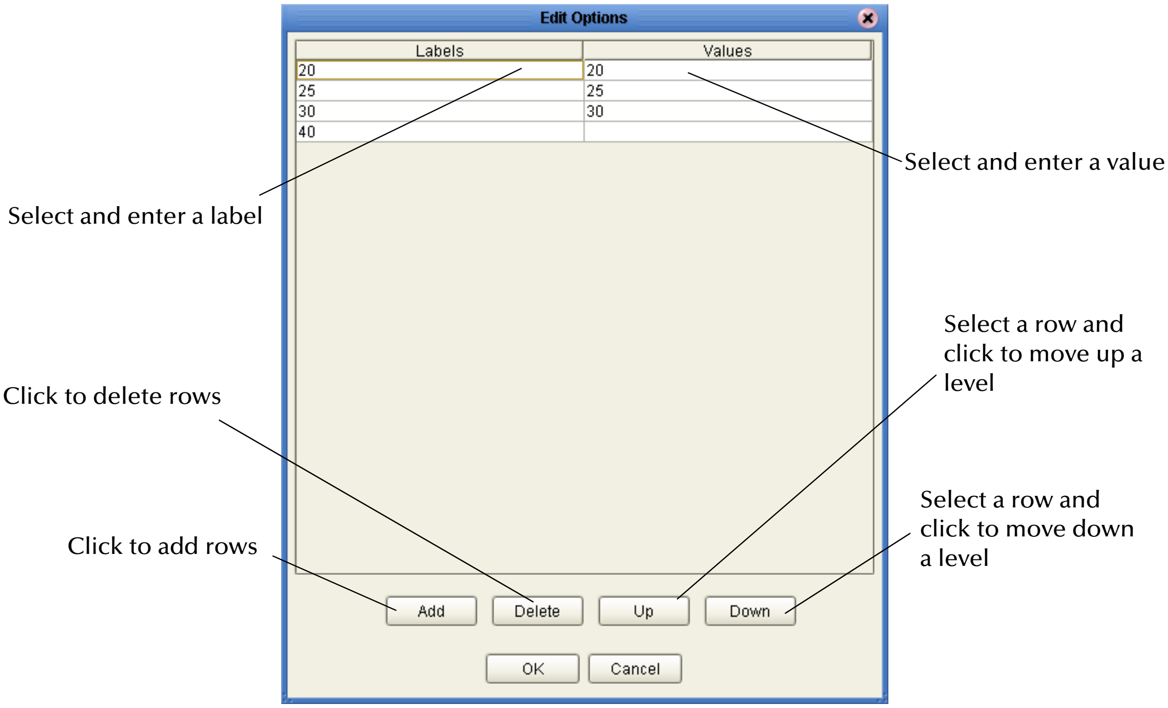
B In row 2, select the **Labels** field and type **25** and repeat in the **Values** field.

C In row 3, select the **Labels** field and type **30** and repeat in the **Values** field.

D In row 4, select the **Labels** field and type **40** and repeat in the **Values** field.

The **Edit Options** dialog box should now look like Figure 57.

Figure 57 Edit Options Dialog Box



Edit Options

Labels	Values
20	20
25	25
30	30
40	

Select and enter a label

Select and enter a value

Click to delete rows

Select a row and click to move up a level

Click to add rows

Select a row and click to move down a level

9 Click **OK**.

Create the Submit Button Component

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

2 Place the component below and right-aligned with the **Drop-Down List** component.

The **Properties** sheet for the **Submit Button** component is automatically displayed.

3 In the **Properties** sheet, define the value for **Iname** by doing the following:

A In the left column, click the **Iname** field.

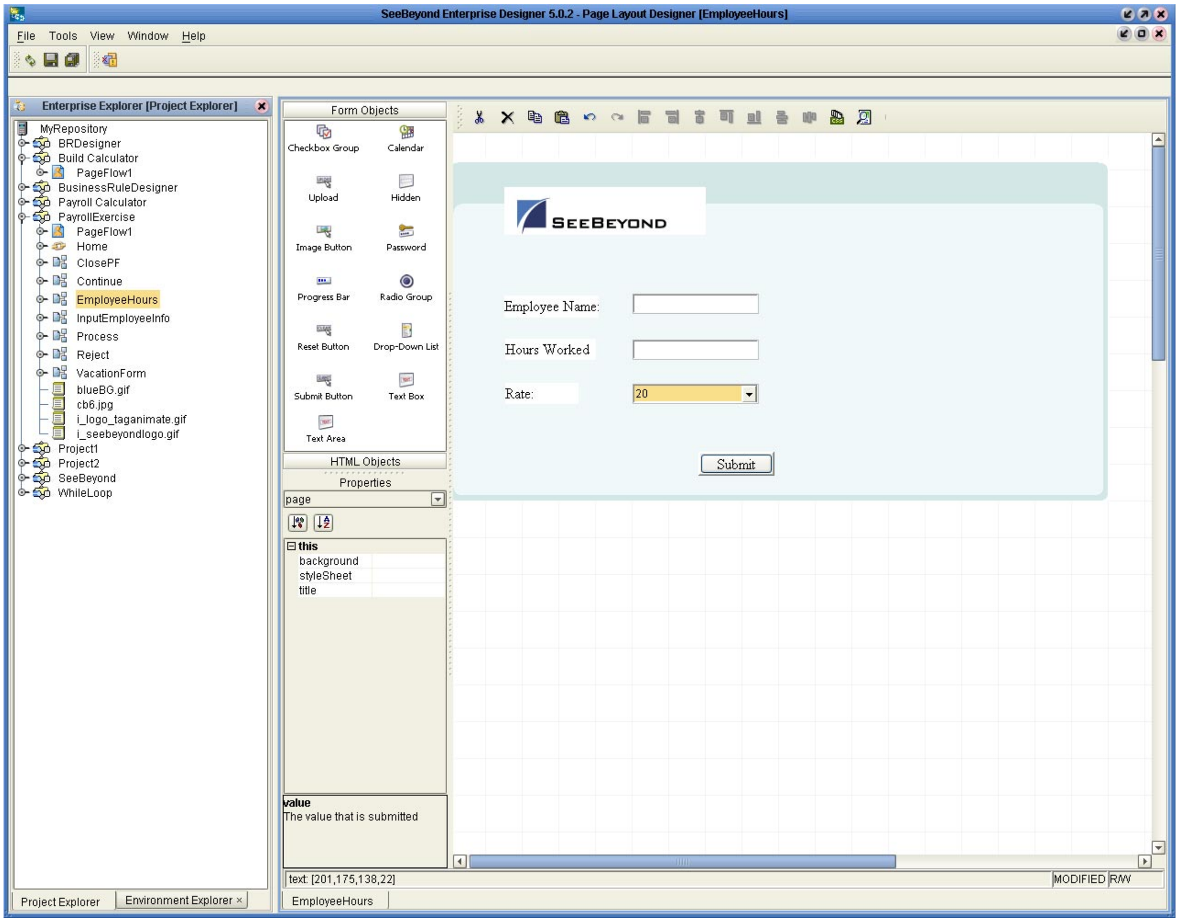
B In the right column, delete the default value and type **Submit**.

This component will be displayed to users as a button with the label **Submit**.

5.1.5 Reviewing the Page Layout

The Input page should look like the example in Figure 58.

Figure 58 Input Page



5.1.6 Aligning Objects Using the Alignment Tools

If the GUI components are not precisely aligned, or if the space between the objects is inconsistent, you can adjust the alignment of the components using the Alignment tools.

To choose two or more objects to be aligned, do the following:

1. Press and hold the Shift key on your computer.
2. Select the objects that you want to align.
3. Click the desired alignment tool on the Page Layout Designer toolbar. Alignment will occur as described in Table 5.

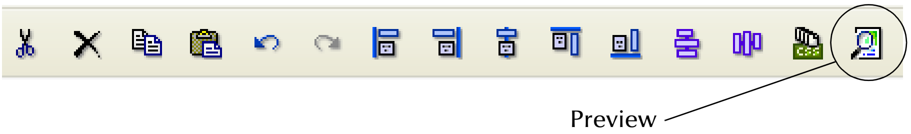
Table 5 eVision Studio Object Alignment Tools

Tool	Name	Purpose
	Left Align	Aligns two or more selected components with the left-most component in the selected group.
	Right Align	Aligns two or more selected components with the right-most component in the selected group.
	Center Align	Aligns two or more selected components with the center of the canvas (the center of the Web page).
	Top Align	Aligns two or more selected components with the top-most component in the selected group.
	Bottom Align	Aligns two or more selected components with the bottom component in the selected group.
	Vertical Spacing	Creates equal vertical spacing between object in a group based on an averaging algorithm.
	Horizontal Spacing	Creates equal horizontal spacing between objects in a group based on an averaging algorithm.

5.1.7 Previewing the Finished Web Page

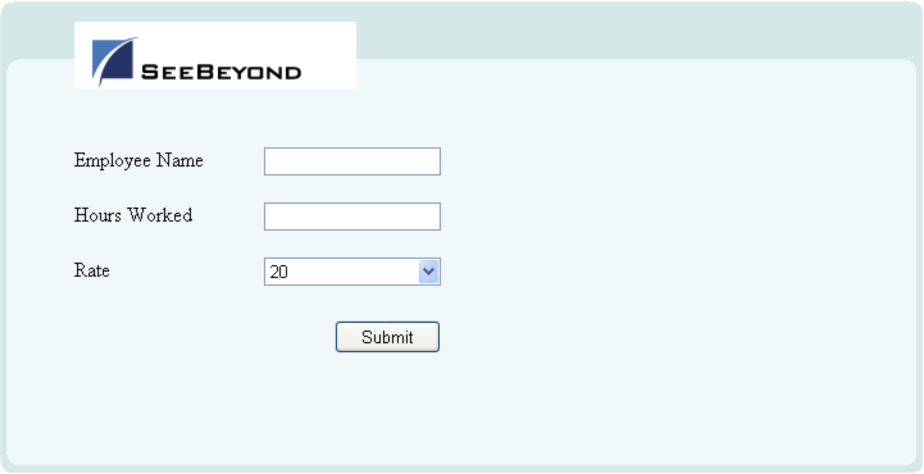
To preview the finished Web page in your browser, click the **Preview** icon on the Page Layout Designer toolbar.

Figure 59 Page Layout Designer Toolbar



The page should look like the example in Figure 60.

Figure 60 Web Page Preview



Employee Name	
Hours Worked	
Rate	20

5.1.8 Saving the Project in the Repository

On the **File** menu, click **Save All**. The system saves the Project in the Repository.

5.2 Checking the Page Layout into Version Control

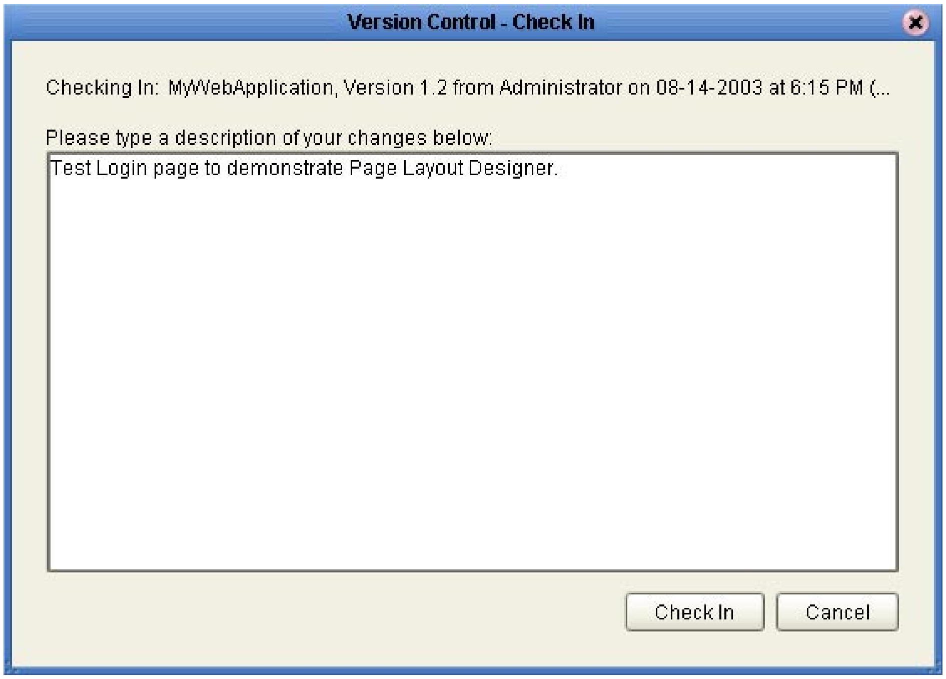
The Enterprise Designer Version Control system allows you to track changes and also prohibits simultaneous check-out of a Page Layout. When a page is checked out, it is locked and unavailable to other users.

For more information on the Version Control system, see the *eGate Integrator User's Guide*.

To Check In the Page Layout

1. In the Project Explorer, right-click the Page Layout name.
2. On the context menu, select **Check In**. The **Version Control - Check In** dialog box appears.
3. Type a comment to accompany the Page Layout. See Figure 61.

Figure 61 Version Control - Check In Dialog Box



Version Control - Check In

Checking In: MyWebApplication, Version 1.2 from Administrator on 08-14-2003 at 6:15 PM (...

Please type a description of your changes below:

Test Login page to demonstrate Page Layout Designer.

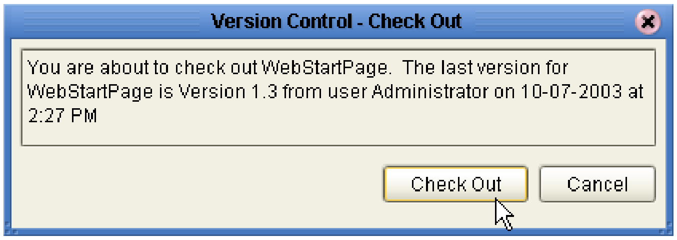
4 Click **Check In**. A lock icon appears in the component.

To Check Out a Page Layout

1 In the Project Explorer, right-click the Page Layout name.

2 On the context menu, select **Check Out**. The **Version Control - Check Out** dialog box appears. See Figure 62.

Figure 62 Version Control - Check Out Dialog Box



Version Control - Check Out

You are about to check out WebStartPage. The last version for WebStartPage is Version 1.3 from user Administrator on 10-07-2003 at 2:27 PM

3 Click **Check Out**.

5.3 Manipulating Objects on the Canvas

When you are viewing the page in Preview mode, some components may not be located exactly where you want them, or a component may be too large or too small. For example, if two horizontal lines are not of equal length, and they are supposed to be separate but identical, this may detract from the overall composition of the page layout. This section describes the various ways you can manually move and resize page components.

5.3.1 Moving a Single Object

To move a single object to any location, select the object. When the object is active, it will be highlighted with a color border. You can click-and-drag the object to any location on the page using the mouse. When the object is where you want it, release the mouse button.

You can also move an object using the arrow keys on your computer keyboard. To release the object and place it, when it is in its new location, select the canvas in a blank area.

5.3.2 Moving a Temporary Group of Objects

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

- Select the canvas in a blank area adjacent to the objects you want to move, and drag the cursor over both objects. Your mouse movement will trace a bounding box for visual reference. You can surround the objects you want to move, or more simply, one edge of the bounding box must intersect with an object to add it to the group.
- Press and hold the Shift key on your computer and sequentially select the objects you want to move.

When the objects are selected, click-and-drag one of the objects and the others will follow. You can also move a group of objects using the arrow keys on your computer keyboard. To release the objects and place them, when they are in their new location, select the canvas in a blank area.

5.3.3 Resizing Objects

Objects can be resized vertically or horizontally, depending on the attributes of the object. This exercise describes horizontal resizing.

Select the object or objects as described in **"Moving a Single Object" on page 79**, and **"Moving a Temporary Group of Objects" on page 79**. Move the pointer over the left or right border of the object. The pointer will change to a bidirectional arrow. Click-and-drag the border of the object to expand it horizontally.

Resizing Horizontal Lines

To resize a horizontal line, do either of the following:

- To modify a line horizontally, click the line at either end. When the line is highlighted, drag the bidirectional cursor arrow to the right or left to shrink or stretch the line in either direction. To finish, click the canvas anywhere outside the line object.
- To modify line thickness, click the line at either end. When the line is highlighted, drag the bidirectional cursor arrow diagonally or vertically to increase or decrease the line’s thickness. To finish, click the canvas anywhere outside the line object.

Chapter 6

Page Flow Designer

To build a functioning Web application, you first create the individual user-facing pages that you want users to interact with. You then use eVision Studio's Page Flow Designer to link the Web pages together and create a logical chain called a Page Flow.

This Chapter Includes:

6.1 Creating a Page Flow

A Page Flow is a structured series of Web pages that comprise a Web-enabled business process. A Web-enabled process can be an internal service, or 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 an empty Page Flow canvas that enables you to place pages and other design (sub-process) elements and then connect them together in a sequence. Before you can start building a Page Flow, you must first add the Page Flow to your Project.

To Add a Page Flow to a Project

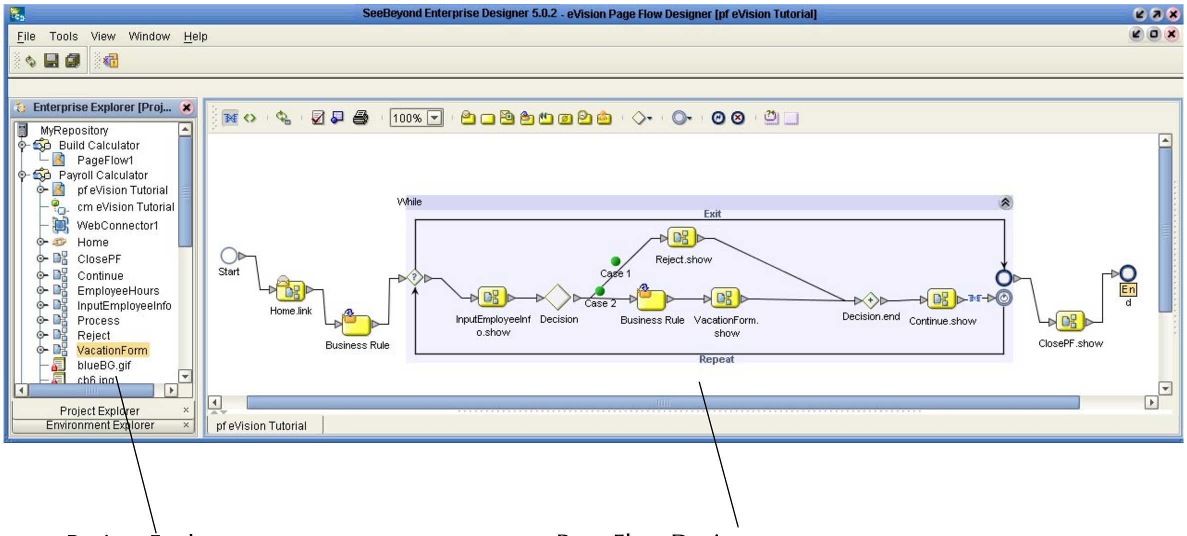
1. In the Project Explorer, right-click the Project.
2. On the context menu, click **New**, and then select **Page Flow**.
3. If desired, change the default name of the Page Flow.

6.1.2. Adding Web Pages to a Page Flow

To create a Page Flow, you drag and drop Page Flow elements on the canvas to visually create a flowing design for the flow of the Web application. Then, from the Project Explorer, you drag and drop Web pages onto Page Flow elements, and then link the collection of elements together to create a logical page flow for the application.

The Page Flow Designer provides the tools to lay out and connect Page Flow elements and other specialized sub-process elements, including editing capabilities that allow you to adjust and align Web page and sub-process components. Figure 63 shows an example of the Page Flow Designer GUI.

Figure 63 Page Flow in the Page Flow Designer



Project Explorer

Page Flow Designer canvas

When you start a new Web application project, the **Start** and **End** icons automatically appear on the blank Page Flow Designer canvas. There can be only one starting point for a Page Flow; however, there can be multiple end points.

To Create an Initial Page Flow

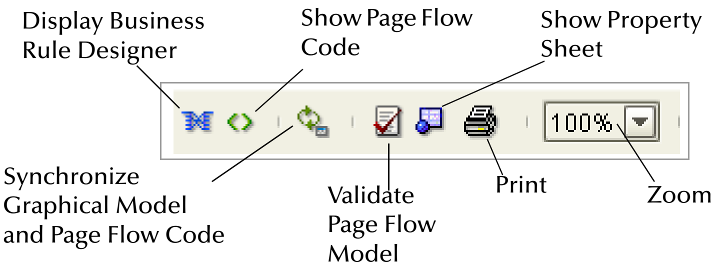
1. Drag the desired Page Flow elements onto the Page Flow Designer canvas.
2. Create links between the elements to create the Page Flow.
3. From the Project Explorer, drag Web page icons onto appropriate Page Flow elements.
4. Save the Page Flow to the Repository.

   The **Save** action validates the connectivity of the Page Flow, generates the code to run it, and saves the Page Flow in the Repository.

6.2 Page Flow Designer Tools

The Page Flow Designer tools are shown in Figure 64.

Figure 64 Page Flow Designer Toolbar


Display Business Rule Designer
Show Page Flow Code
Show Property Sheet
100%
Synchronize Graphical Model and Page Flow Code
Validate Page Flow Model
Print
Zoom

The Page Flow Designer tools are described in Table 6.

Table 6 Page Flow Designer Tools

Tool	Purpose	Description
	Display Business Rule Designer	Opens a window in the lower portion of the Page Flow Designer screen 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 screen and displays the 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 execution errors in the Page Flow code.
	Show Property Sheet	Allows you to set alert and logging properties for the Page Flow.
	Print	Prints the Page Flow screen image. Allows you to control the scale of the printed image.
100% ▾	**Zoom**	Increases or decreases the Page Flow image on the screen.

6.3 Page Flow Elements

The Page Flow Designer provides a palette of elements that allow you to customize and extend a Page Flow. Pages are dragged from Enterprise Explorer and dropped onto the Design canvas. In addition to Web pages, Page Flows can consist of combinations of process elements, branching elements, and intermediate events.

6.3.1. Sub-process Elements

You can add several types of elements to a Page Flow. See Table 7.

Table 7 Connecting and Sub-process Elements

Button	Command	Function
○	Start Node	The **Start Node** indicates the start of the process. The Start node is automatically added to the canvas when new Page Flow is initialized. 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. When you select a link, a context menu allows you to configure how data is going to be passed to and from the underlying component or Web Service using Attributes. The Page Flow Designer ensures that the Page Flow is correctly linked by rejecting invalid links. Links can also accept mapped values. A link with mapped values will displayed with the “map” icon.
◯	End Node	The **End Node** indicates the completed state of a Page Flow. The End node is automatically added to the canvas when a new Page Flow is initialized.
✉	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 will invoke a Web Service or an eGate Component. Depending on the configuration of the component, a response may or may not be required. One example would be a synchronous extraction process from a database to return the credit status of a trading partner.

Button	Command	Function
	Reply	The **Reply** element allows 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 business process is paired with the Reply at the end of the process. In cases where a message must be sent back to the caller of the process, the Reply uses information that correlates the message in the calling system. A Reply acts as the last step in a Page Flow, in which the process is acting 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.
	Assign	The **Assign** element sets data values, including task assignments. It 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.
	Empty	This **Empty** element allows data to pass through without changes.
	Wait	The **Wait** element acts as a timer. If the user builds a Page Flow in which there are two simultaneous paths within a set framework (one for the page flow, one for the timer, if the timer condition takes place first, an exception will be thrown, handled, and the Page Flow will then be abandoned.

6.3.2. Branching Elements

Branching elements are objects that you add to a Page Flow to specify the logical flow of information. eVision provides three types of branching elements: Decision, Event Based Decision, and Flow. See Table 8.

Table 8 Branching Elements

◇	Decision	The **Decision** element allows 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 allows you to define expressions that are evaluated to determine the routing of the Page Flow. Expressions are built using the mapping interface and Page Flow attributes. ***Note:*** *The Decision element is structured to automatically raise a run-time Exception to alert you to Page Flow errors during construction. You can set this value to Return True or Return False; however, it is not recommended during the construction phase; errors can be hidden during development and only surface in the post-deployment, run-time environment.*
◇	Event Based Decision	The **Event Based Decision** element allows one of several possible paths to execute, based on which link the user has selected.
◇	Flow	The **Flow** element specifies that one or more pages and/or processes are to flow concurrently.

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.

6.3.3. Intermediate Events

Intermediate events are elements that can interrupt a Page Flow. Some intermediate events handle exceptions that may occur at run-time or compensate for exceptions. See Table 9.

Table 9 Intermediate Events

	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 allows you to design the process and circumstances, in which the compensation takes place.
	Catch Named Exception	Each automated system (backend system) or Web service can publish their possible error codes (for instance, fault 15 is "bad data"). Those 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 will throw the exceptions when errors occur and the exception handler will automatically initiate the appropriate process to handle the problem.
	Catch All Exceptions	The **Catch All Exceptions** handler is configured to handle all exceptions.
	Message Event	The **Message Event** is similar to a Receive Activity, but it occurs only in the middle of a process. Each of these elements can be a different message.
	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 allow the creation of the process that takes place after a time-out condition takes place.
	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 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.

6.3.4. While

A **While** loop allows you to encapsulate all or part of a Page Flow within a looping process. See Table 10.

Table 10 While loop

	While	The **While** loop 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.

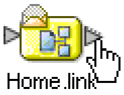
6.3.5. Links

In a Page Flow, you connect Page Flow elements manually using links. Links are used to create the flow between Pages.

To Link Web Pages and Other Page Flow Elements

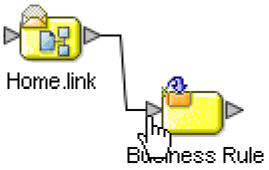
1. Move your cursor over the connector portion of the Page Flow element.
2. Hold the cursor over the outside edge of the element until it changes from the arrow pointer to the hand pointer. See Figure 65.

Figure 65 Starting a Link



3. Drag a line from the forward connector on the first Web page to the connector on the second Web page or design element, then release the mouse. See Figure 66.

Figure 66 Finished Link



Note: *On a page with a mix of links and HTML forms, the links need to correspond to top-level entry points of the Page Flow, or to message events that trigger the Page Flow.*

6.3.6. Validating a Page Flow

After creating a Page Flow, you can check to see if there are any problems, such as pages that are not connected or an incorrect number of output links from a page.

To check the Page Flow for errors, click **Validate Page Flow Model** on the Page Flow Designer toolbar. If an error is encountered, a message box displays information about the error. If there are no errors, a message appears stating that there were no errors.

6.3.7. Saving a Page Flow

Even if a Page Flow is unfinished and/or contains errors, you can save it as a work in progress and return to it later. To save an unfinished Page Flow, do one of the following:

- Select the **File** menu and click **Save**.
- Press **Ctrl+S** on the your computer keyboard
- On the Page Flow Designer toolbar, click **Save**.

6.4 Configuring Page Flow Designer Elements

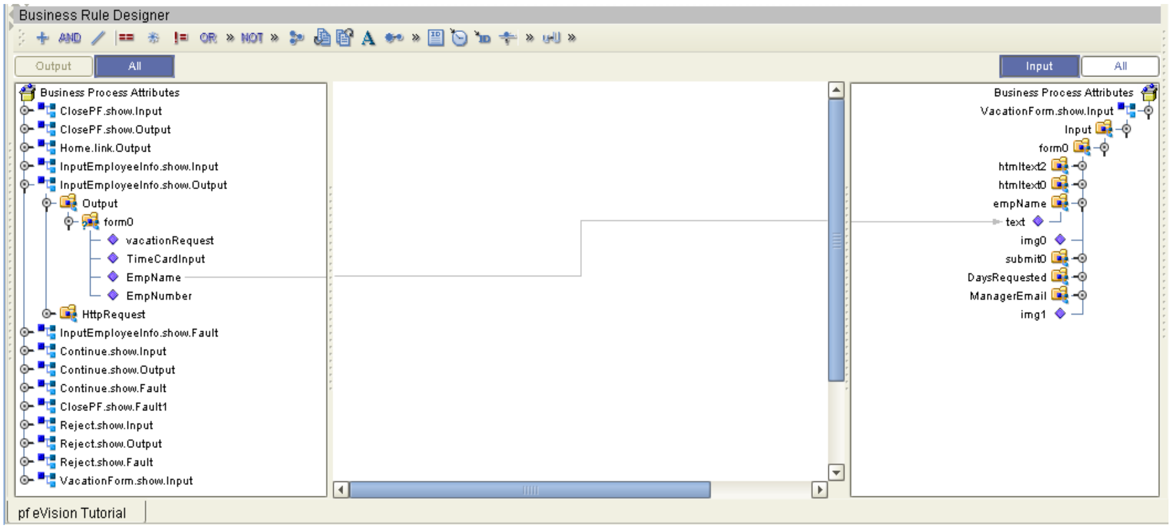
Some elements in the Page Flow Designer have configurable options. This section describes the elements and explains how to configure the options.

6.4.1. About Business Rule Designer

The Business Rule Designer allows 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*. See Figure 67.

The Business Rule Designer appears when you click the **Display Business Rule Designer** icon on the Page Flow Designer toolbar, or when you double click an inline Business Rule or Assign element.

Figure 67 Business Rule Designer



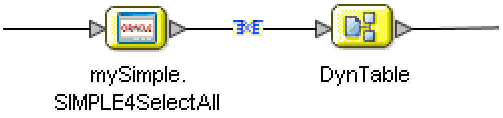
6.4.2. Adding an Inline Business Rule

You can add Business Rules to a link that connects two Page Flow elements.

To Add an Inline Business Rule

1. In the Page Flow, right-click a link between two Page Flow elements.
2. On the context menu, select **Add Business Rule**. The Business Rule icon appears on the link.

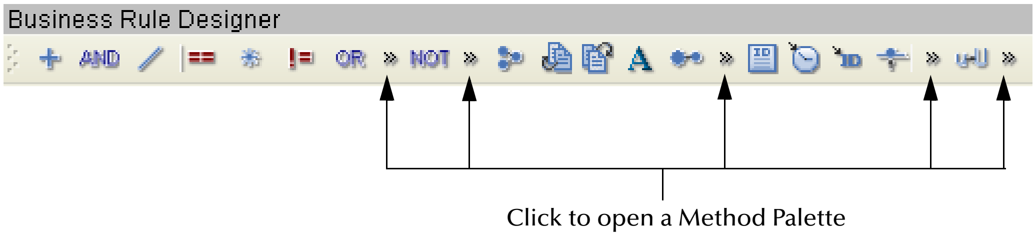
Figure 68 Business Rule Icon



6.4.3. Method Palettes

Use the Method Palettes in the Business Rule Designer to configure data that is passed between input and output Pages. You can drag and drop a method from a method palette to the Business Rule Designer and then configure the method. To open a Method Palette, click the horizontal chevrons on the Business Rule Designer toolbar. See Figure 69.

Figure 69 Business Rule Designer Toolbar



The methods in the Method Palettes are detailed in **"Method Palette" on page 132**.

6.4.4. Sub-Processes

Sub-processes are deployed Page Flows that can be invoked within the same SeeBeyond Integration Server. When a sub-process is dropped into a Page Flow, the Mapper is used to configure the input and output attributes for the sub-process.

Note: *The implementation of sub-processes required eInsight.*

To Use a Sub-process

1. Create the parent process – add an empty Activity as a placeholder for the sub-process.
2. Create the child process – add an empty Receive at the beginning and a Reply at the end.
3. To create a WSDL file that will represent the inputs and outputs of the process, within the child sub-process, do the following:
   - A In the Project Explorer, right click the Page Flow to which you want to add the sub-process.
   - B On the context menu, select **Properties**.
   - C On the **Page Flow Properties** dialog box, select the **WSDL** tab.
   - D On the **WSDL** tab, click **Create**.
   - E Specify values for **portType**, **Operation**, **Input** and **Output**.
   - F Click **OK**.
   - G Open the properties of the initial Receive and select the appropriate partner, portType, and operation.
   - H Do the same for the Reply.

Note: *If the child sub-process is to be invoked as a synchronous request/reply Web service, the Receive and Reply must have the same partner, portType, and operation.*

4. Save the child process.
5. Return to the parent process – from the Project Explorer, drag the first operation under the child process to the sub-process placeholder in the parent process.
6. Perform any appropriate mappings using **Assign Activities** in the parent Page Flow.
7. Deploy both processes.

   For more information on the **Page Flow Properties** dialog box, and how to add WSDL to a Page Flow, see **“WSDL Files” on page 99**.

6.5 Page Flow Properties

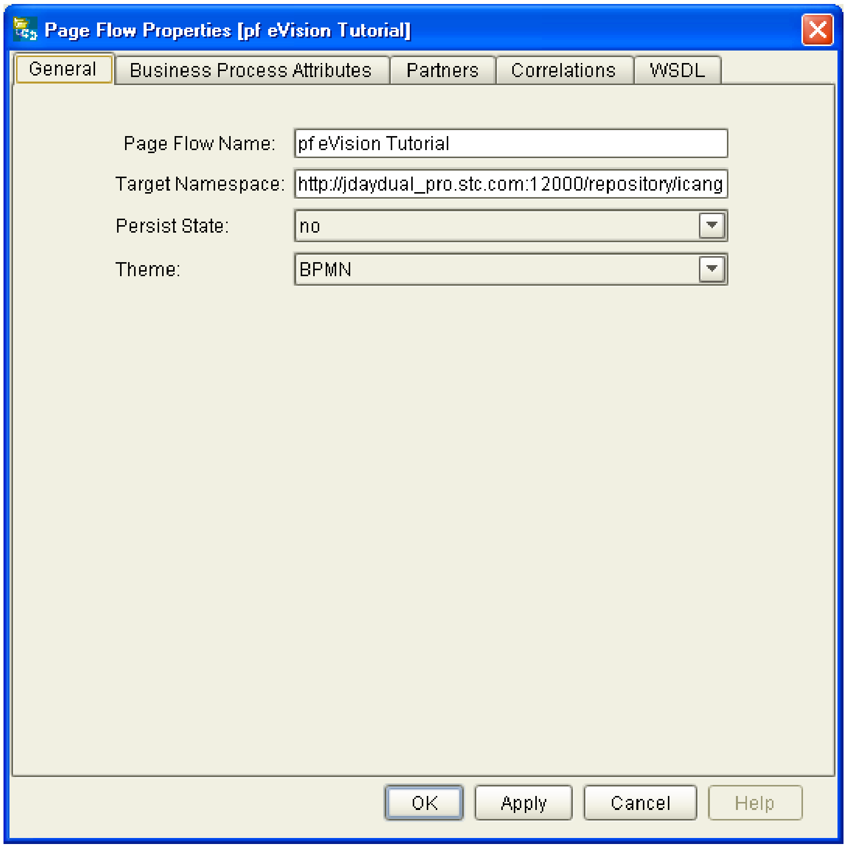
Each Page Flow has a set of properties. These properties enable the rapid creation and removal of Page Flow attributes. eVision uses this information to automatically create the appropriate Page Flow attributes and input/output structures, for use in the Business Rule Designer.

To Edit Page Flow Properties

1. Right-click the Page Flow that you want to edit.
2. On the context menu, select **Properties**.

   The **Page Flow Properties** dialog box appears. See Figure 70.

Figure 70 Page Flow Properties: General Tab



Page Flow Properties [pf eVision Tutorial]

Page Flow Name: pf eVision Tutorial
Target Namespace: http://jdaydual_pro.stc.com:12000/repository/icang
Persist State: no
Theme: BPMN

6.5.1. General

The **General** tab allows you to do the following:

- Change the Page Flow name.
- Edit the target namespace.
- Select the Persistence State.
- Change the theme. The default theme is **BPMN**. Select **Custom1** for a different look.

6.5.2. Page Flow Attributes

Page Flow Attributes are data values used by a Page Flow. They 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 Enterprise 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. Page Flow Attributes can also be assigned 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.

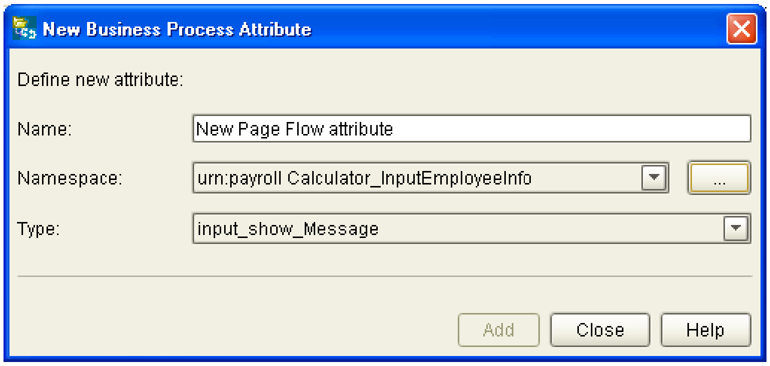
eVision can pass all or part of a complex structure or it can even assemble a composite input to a component or Web Service from multiple Page Flow attributes.

To Create a Page Flow Attribute

1. In the Project Explorer, right-click the Page Flow.
2. On the context menu, select **Properties**.
3. Select the **Page Flow Attributes** tab.
4. Click **Create**.

   The **New Page Flow Attribute** dialog box appears. See Figure 71.

Figure 71 New Page Flow Attribute Dialog Box



Define new attribute:

Name: New Page Flow attribute

Namespace: urn:payroll Calculator_InputEmployeeInfo

Type: input_show_Message

5 Do the following:

- Enter a **Name** for the attribute.
- Select or browse for an existing **Namespace**.
- Select an available **Type** for your attribute.

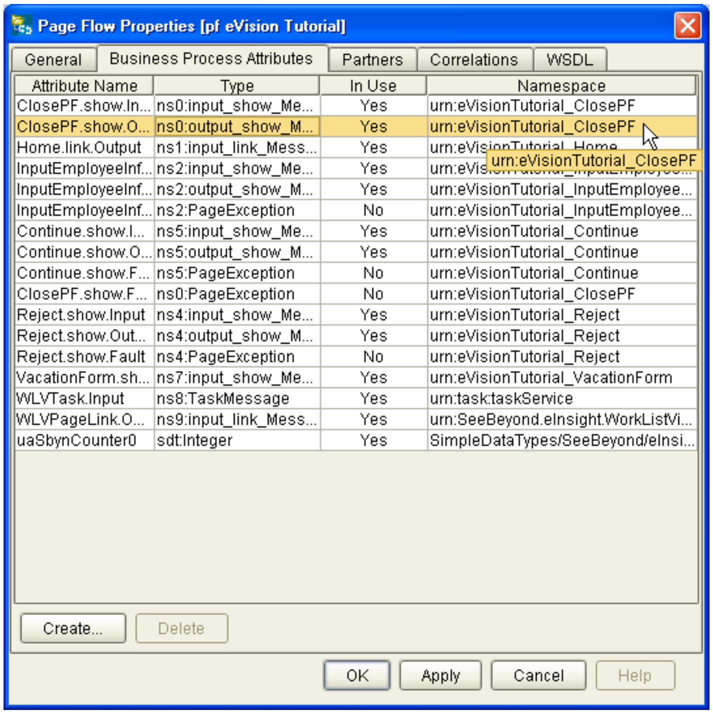
6 To save the attribute, click **Add**.

7 To return to the **Page Flow Properties** dialog box, click **Close**.

To Edit a Page Flow Attribute

1 On the **Page Flow Properties** dialog box, select the **Page Flow Attributes** tab. See Figure 72.

Figure 72 Page Flow Properties: Page Flow Attributes tab


Page Flow Properties [pf eVision Tutorial]
General
Business Process Attributes
Partners
Correlations
WSDL
urn:eVisionTutorial_ClosePF
Create...
Delete
OK
Apply
Cancel
Help

Attribute Name	Type	In Use	Namespace
ClosePF.show.In...	ns0:input_show_Me...	Yes	urn:eVisionTutorial_ClosePF
ClosePF.show.O...	ns0:output_show_M...	Yes	urn:eVisionTutorial_ClosePF
Home.link.Output	ns1:input_link_Mess...	Yes	urn:eVisionTutorial_Home
InputEmployeeInf...	ns2:input_show_Me...	Yes	urn:eVisionTutorial_InputEmployee...
InputEmployeeInf...	ns2:output_show_M...	Yes	urn:eVisionTutorial_InputEmployee...
InputEmployeeInf...	ns2:PageException	No	urn:eVisionTutorial_InputEmployee...
Continue.show.I...	ns5:input_show_Me...	Yes	urn:eVisionTutorial_Continue
Continue.show.O...	ns5:output_show_M...	Yes	urn:eVisionTutorial_Continue
Continue.show.F...	ns5:PageException	No	urn:eVisionTutorial_Continue
ClosePF.show.F...	ns0:PageException	No	urn:eVisionTutorial_ClosePF
Reject.show.Input	ns4:input_show_Me...	Yes	urn:eVisionTutorial_Reject
Reject.show.Out...	ns4:output_show_M...	Yes	urn:eVisionTutorial_Reject
Reject.show.Fault	ns4:PageException	No	urn:eVisionTutorial_Reject
VacationForm.sh...	ns7:input_show_Me...	Yes	urn:eVisionTutorial_VacationForm
WLVTask.Input	ns8:TaskMessage	Yes	urn:task:taskService
WLVPageLink.O...	ns9:input_link_Mess...	Yes	urn:SeeBeyond.eInsight.WorkListVi...
uaSbynCounter0	sdt:Integer	Yes	SimpleDataTypes/SeeBeyond/eInsi...

2. Select an existing attribute and do one of the following:
   - To rename an attribute, double click the attribute text and type a new name. Some attributes cannot be renamed.
   - To remove an attribute, select the attribute text and click **Delete**.
3. To save your changes and exit the **Page Flow Properties** dialog box, click **OK**.

6.5.3. Correlation Keys and Sets

To configure correlation, perform the following steps:

- **"Creating a Correlation Key" on page 95**
- **"Adding Correlation Sets" on page 97**
- **"Binding Correlation Sets to Page Flow Elements" on page 98**

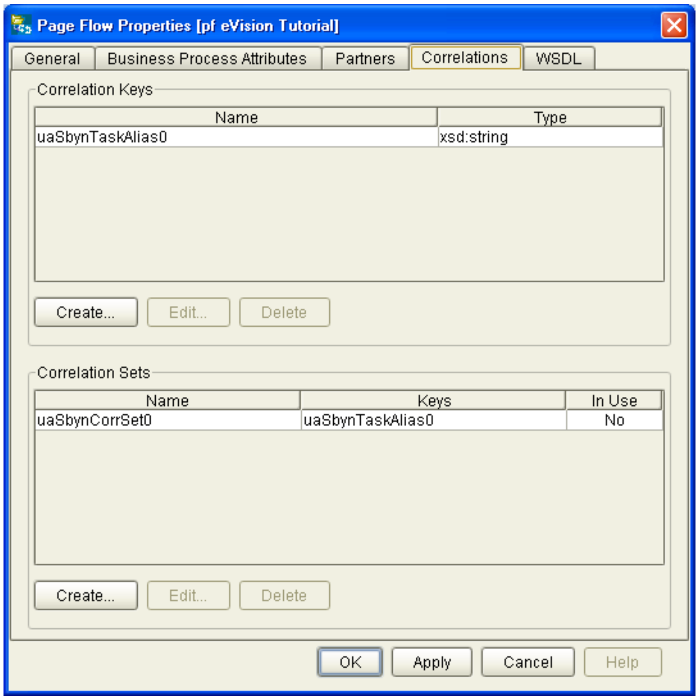
Creating a Correlation Key

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 icon.
2. On the context menu, select **Properties**.
3. Select the **Correlations** tab. See Figure 73.

Figure 73 Page Flow Properties: Correlations Tab



Correlation Keys

Name	Type
uaSbynTaskAlias0	xsd:string

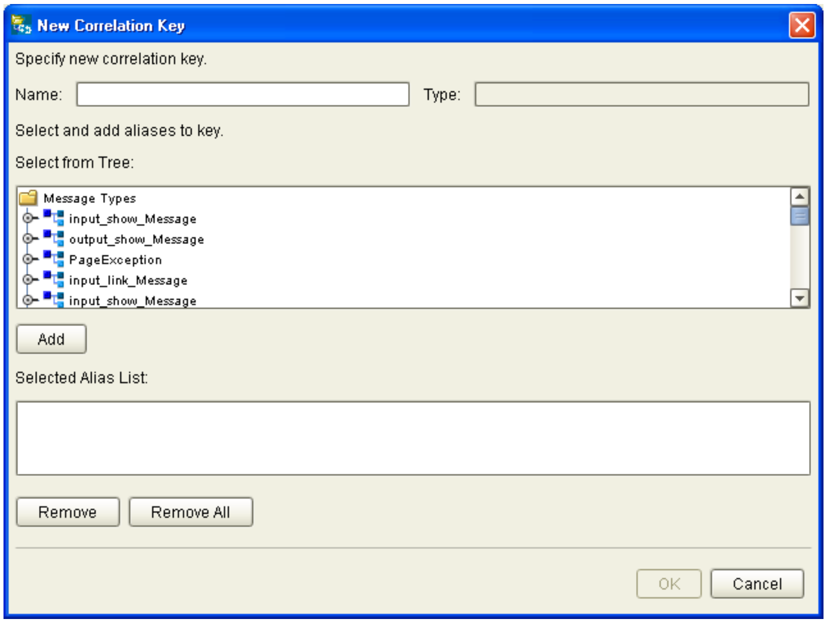
Correlation Sets

Name	Keys	In Use
uaSbynCorrSet0	uaSbynTaskAlias0	No

4 In the **Correlation Keys** section, click **Create**.

The **New Correlation Key** dialog box appears. See Figure 74.

Figure 74 New Correlation Key Dialog Box



Specify new correlation key.

Name: Type:

Select and add aliases to key.

Select from Tree:

- Message Types
  - input_show_Message
  - output_show_Message
  - PageException
  - input_link_Message
  - input_show_Message

Selected Alias List:

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**, click **Add**.

8 To save your changes and exit the **New Correlation Key** dialog box, 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 may 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.

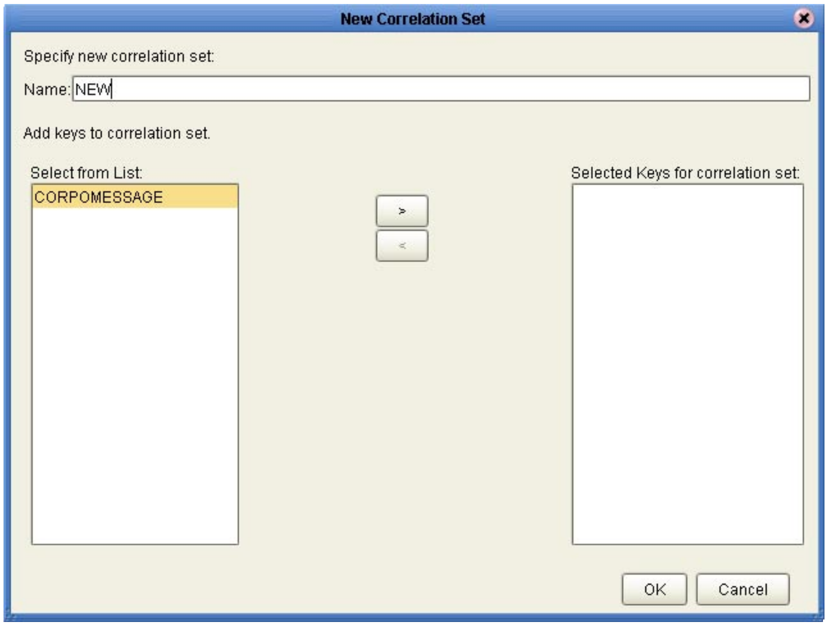
To add a correlation set

1 Select the **Correlations** tab. See Figure 73.

2 In the **Correlation Sets** section, select **Create**.

The **New Correlation Set** dialog box appears. See Figure 75.

Figure 75 New Correlation Set Dialog Box



New Correlation Set

Specify new correlation set:

Name: NEW

Add keys to correlation set.

Select from List:

CORPOMESSAGE

Selected Keys for correlation set:

3 Enter a **Name** for the new 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 To save your changes and exit the **New Correlation Set** dialog box, click **OK**.

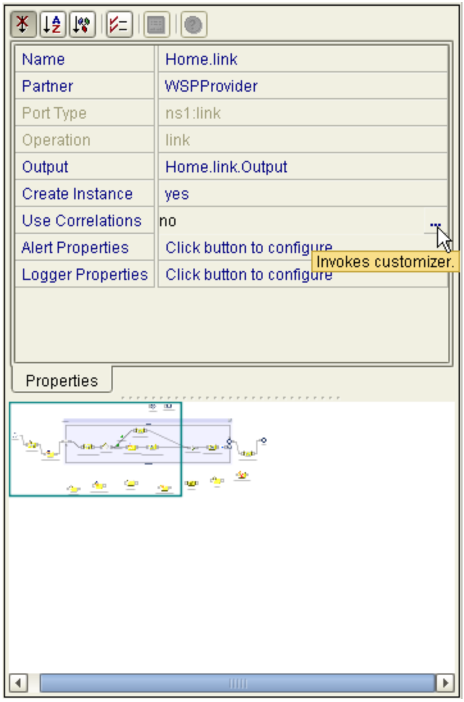
Binding Correlation Sets to Page Flow Elements

When using one or more correlation sets within a Page Flow, the values must be initialized at some point. If the user chooses to initialize the set within an Activity, they will also identify which Page Flow Attribute will be used (or both).

To bind correlation sets to Page Flow elements

1. Select a Page Flow element.
2. On the Page Flow Designer toolbar, click the **Show Property** icon to open the **Element Properties** window.
3. Locate **Use Correlations**, and then click the **no** field.
4. In the **no** field, click the **Command (...)** button. See Figure 76.

Figure 76 Page Flow Designer Element Properties Window



Name	Home.link
Partner	WSPProvider
Port Type	ns1:link
Operation	link
Output	Home.link.Output
Create Instance	yes
Use Correlations	no
Alert Properties	Click button to configure
Logger Properties	Click button to configure

5. On the **Use Correlations** dialog box, click **Add**.
6. On the **Assign Correlation Sets** dialog box, 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)** window.
8. Click **OK**.
9. On the **Use Correlations** dialog box, click **OK**.

6.5.4. WSDL Files

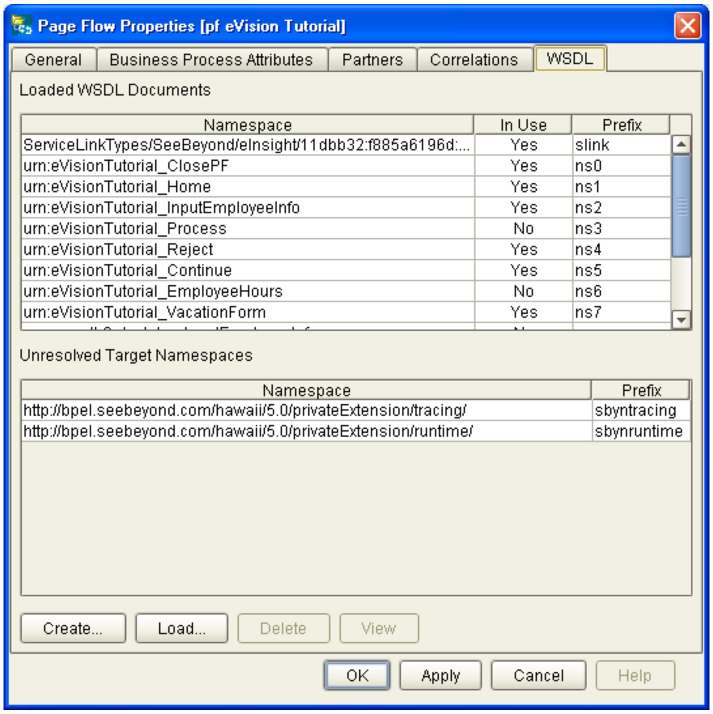
The Web Services Description Language (WSDL) is an XML-based language used to describe business services. WSDL provides a way for individuals and other businesses to electronically access those services. In the Page Flow Designer, WSDL files are used to invoke and operate Web services on the Internet and to access and invoke remote applications and databases.

The WSDL tab is available from the **Page Flow Properties** dialog box. You can upload a WSDL file with predefined Page Flow Attributes for use in your Page Flow.

To Upload a WSDL File

1. In the Project Explorer, right-click the Page Flow icon.
2. On the context menu, select **Properties**.
3. Select the **WSDL** tab. See Figure 77.

Figure 77 Page Flow Properties: WSDL Tab



Page Flow Properties [pf eVision Tutorial]
General
Business Process Attributes
Partners
Correlations
WSDL

Loaded WSDL Documents

Namespace	In Use	Prefix
ServiceLinkTypes/SeeBeyond/eInsight/11dbb32:f885a6196d:...	Yes	slink
urn:eVisionTutorial_ClosePF	Yes	ns0
urn:eVisionTutorial_Home	Yes	ns1
urn:eVisionTutorial_InputEmployeeInfo	Yes	ns2
urn:eVisionTutorial_Process	No	ns3
urn:eVisionTutorial_Reject	Yes	ns4
urn:eVisionTutorial_Continue	Yes	ns5
urn:eVisionTutorial_EmployeeHours	No	ns6
urn:eVisionTutorial_VacationForm	Yes	ns7

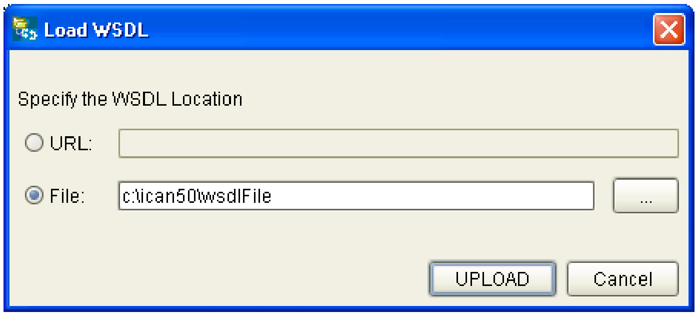
Unresolved Target Namespaces

Namespace	Prefix
http://bpel.seebeyond.com/hawaii/5.0/privateExtension/tracing/	sbyntracing
http://bpel.seebeyond.com/hawaii/5.0/privateExtension/runtime/	sbynruntime

4. To upload a WSDL file, click **Load**.

   The **Load WSDL** dialog box appears. See Figure 78.

Figure 78 Load WSDL Dialog Box



Specify the WSDL Location

○ URL:

● File: c:\ican50\wsdlFile

5 From the **Load WSDL** dialog box:

   A To specify the location of your WSDL, select **URL** or **File**.

   B In the text field, type the path to the WSDL file.

6 Click **UPLOAD**.

Chapter 7

Creating and Deploying a Web Application

This chapter guides you through the process of creating and deploying a Web application.

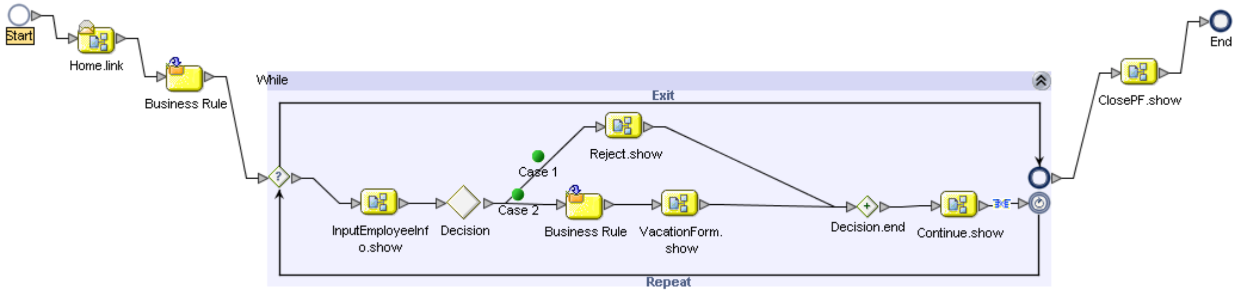
This Chapter Includes:

7.1 Overview

This section provides an overview of importing, assembling, and running the sample Web application that is packaged with eVision Studio. Going through the Page Flow creation process allows you to work with several of the most-used Page Flow Designer tools while assembling the necessary components to complete a Web application that you can deploy, then run on the Logical Host, and access with your browser.

Figure 79 shows the Page Flow Designer view of the sample application, which allows users to request vacation time.

Figure 79 Page Flow Designer View of Sample Application



Creating the sample application consists of the following procedures:

- Downloading the sample project .zip files
- Extracting the contents of the archive files onto your computer
- Importing the sample project components into Enterprise Designer
- Starting a new Page Flow Project
- Adding Page Flow Components to the Page Flow Designer canvas
- Adding pre-configured Web page components to the Page Flow Designer canvas
- Connecting the components
- Configuring the application logic
- Creating a Connectivity Map
- Creating an Environment
- Creating a Deployment Profile
- Activating, then running the application
- Accessing the application Web pages with your browser

7.2 Downloading the Sample Project

Download the sample project only if you have not already done so. If you created the sample Web page as described in **"Laying Out a Web Page" on page 60**, then you already downloaded the files. Skip to **"Importing the Sample Project into the Repository" on page 104**.

The sample project is provided in .zip file format and was installed on the Repository server when you uploaded the eVision Studio documentation **.sar** file. The first step in the process is to acquire the sample project files.

The Repository server must be running before you can use Enterprise Manager or Enterprise Designer. The first step of any procedure is to make sure that the server is running, and if not, to start it. The *SeeBeyond ICAN Suite Installation Guide* describes how to start the Repository server.

To download the sample project

1. Start Internet Explorer.
2. In the **Address** field, enter **http://*hostname*:*portnumber***

   where

   hostname is the TCP/IP host name of the server where the Repository is installed.

   portnumber is the base port number of the Repository.
3. Enter your username and password.

   The **SeeBeyond Customer Login** window of Enterprise Manager appears.
4. Click **Login**.

   The Enterprise Manager home page appears.
5. Click the **DOCUMENTATION** tab.
6. In the left frame of the Documentation page, click **eVision Studio**.

Note: *Documentation is uploaded in a process that is separate from the product uploads. If you do not see an eVision Studio link in the left frame, then the eVision Studio Documentation* ***.sar*** *file must be uploaded. See* **Chapter 3**, **"Installing eVision Studio" on page 21**.

7. In the right frame of the Documentation page, click **Download Sample**.
8. Using an archive utility (such as WinZip), extract the sample files to a folder on your computer. Remember where you saved the files.

   The **eVisionSampleComponents.zip** file contains the pre-built Web pages that you will use to create the sample application.

   The **eVisionTutorial_sample.zip** file is for reference and analysis *after* you successfully build the sample application from the pre-built components.

7.3 Importing the Sample Project into the Repository

You import the **eVisionSampleComponents.zip** Project using Enterprise Designer. Make sure that the Repository server is running before you start Enterprise Designer.

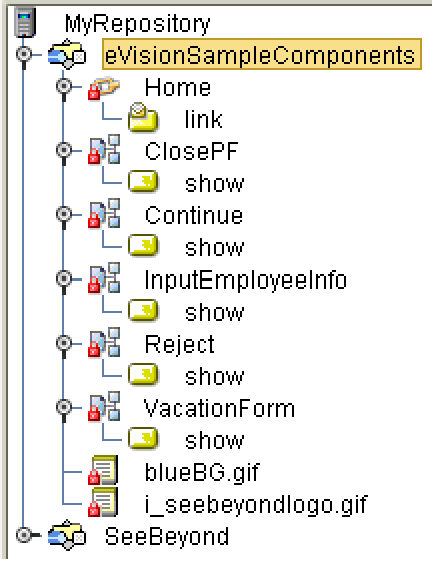
Note: Do not import the ***eVisionTutorial_sample.zip*** *Project until you finish the* ***eVisionSampleComponents.zip*** *tutorial.*

To import the sample project into the Repository

1. Start Enterprise Designer.
2. In the Project Explorer, right click the Repository.
3. On the context menu, select **Import**. The **Import** dialog box appears.
4. Click **Yes**. Import Manager appears.
5. Click **Browse** and navigate to the folder that contains the extracted sample files.
6. Select **eVisionSampleComponents.zip** and click **Open**.
7. Click **Import**.

   When the import process is finished, a confirmation message appears.
8. Click **OK** and then click **Close**.
9. In the Project Explorer, expand the Project to reveal the components. The Project contains a Page Link, five Page Layouts, and two image files. See Figure 80.

Figure 80 Sample Project Components in Project Explorer



MyRepository
eVisionSampleComponents
Home
link
ClosePF
show
Continue
show
InputEmployeeInfo
show
Reject
show
VacationForm
show
blueBG.gif
i_seebeyondlogo.gif
SeeBeyond

7.4 Checking Out the Project Components

Each component has a lock icon, which means that the component is checked into the Version Control system. Before you can use the components, you must check them out.

Note: *Components must be checked out one at a time.*

To check out the Project components

1. In the Project Explorer, under **eVisionSampleComponents**, right-click the **Home** Page Link.
2. On the context menu, click **Check Out**. The **Version Control - Check Out** dialog box appears.
3. Click **Check Out**. The lock icon disappears.
4. Repeat these steps for each component.

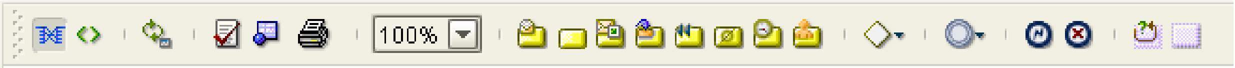
7.5 Creating the Page Flow

This section describes how to assemble the eVision components to create a Page Flow.

Because the Web pages are provided in the sample project, you do not need to create them.

You will use elements in the Page Flow Designer's toolbar to connect the various components in a Page Flow. These elements are described in detail in the previous chapter. See Figure 81.

Figure 81 Page Flow Designer Toolbar



7.5.1 Starting a New Page Flow

To start a new Page Flow

1. In the Project Explorer, right-click **eVisionSampleComponents**.
2. On the context menu, click **New**, and then select **Page Flow**.

   A new Page Flow icon appears in the Project Explorer tree under **eVisionSampleComponents**, and a blank Page Flow appears on the Page Flow Designer canvas (the right pane of Enterprise Designer). A new Page Flow always contains both **Start** and **End** elements.

7.5.2 Adding the Page Flow Elements

In this procedure, you add the Page Flow elements to the Page Flow.

To add the initial Page Flow elements

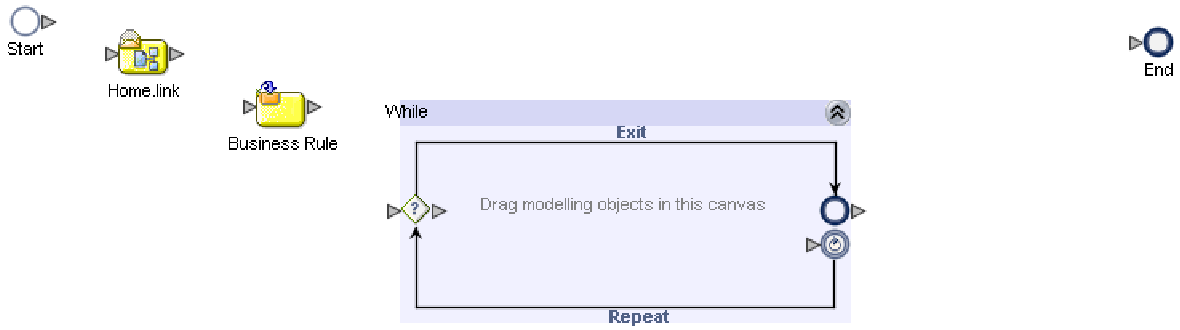
1. In the Project Explorer, expand the **Home** Page Link, if necessary.
2. In the Project Explorer, drag the **Home.link** Web service onto the canvas.

   Place the **Home.link** icon to the right of and below the **Start** element.
3. On the Page Flow Designer toolbar, drag a **Business Rule** onto the canvas.

   Place the **Business Rule** element to the right of and below the **Home.link** icon.
4. On the Page Flow Designer toolbar, drag a **While** element onto the canvas.

   Place the **While** icon in the center of the canvas.
5. Double-click the **While** icon to expand it.
6. Check your progress. The Page Flow should look like the example in Figure 82.

Figure 82 Initial Page Flow Elements



To add the Page Flow elements inside the While loop

1. In the Project Explorer, drag the **InputEmployeeInfo.show** Web service onto the canvas.

   Without releasing the mouse button, place the **InputEmployeeInfo.show** icon to the far left inside the **While** Loop boundaries.

Caution: *Do not place the icon outside of the **While** Loop and then drag the icon inside the **While** Loop.*

2. On the Page Flow Designer toolbar, click the **Branching Activities** icon. On the drop-down menu, click **Decision**, and drag the **Decision** element onto the canvas.

   Without releasing the mouse button, place the **Decision** element to the right of and aligned with the **InputEmployeeInfo.show** icon inside the **While** loop.

   Two icons appear: **Decision** and **Decision.end**.

3 Move the **Decision.end** icon to the right so that you have enough room to place two icons between **Decision** and **Decision.end**. Notice how the **While** loop becomes wider.

4 On the Page Flow Designer toolbar, drag a **Business Rule** onto the canvas.

Without releasing the mouse button, place the **Business Rule** element to the right of and aligned with the **Decision** element inside the **While** loop.

5 In the Project Explorer, drag the **VacationForm.show** Web service onto the canvas.

Without releasing the mouse button, place the **VacationForm.show** icon to the right of and aligned with the **Business Rule** element inside the **While** loop.

6 In the Project Explorer, drag the **Continue.show** Web service onto the canvas.

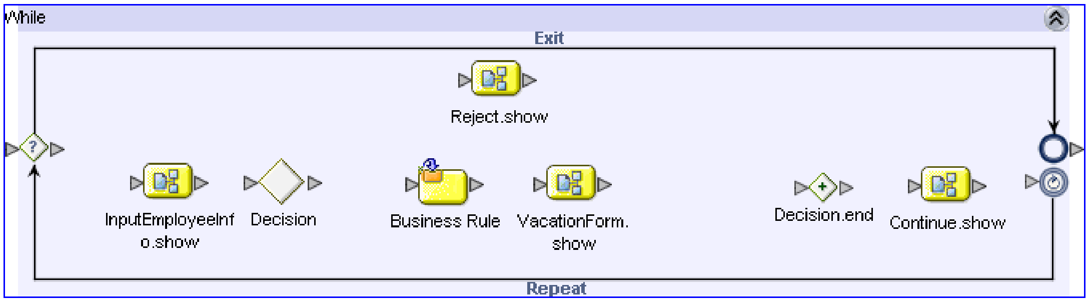
Without releasing the mouse button, place the **Continue.show** icon to the right of and aligned with the **Decision.end** element inside the **While** loop.

7 In the Project Explorer, drag the **Reject.show** Web service onto the canvas.

Without releasing the mouse button, place the **Reject.show** icon above the other elements and center it in **While** loop. Notice how the **While** loop becomes taller.

8 Check your progress. The **While** loop should look like the example in Figure 83.

Figure 83 Page Flow Elements Inside the While Loop



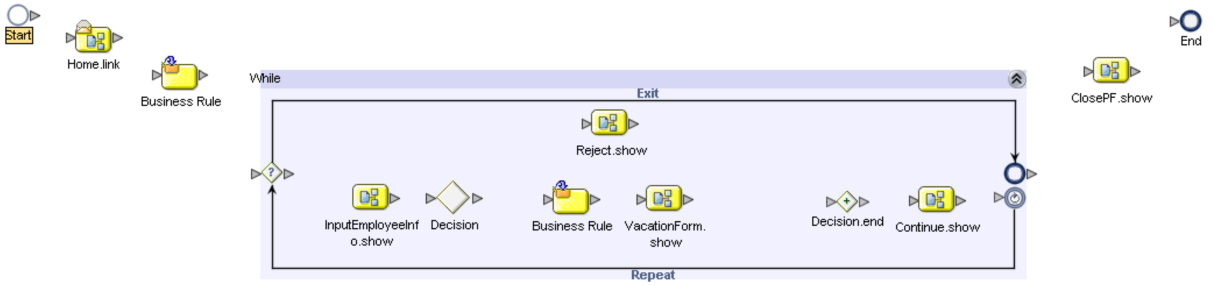
To add the final Page Flow element

1 In the Project Explorer, drag the **ClosePF.show** Web service onto the canvas.

Place the **ClosePF.show** icon between the **While** loop and the **End** node.

2 Check your progress. The Page Flow should look like the example in Figure 84.

Figure 84 Unconnected Page Flow Components



7.5.3 Connecting the Page Flow Elements

In this procedure, you link the Page Flow elements that you added to the Page Flow.

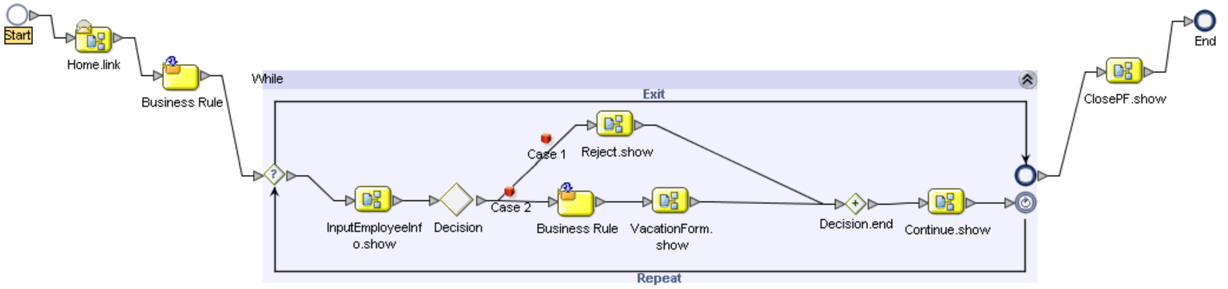
To connect the Page Flow elements

1 Hover over the arrow on the right edge of the **Start** icon until the pointer changes to the hand cursor.

2 Click and drag a connection line onto the **Home.link** element and release the mouse button.

3 Proceeding from left to right, connect all of the Page Flow components. See Figure 85. Note that the **Decision** element has two branching links.

Figure 85 Connected Page Flow Components

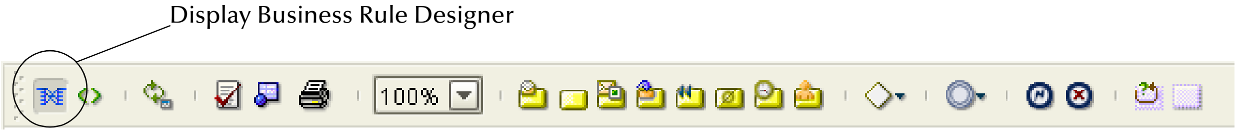


7.5.4 Configuring the Business Rule Elements

In the following procedures, you will use the Business Rule Designer to configure relationships between Input and Output Attributes. You will map outputs to inputs using an area called the *Mapper*. The Business Rule Designer window consists of three panes: **Output** (left), **Input** (right), and the **Mapper** (center).

1 To display the Mapper, click the **Display Business Rule Designer** icon on the Page Flow Designer toolbar.

Figure 86 Starting the Business Rule Designer



The Business Rule Designer appears in the lower pane of the Page Flow Designer.

2 On the Page Flow Designer canvas, select the **Business Rule** that resides outside and to the left of the **While** loop.

Caution: *Do not select the **Business Rule** that resides inside the **While** loop.*

The Business Rule Designer is populated with Output and Input Attributes.

3 In the Business Rule Designer **Input** pane (the right pane), expand **Continue.show.Output** to expose the **radiogroup0** node.

The **radiogroup0** node represents the **Radio Group** component in the **Continue** Page Layout. **radiogroup0** is the value of the component's **lname** property.

4 Expand the **radiogroup0** node to expose the **value** element.

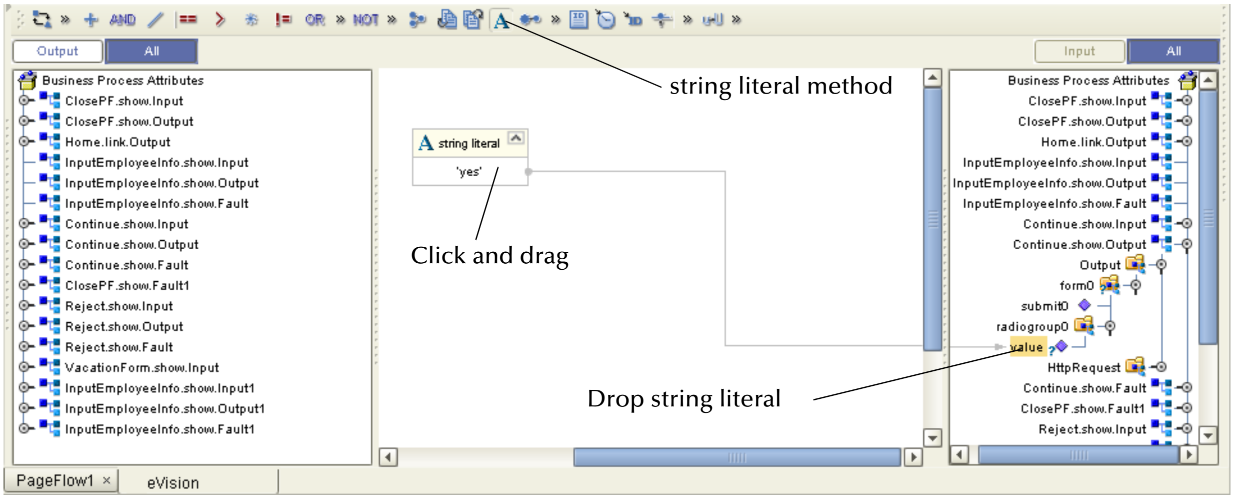
5 On the Business Rule Designer **Method Palette**, drag the **string literal** icon onto the Mapper canvas. The **Input** dialog box appears.

6 In the **Enter a Literal Value** box, type **yes** (make sure the text is lower-case).

7 Click **OK**.

8 Select the **string literal** container within the **"Yes"** box, drag it into the **Input** pane, and drop it onto the **value** element. See Figure 87.

Figure 87 Mapping a String Literal



string literal method

Click and drag

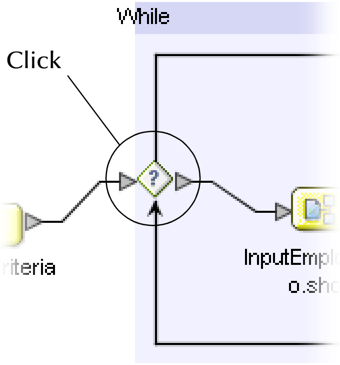
Drop string literal

7.5.5 Configuring the While Loop

To configure the While loop

1 At the left boundary of the **While** loop, click the diamond with a question mark inside it. See Figure 88.

Figure 88 While Loop Icon



Click

2 In the Business Rule Designer **Output** pane (the left pane), expand the **Continue.show.Output** node to expose the **radiogroup0** node.

3 Expand the **radiogroup0** node to expose the **value** element.

4 On the Business Rule Designer **Method Palette**, drag the **string literal** icon onto the Mapper canvas. The **Input** dialog box appears.

5 In the **Enter a Literal Value** box, type **yes** (make sure the text is lower-case).

6 Click **OK**.

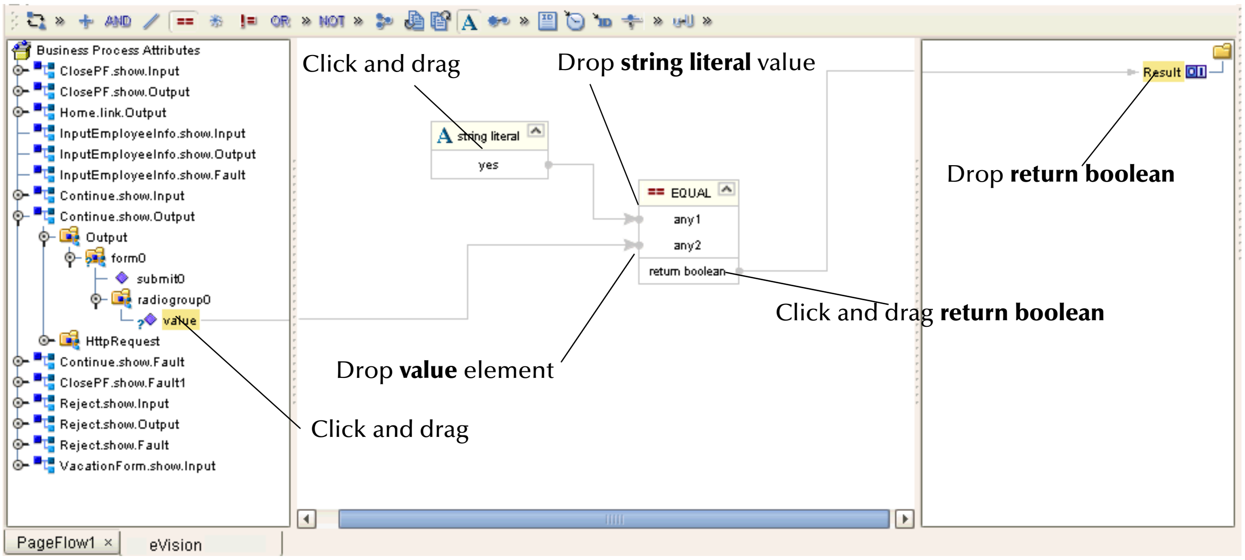
7 On the Business Rule Designer **Method Palette**, drag the **EQUAL** icon (a "double-equal" sign) onto the Mapper canvas.

8 Click the **string literal** container within the **"Yes"** box and drag the cursor onto the **any1** connector node on the **EQUAL** container (the upper node on the left of the container).

9 In the **Output** pane, click the **radiogroup0 value** node and drag it onto the **any2** connector node on the **EQUAL** container (the lower node on the left of the container).

10 Click the **EQUAL** container on the **return boolean** operator window, drag the **return boolean** element into the **Input** pane, and drop it onto the **Result** icon. See Figure 89.

Figure 89 Mapping Within the While Loop



Click and drag

Drop **string literal** value

Drop **return boolean**

Click and drag **return boolean**

Drop **value** element

Click and drag

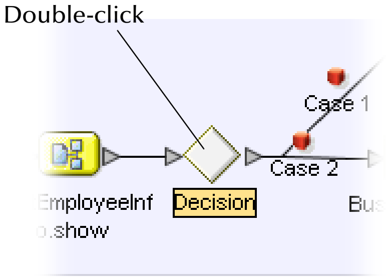
7.5.6 Configuring the Decision Logic

A **Decision Gate** element must have its properties defined such that, based on incoming events, there can be two or more path possibilities within the Page Flow. This section describes how to define properties for a **Decision Gate** element that contains two cases.

To configure the decision logic

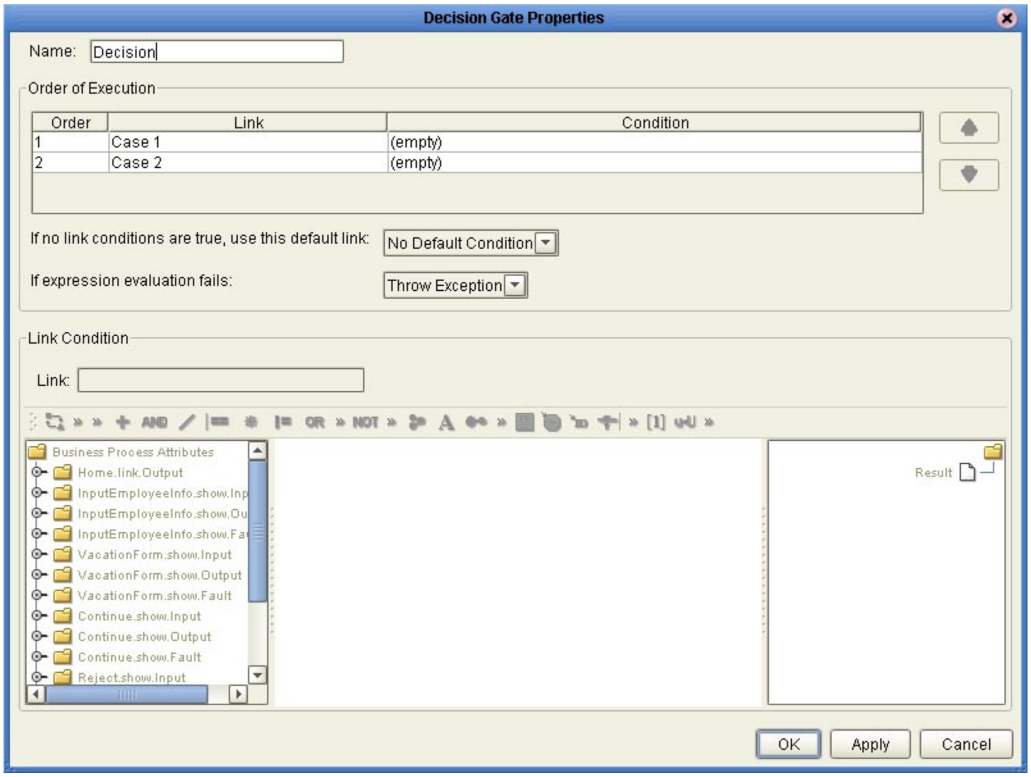
1 On the Page Flow Designer canvas, double-click the **Decision** element inside the **While** loop. See Figure 90.

Figure 90 Decision Element in While Loop



The **Decision Gate Properties** window appears. See Figure 91.

Figure 91 Decision Gate Properties Window



Name: Decision

Order	Link	Condition
1	Case 1	(empty)
2	Case 2	(empty)

If no link conditions are true, use this default link: No Default Condition

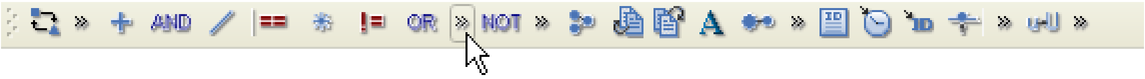
If expression evaluation fails: Throw Exception

Link:

2 Under the **Order of Execution** heading, click **Case1** to activate the Mapper.

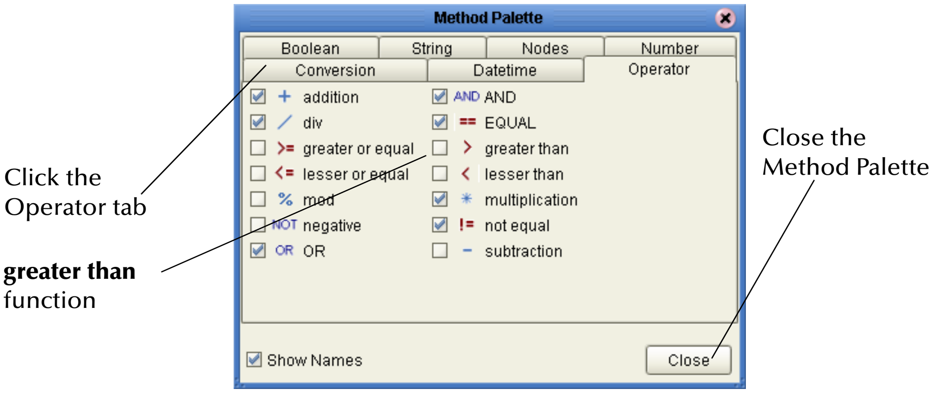
3 To launch the **Method Palette**, click the chevrons just to the right of the **OR** operator. See Figure 92.

Figure 92 Launching the Method Palette



4 On the **Method Palette**, click the **Operator** tab. See Figure 93.

Figure 93 Method Palette with Operator Tab Selected



5 Click the **greater than** check box to add it to the toolbar on the **Decision Gate Properties** window.

6 Click **Close** to exit the palette and return to the **Decision Gate Properties** window.

Defining the Properties for Case 1

Case 1 tests for a valid employee ID number. If a number equal to or greater than 5000 is received, the user will receive an error message and be prompted to try again.

To define the properties for Case 1

1 On the **Decision Gate Properties** toolbar, drag the **string literal** icon onto the Mapper.

2 In the **Input** dialog box, type **5000** to set a high maximum for the employee numbers.

3 Click **OK**.

4 On the **Decision Gate Properties** toolbar, drag the **greater than** icon onto the Mapper.

5 In the **Output** pane of the **Decision Gate Properties** window, expand the **InputEmployeeInfo.show.Output** node until the **EmpNumber** node is exposed.

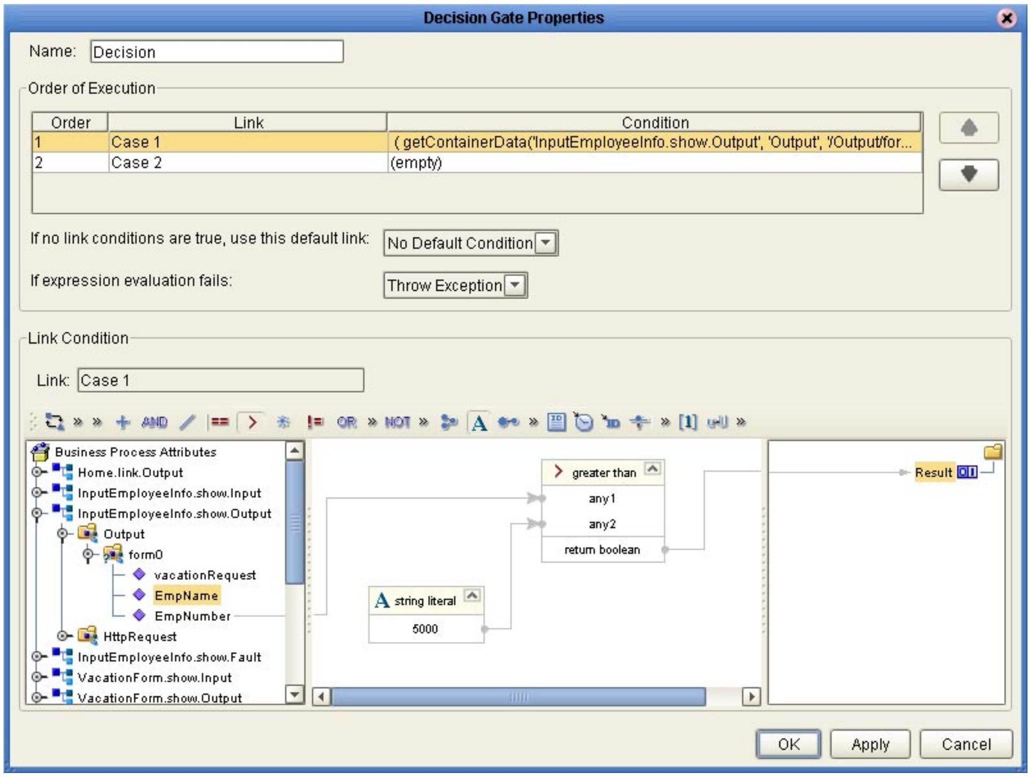
The **EmpNumber** node represents the text box to the right of the **Enter Employee Number** label in the **InputEmployeeInfo** Page Layout. **EmpNumber** is the value of the text box component's **lname** property.

6 Drag the **EmpNumber** node onto the Mapper and, on the **greater than** container, drop it onto the **any1** node (the upper node on the left of the container).

7 Click the **string literal** container near the center of the value box (the value is 5000) and, on the **greater than** container, drop it onto the **any2** node (the lower node on the left of the container).

8 Click the **greater than** container near the center of the **return boolean** box, drag it into the **Input** pane, and drop it onto the **Result** icon. See Figure 94.

Figure 94 Defining Decision Gate Properties: Case 1



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	Case 1	(getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/for...
2	Case 2	(empty)

If no link conditions are true, use this default link: No Default Condition

If expression evaluation fails: Throw Exception

Link Condition

Link: Case 1

Defining the Properties for Case 2

To define the properties for Case 2

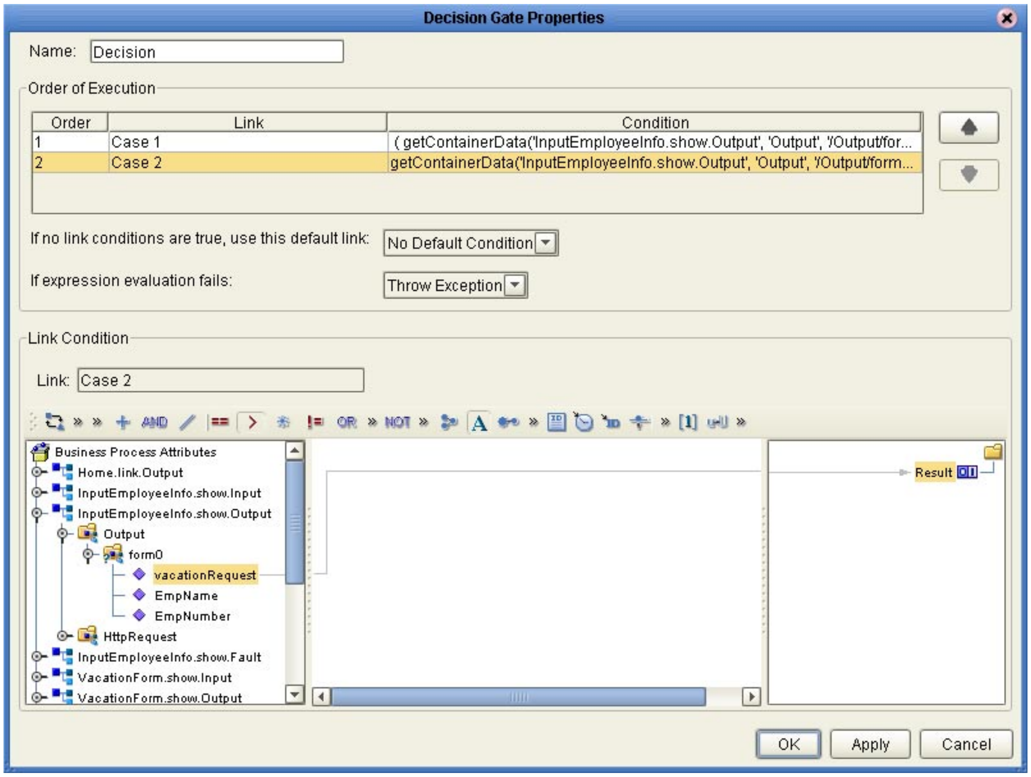
1 Under the **Order of Execution** heading, click **Case2**.

2 In the **Output** pane, expand the **InputEmployee.show.Output** node until the **vacationRequest** node is exposed.

The **vacationRequest** node represents the **Request Vacation** button in the **InputEmployeeInfo** Page Layout. **vacationRequest** is the value of the component's **lname** property.

3 Click the **vacationRequest** node, drag it into the **Input** pane, and drop it onto the **Result** icon. See Figure 95.

Figure 95 Defining Decision Gate Properties: Case 2



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	Case 1	(getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/for...
2	Case 2	getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/form...

If no link conditions are true, use this default link: No Default Condition

If expression evaluation fails: Throw Exception

Link Condition

Link: Case 2

4 Click **Apply**.

5 To close the **Decision Gate Properties** window, click **OK**.

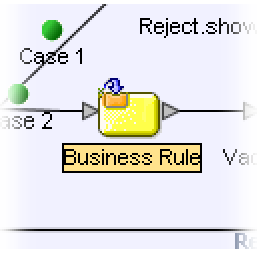
7.5.7 Mapping the Employee Name Output

In this step, the **InputEmployeeInfo.show.Output** attribute is mapped to receive only a text string: the employee's name.

To map the Employee Name output

1 On the Page Flow Designer canvas, click the **Business Rule** element inside the **While** loop. See Figure 96.

Figure 96 Business Rule Element in the While Loop



2 In the Mapper **Output** pane, expand the **InputEmployeeInfo.show.Output** node to to expose the **EmpName** node.

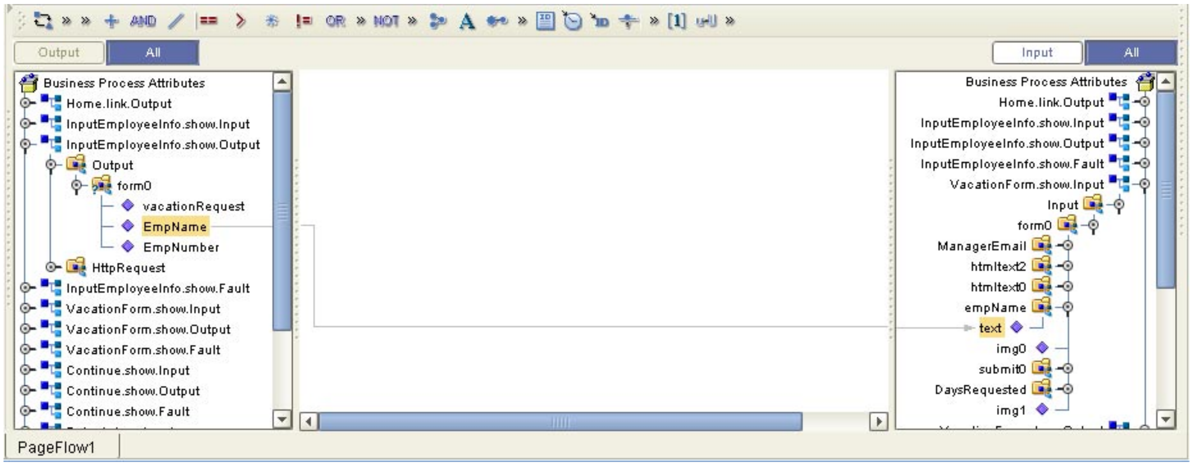
The **EmpName** node represents the text box to the right of the **Enter Name** label in the **InputEmployeeInfo** Page Layout. **EmpName** is the value of the text box component's **Iname** property.

3 In the Mapper **Input** pane, expand the **VacationForm.show.Input** node to expose the **empName text** node.

The **empName** node represents the upper **HTML Text** component in the **VacationForm** Page Layout. **empName** is the value of the component's **Iname** property.

4 Click the **EmpName** node, drag it onto the **Input** pane, and drop it onto the **text** node. See Figure 97.

Figure 97 Mapping Employee Name Output to Text Input



7.5.8 Mapping the Page Flow Attributes

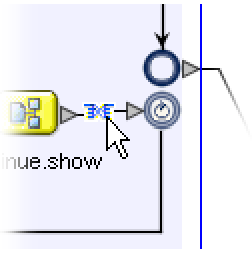
The last step uses the Mapper to map **Outputs** directly to **Inputs**. In the Web application, when the user returns to the **InputEmpInfo** page, the input values are restored.

To map the Page Flow attributes

1 On the Page Flow Designer canvas, right-click the connector between the **Continue.show** page and the **While** loop.

2 On the context menu, click **Add Business Rule**. See Figure 98.

Figure 98 Adding a Business Rule



3 In the Business Rule Designer, in the **Output** pane, click **All**.

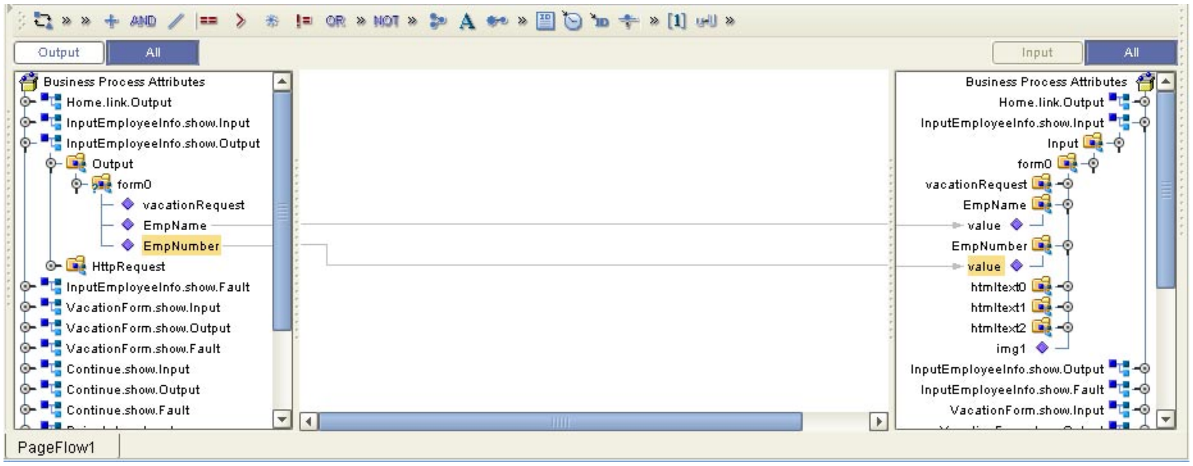
4 In the **Output** pane, expand the **InputEmployeeInfo.show.Output** node to expose the **EmpName** and **EmpNumber** nodes.

5 In the **Input** pane, expand the **InputEmployeeInfo.show.Input** node to expose the **value** nodes under the **EmpName** and **EmpNumber** nodes.

6 In the **Output** pane, click the **EmpName** node, drag it into the **Input** pane, and drop it onto the **value** node under the **EmpName** node

7 In the **Output** pane, click the **EmpNumber** node, drag it into the **Input** pane, and drop it onto the **value** node under the **EmpNumber** node. See Figure 99.

Figure 99 Output to Input Mapping



The mapping procedures are finished.

8 On the **File** menu, click **Save All**.

The Page Flow is finished.

7.6 Creating the Connectivity Map

The Connectivity Map enables the connections between the system components and the backend systems.

The Connectivity Map contains an eGate **Service** and a **Web Connector**. These components enable the project to run on the Logical Host. The Web Connector is a logical representation of the Web container in which the Web application runs.

To create the Connectivity Map

1. In the Project Explorer, right-click the Project, and on the context menu, click **New** and then select **Connectivity Map**.

   A new Connectivity Map node appears in your Project. The default name is **CMap1**. You will be switched to the **Connectivity Map Editor** canvas.

2. On the **Connectivity Map Editor** toolbar, select the **Service** icon and drag it onto the canvas.

3. In the Project Explorer, select the **Page Flow** icon and drag it onto the **Service** icon on the canvas.

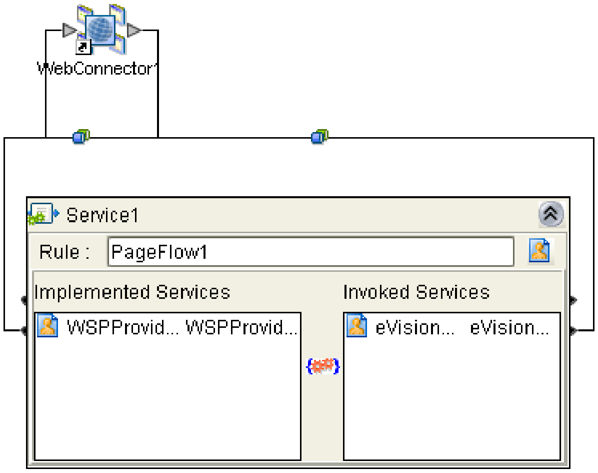
   The **Service** icon changes to reflect containment of the Page Flow.

4. On the **Connectivity Map Editor** toolbar, select the **Web Connector** icon and drag it onto the canvas above the **Service** icon.

5. Double-click the **Service** icon.

6. Connect the arrows to the **Web Connector** icon. See Figure 100.

Figure 100 Connecting the Service and the Web Connector



Rule : PageFlow1

7. Close the **Service** icon. The connectors will appear crossed. This is normal.

   The connection icon between the Web Connector and the Service enables you to configure properties for the presentation component of the application. In this tutorial, you do not need to configure any of the properties.

7.7 Creating the Environment

In this procedure, you create the run-time environment for the eVision application. The environment consists of:

- A Logical Host (the run-time platform)
- An Integration Server (provides run-time services for the application)
- An eVision External System (represents where the presentation component of the application executes)

To create the Environment

1. On the **View** menu, click **Environment Explorer**.
2. Right-click the **Repository** icon and on the context menu, select **New Environment**.

   The **Environment** name defaults to **Environment1**. You can accept the default name.
3. Right-click the new **Environment** and on the context menu, select **New Logical Host**.

   The **Logical Host** name defaults to **LogicalHost1**. You can accept the default name.
4. Right-click the **Logical Host** and on the context menu, select **New SeeBeyond Integration Server**.

   The **Integration Server** name defaults to **IntegrationSvr1**. You can accept the default name.
5. Right-click the **Environment** and on the context menu, select **New eVision External System**.
6. When prompted, enter **eVisionExtSys** and click **OK**.
7. On the **File** menu, click **Save All**.

7.8 Creating and Configuring the Deployment Profile

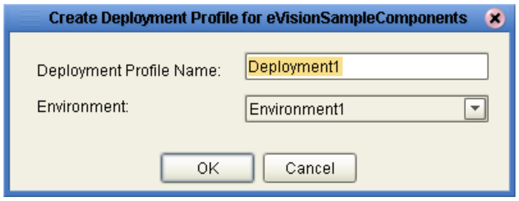
Before you can deploy the application to the Environment, you must create and configure a Deployment Profile.

To create the Deployment Profile

1. On the **View** menu, click **Project Explorer**.
2. In the Project Explorer, right-click **eVisionSampleComponents**.
3. On the context menu, click **New**, and then select **Deployment Profile**.

   The **Create Deployment Profile** dialog box appears. See Figure 101.

Figure 101 Create Deployment Profile Dialog Box


Create Deployment Profile for eVisionSampleComponents

Deployment Profile Name: Deployment1

Environment: Environment1

The Deployment Profile defaults to **Deployment1**. You can accept the default name.

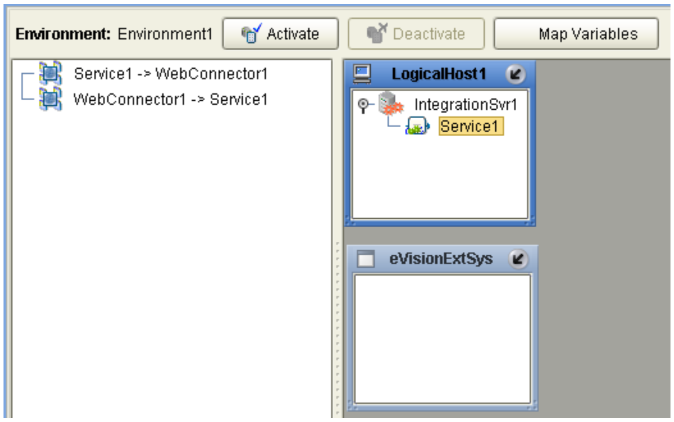
The Environment defaults to **Environment1**. You can accept the default name, or, if there is more than one Environment, click the drop-down arrow and select the Environment name that you just created.

4 Click **OK**.

To configure the Deployment Profile

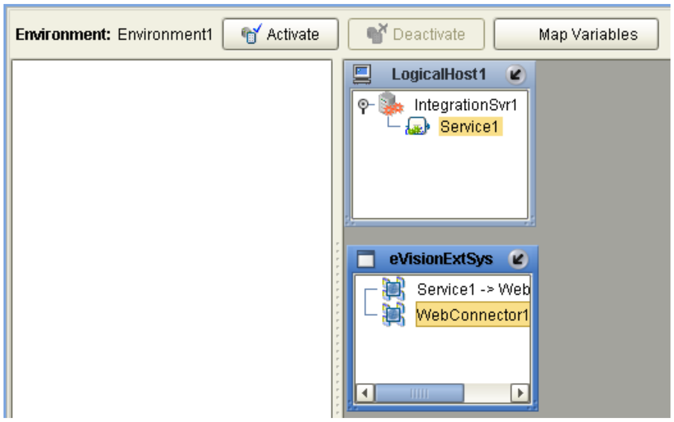
1 In the **Environment** canvas, click the **Service** icon, drag it into the **Logical Host** window, and drop it onto the **Integration Server** icon. See Figure 102.

Figure 102 Service Icon in Logical Host


Environment: Environment1

2 One at a time, drag both **Web Connector** icons into the **eVisionExtSys** window. See Figure 103.

Figure 103 Web Connector Icons in eVision External System

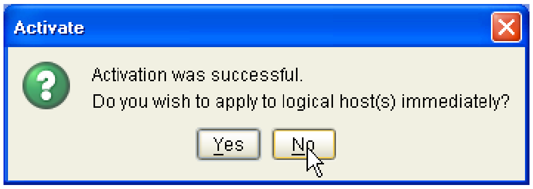


3 On the **Environment** canvas, click **Activate**.

4 When the **Activate** dialog box appears, click **No**.

Click **No** because the **Logical Host** is not yet running. If you click **Yes**, the operation will fail. You will start the Logical Host in another operation. See Figure 104.

Figure 104 Activate Dialog Box

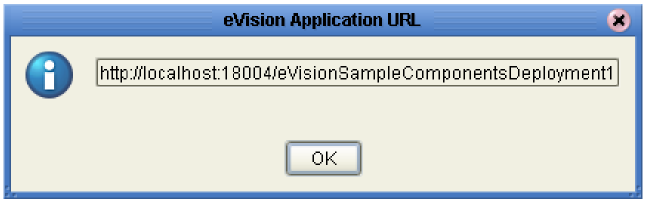

Activate

Activation was successful.
Do you wish to apply to logical host(s) immediately?

Once a project is deployed and you modify it and then re-deploy it, you can click **Yes** (providing the Logical Host is already running).

The **eVision Application URL** dialog box displays the URL for the eVision Web application that you just created. You will need to enter the URL in the address field of your browser to access the Web application. Note that the **Application URL** dialog box allows you to copy and paste the URL. See Figure 105.

Figure 105 eVision Application URL Dialog Box


eVision Application URL

http://localhost:18004/eVisionSampleComponentsDeployment1

The URL has the form:

http://<***hostname***>:<***portnumber***>/<***projectname***><***deploymentprofilename***>

For example:

http://localhost:18004/eVisionSampleComponentsDeployment1

5 Click **OK**.

7.9 Running and Testing the System

The Logical Host bootstrap process executes **eVisionSampleComponents** and starts the process of polling for input data. The bootstrap process will pick up the Deployment Profile the first time it runs; after that, you select **Reactivate** and click **Yes** to apply the most recent changes to the Logical Host.

7.9.1 How to Start the Logical Host

You start the Logical Host by opening a command prompt and typing a command on the command line which includes a string of required, optional, and initial (first-time only) command line arguments appended to the startup command. The startup command string uses the following syntax:

```
<path>\logicalhost\bootstrap\bin\bootstrap <arguments>
```

For example, the command

```
c:\ican50\logicalhost\bootstrap\bin\bootstrap -h
```

displays Help text that explains the command line arguments in detail.

Note: *In the Environment you will be using, the Logical Host must have an Integration Server.*

7.9.2 Logical Host Startup Parameters

The startup command syntax is as follows:

```
bootstrap arguments
```

Table 11 Command Arguments for bootstrap

Parameter	Description	Req/Opt
-d *debug*	Overrides bootstrap sequence. Displays all cached (default) argument values	Optional
-e *environment name*	The name of the Environment to which this Logical Host belongs.	Required (first time only)
-h *help*	Overrides bootstrap sequence. Displays the usage report.	Optional

Table 11 Command Arguments for bootstrap (Continued)

Parameter	Description	Req/Opt
-i *id*	The user ID used for accessing the Repository. Note that the user ID is the same as the username, and that the Administrator can set up more than one user ID.	Required (first time only)
-l *logicalhost name*	The name of this Logical Host.	Required (first time only)
-n *physical host name*	The name of this Physical Host.	Required (first time only)
-p *password*	The password used for accessing the Repository.	Required (first time only)
-r *repository URL*	The root URL for the Repository containing the Logical Host data.	Required (first time only)

Note: *Required (first time only) indicates that the argument is required the first time you start the Repository. You do not need to use it again unless you change the name of the Logical Host.*

A typical bootstrap command would look like the following:

```
bootstrap -e environment_name -l logicalhost_name
-r repository_URL -i username -p password
```

Where:

environment_name is the name of your environment (for example, **Environment1**),

logicalhost_name is the name of your Logical Host (for example, **LogicalHost1**),

repository_URL is the full URL of your Repository including the Repository name (for example, **http://localhost:12000/MyRepository**),

username is your user name, and

password is your password.

7.9.3 Starting the Logical Host

Before you begin, make sure that the Integration Server on the Logical Host has been started. The Logical Host will not start if the Integration Server is not already running.

To start the Logical Host

1. On your desktop, click **Start**, and then select **Run**.
2. In the **Run** dialog box, to open a **Command Prompt**, type **cmd**.
3. Change to the Logical Host **\bin** directory, for example:

c:\ican50\logicalhost\bootstrap\bin

4 To start the Logical Host, on the command line, do one of the following:

- If you are starting the Logical Host for the first time, run the bootstrap command and include all of the required parameters described in **"Command Arguments for bootstrap" on page 122**. For example:

```
bootstrap -e Environment1 -l LogicalHost1
-r http://frodo.acme.com:12000/MyRepository
-i Administrator -p STC
```

- If you are starting the Logical Host after having started it at least once previously, do the following:

  A Using Windows, explore to **c:\ican50\logicalhost\bootstrap\bin**.

  B Double-click **bootstrap.bat**.

Note: *On the Windows operating system, the Bootstrap command is case-sensitive.*

5 Wait until a message appears indicating that the Logical Host is ready.

7.9.4 Accessing the Web Application

With the application running on the Logical Host, you can now access it in your browser.

To access the Web application

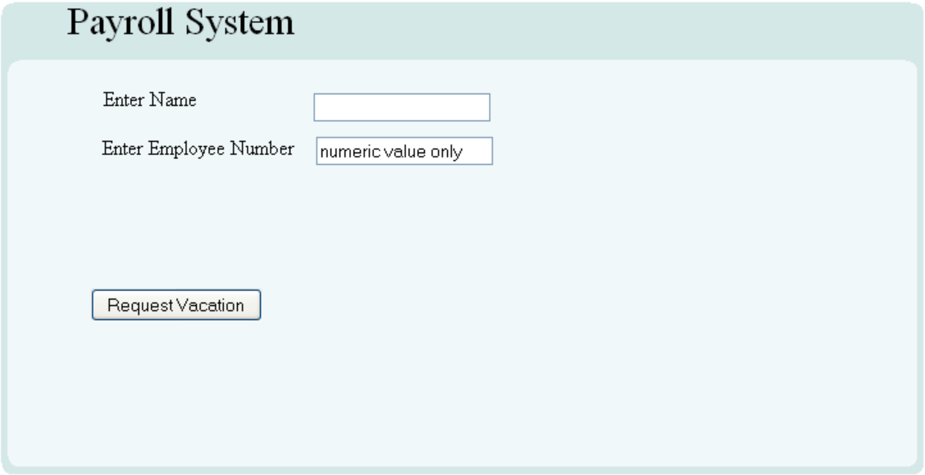
1 Make sure the Repository server is started and running.

2 Start your Web browser.

3 Enter the application URL.

The first page should look like the example shown in Figure 106.

Figure 106 Sample Web Application Startup Page



Payroll System

Enter Name

Enter Employee Number: numeric value only

You can enter an employee number less than 5000 and an employee name, and then proceed with process of requesting vacation time, or you can enter an employee number greater than 5000. A number greater than 5000 will take you to a page with an error message and a prompt to try again. From there you can loop back to the start page.

7.10 Importing the Working Sample Project

When you are satisfied with the “starter” Project, **eVisionSampleComponents**, you can import the sample “reference” Project, **eVisionTutorial_sample**. This Project is more complex than the “starter” project, and it is completely finished and ready for deployment. After importing it, you can analyze its composition before deploying it onto a Logical Host. Once deployed, you can access it in your browser, and work through the finished application from a user’s perspective. You can also add to and modify the Project when you feel you are ready.

Important: *When you import **eVisionTutorial_sample**, you must give it the project name: **eVisionTutorial**. If you use another name, some of the links may break when you open it in the Page Flow Designer.*

Chapter 8

Authentication and Error Handling

This chapter describes how to add authentication to eVision Web applications, as well as how to return preconfigured pages for certain errors.

This Chapter Includes:

8.1 Overview

You have for three authentication options for an eVision Web application:

- No authentication.
- Using the default ICAN authentication.
- Using the preconfigured authentication pages that are provided with eVision. This is the most powerful option.

Table 12 describes the preconfigured authentication pages.

Table 12 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 username and/or password.	yes
Access Denied Error	This page is returned if the user is not allowed to see the requested eVision page. Access Denied Error corresponds to the HTTP response codes 401 and 403.	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 13 describes the preconfigured error-handling pages that are provided with eVision.

Table 13 Preconfigured Error-Handling Pages

Authentication Page	Description
No Such Resource	This page is returned if the requested eVision page cannot be found. No Such Resource corresponds to the HTTP response code 404.
Internal Server Error	This page is returned to the user 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.

8.2 Creating Authentication and Error-Handling Pages

The following procedure describes how to 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.

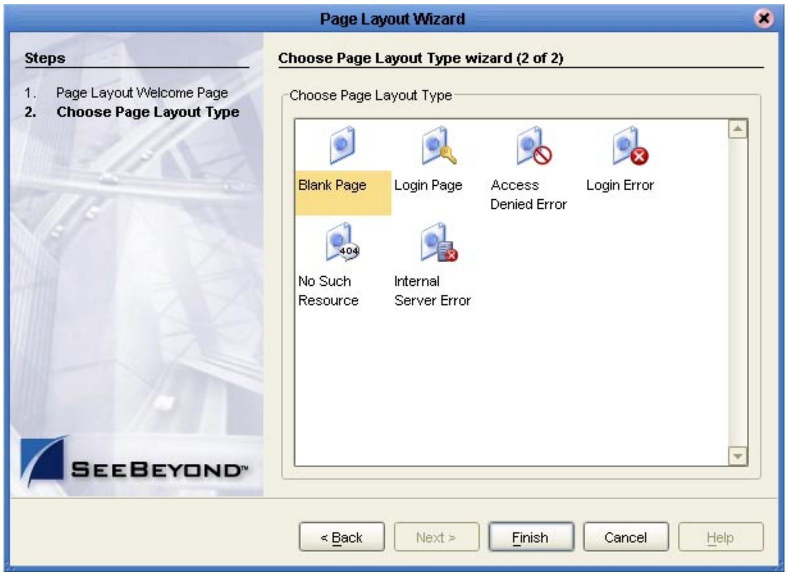
To Create an Authentication or Error-Handling Page

1. In the Project Explorer of Enterprise Designer, right-click the Project.
2. On the context menu, click **New**, and then click **Page Layout**.

   The **Page Layout Wizard** appears.
3. In the **Page Layout Name** box, type a unique name for the new Page Layout (for example, **MyLoginPage**).
4. Click **Next**.

   The **Choose Page Layout Type** window appears.

Figure 107 Page Layout Wizard Page 2



5 Select the authentication or error-handling page that you want to create.

6 Click **Finish**.

The Page Layout Designer appears with the page that you chose.

7 Because the page is preconfigured, you do not need 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** window.

8.3 Configuring the Connectivity Map

If you are using the default ICAN authentication or preconfigured authentication pages, you must set configuration parameters for the Web Connector in the Connectivity Map.

You must also set configuration parameters for the Web Connector if you are using either of the preconfigured error-handling pages.

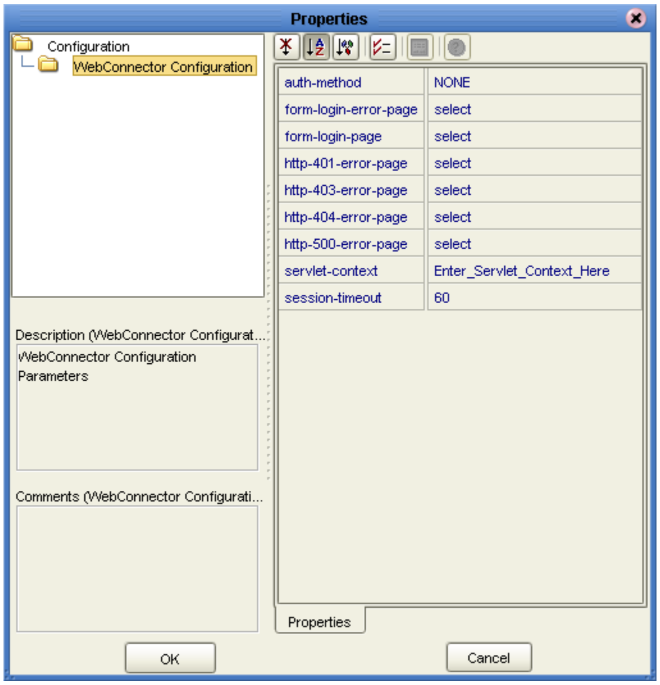
The following procedure assumes that you have created the Connectivity Map for your application. **“Creating the Connectivity Map” on page 118** describes how to do this.

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 Double-click the connection icon between the Web Connector and the Service. The **Properties** dialog box appears. See Figure 108.

Figure 108 WebConnector Configuration Properties


Properties
Configuration
WebConnector Configuration
Description (WebConnector Configurat...
Comments (WebConnector Configurati...
Properties
OK
Cancel

WebConnector Configuration Parameters

auth-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
servlet-context	Enter_Servlet_Context_Here
session-timeout	60

3 If you want to use the default ICAN authentication, set the **auth-method** property to **BASIC**.

4 If you want to use the preconfigured authentication pages, do the following:

A Set the **auth-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, set the **http-401-error-page** and **http-403-error-page** properties to the Access Denied Error Page.

Note: *HTTP response codes 401 and 403 are two variations of the same error.*

5 If you created a No Such Resource Page, set the **http-404-error-page** property to the No Such Resource Page.

6 If you created an Internal Server Error Page, set the **http-500-error-page** property to the Internal Server Error Page.

7 As explained in **"To configure the Deployment Profile" on page 120**, the URL for the eVision Web application has the form:

http://<***hostname***>:<***portnumber***>/<***projectname***><***deploymentprofilename***>

The <***projectname***><***deploymentprofilename***> portion is the *servlet context*. If you want to change the servlet context (for example, to a more user-friendly value), then change the value of the **servlet-context** property.

8 If desired, change the default value of the **session-timeout** property. The value is expressed in minutes.

9 Click **OK**.

8.4 Specifying Users and Roles

The final step of implementing the default ICAN authentication or the preconfigured authentication pages is to specify which users and roles can access the application.

This step involves adding users and roles to the Environment and modifying the eVision External System to indicate which roles are authorized.

The following procedures assume that you have created an Environment for your application. The Environment must include an eVision External System. For more information, see **"Creating the Environment" on page 119**.

To add users and roles to the Environment

The "ICAN Security Features" chapter in the *eGate Integrator System Administration Guide* contains detailed information about how to add users and roles to an Environment. The following procedure is a condensed version of this information.

1 In the Environment Explorer of Enterprise Designer, right-click the Environment and select **User Management**. The **User Management** dialog box appears.

2 Click **Add**.

3 In the **User** field, enter a name for the user.

4 In the **Password** field, enter a password for the user.

5 In the **Confirm Password** field, enter the password again.

6 Click **Add Role**.

7 If you want to create a new role that can be assigned, do the following:

   A Click **Create Role**.

   B In the **Role** field, type the name of the role (for example, **analyst**).

   C Click **OK**.

8 Select the desired role(s) and click **OK**.

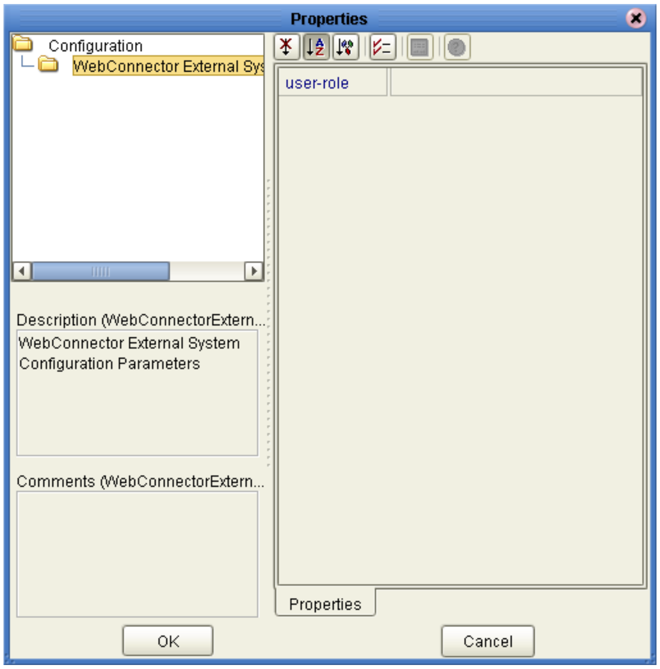
9 Click **OK**.

10 Click **Close**.

To modify the eVision External System

1 In the Environment Explorer of Enterprise Designer, right-click the eVision External System and select **Properties**. The **Properties** dialog box appears. See Figure 109.

Figure 109 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**.

Appendix A

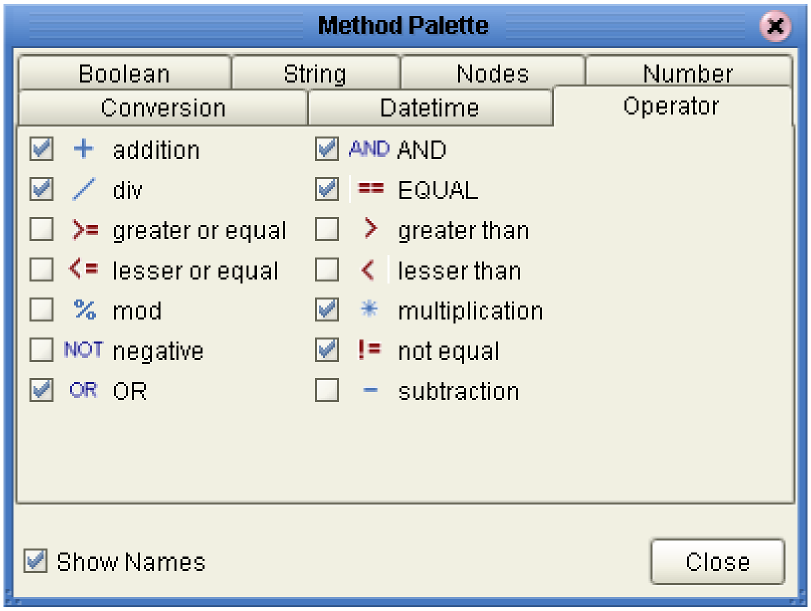
Method Palette

This appendix describes each method that appears in the Method Palette of the Business Rule Designer.

A.1 Operator

Operators are the methods that allow you to manipulate data with standard mathematical operators.

Figure 110 Method Palette: Operator Tab



☑ + addition
☑ / div
□ >= greater or equal
□ <= lesser or equal
□ % mod
□ NOT negative
☑ OR OR
☑ AND AND
☑ == EQUAL
□ > greater than
□ < lesser than
☑ * multiplication
☑ != not equal
□ - subtraction

☑ Show Names

Table 14 Operator Methods

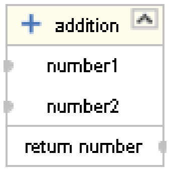
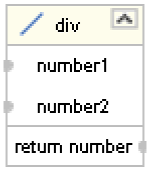
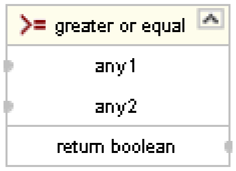
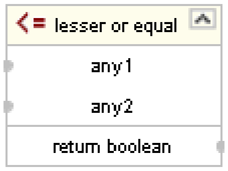
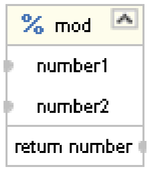
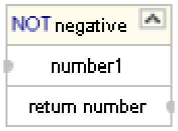
Method Box	Name	Description/Usage
	addition	Adds the value of *number1* to the value of *number2*, returns the sum.
	division	Divides the value of *number1* by the value of *number2*, returns the quotient.
	greater or equal	Returns Boolean true if *number1* is greater than or equal to *number2*; otherwise, returns Boolean false.
	lesser or equal	Returns Boolean true if *number1* is less than or equal to *number2*; otherwise, returns Boolean false.
	mod	Divides number1 by number2 and returns only the remainder.
	negative	Converts the input number to negative. Result is a negative number having the same absolute value as the input number.

Table 14 Operator Methods (Continued)

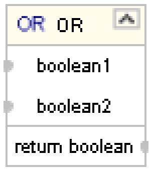
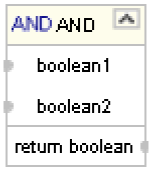
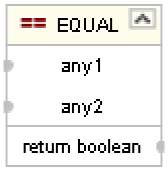
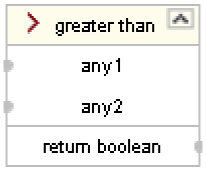
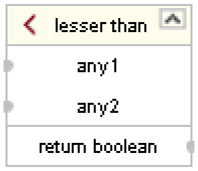
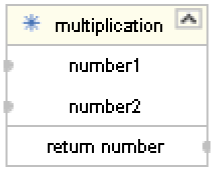
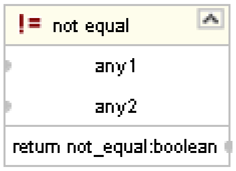
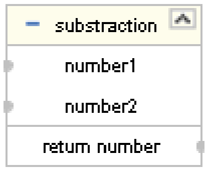
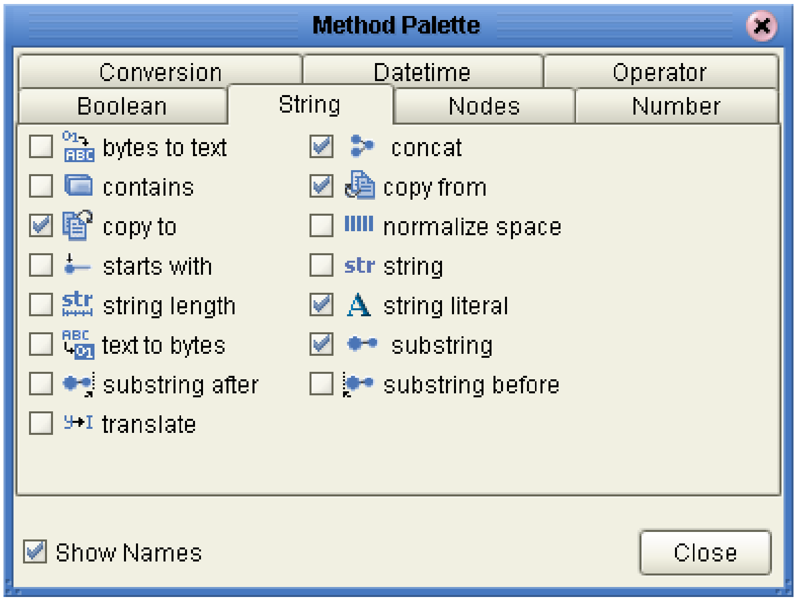
Method Box	Name	Description/Usage
	OR	Returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, returns Boolean true.
	AND	Returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, returns Boolean false.
	EQUAL	Returns Boolean true if *number1* is equal to *number2*; otherwise, returns Boolean false.
	greater than	Returns Boolean true if *number1* is greater than *number2*; otherwise, returns Boolean false.
	less 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.

Table 14 Operator Methods (Continued)

Method Box	Name	Description/Usage
	not equal	Returns Boolean true if *number1* is not equal to *number2*; 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.



- □ 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

Figure 111 Method Palette: String Tab

Table 15 String Methods

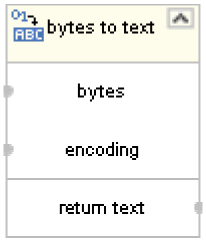
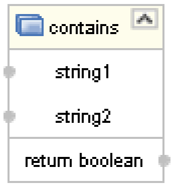
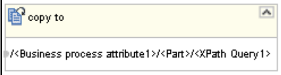
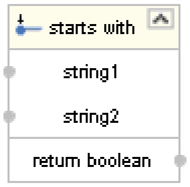
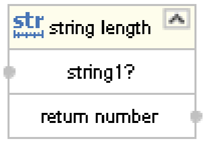
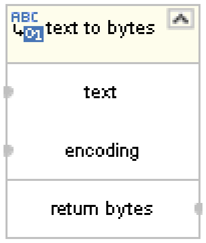
Symbol	Name	Description
	bytes to text	Decodes bytes into text using the specified encoding. If no encoding is specified, the platform's default encoding is used.
	contains	Returns true if the second string is contained within the first string, otherwise it returns false
 /<Business process attribute1>/<Part>/<XPath Query1>	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
	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

Table 15 String Methods (Continued)

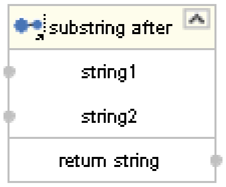
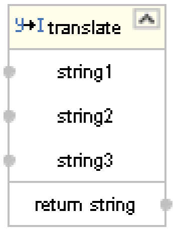
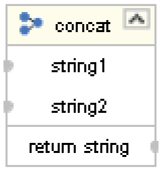
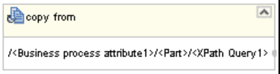
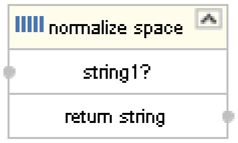
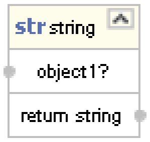
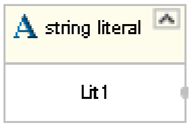
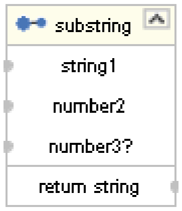
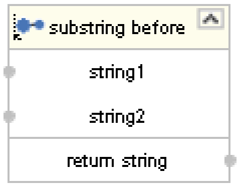
Symbol	Name	Description
	substring after	Returns the part of the string in the string argument that occurs after the substring in the substring argument
	translate	Performs a character by character replacement. It looks in the value argument for characters contained in string1, and replaces each character for the one in the same position in the string2
	concat	Returns the concatenation of all its arguments
	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

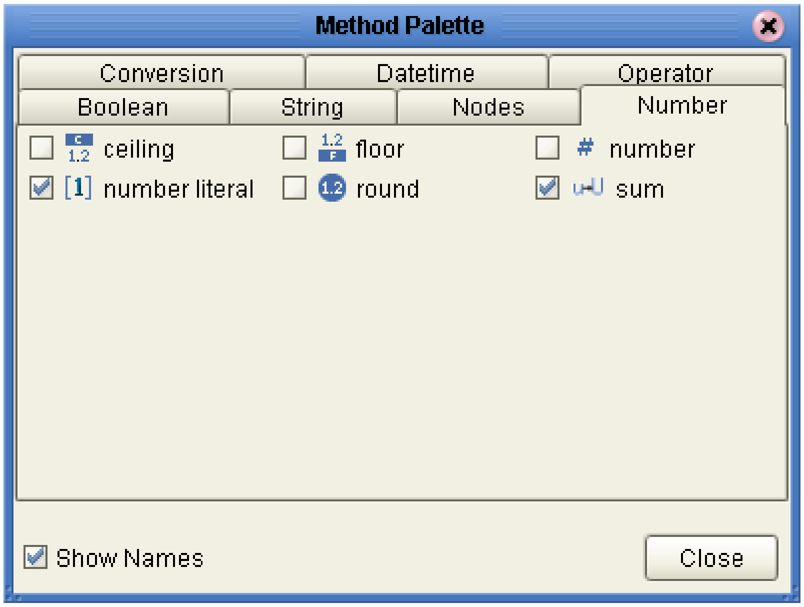
Table 15 String Methods (Continued)

Symbol	Name	Description
	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 112 Method Palette: Number Tab



☐ ceiling ☐ floor ☐ number
☑ [1] number literal ☐ round ☑ sum

☑ Show Names

Table 16 Number Methods

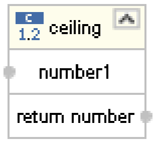
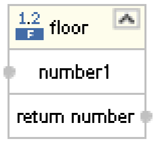
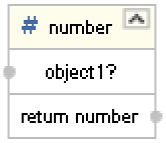
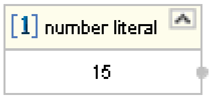
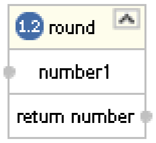
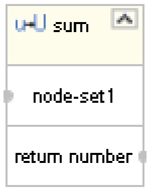
Symbol	Name	Function
	ceiling	Returns the smallest integer that is not less than the number argument
	floor	Returns the largest integer that is not greater than the number argument
	number	Converts the value argument to a number

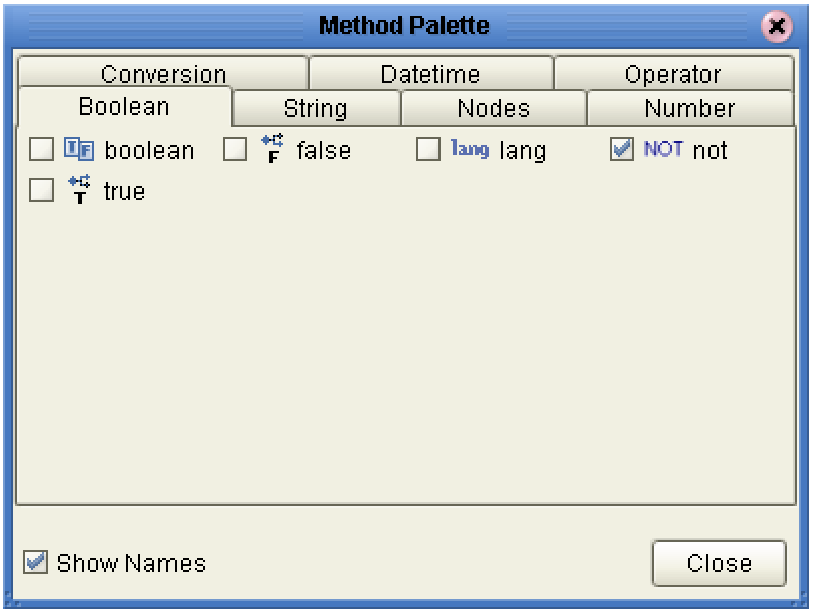
Table 16 Number Methods (Continued)

Symbol	Name	Function
	number literal	A literal number 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

Booleans methods allow you to apply boolean logic to your data.

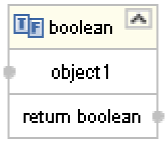
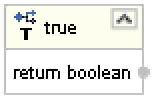
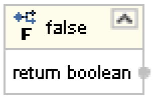
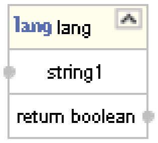
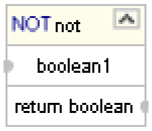
Figure 113 Method Palette: Boolean Tab



□ boolean □ false □ lang lang ☑ NOT not
□ true

☑ Show Names

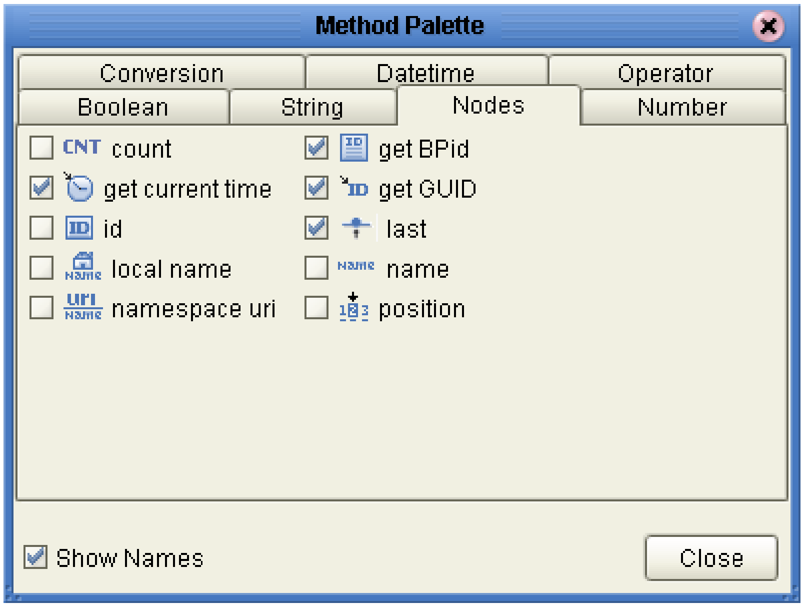
Table 17 Boolean Methods

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 if the condition argument is true

A.5 Nodes

Node methods allow you to manipulate your data.

Figure 114 Method Palette: Nodes Tab



☐ CNT count
☑ get current time
☐ id
☐ local name
☐ namespace uri
☑ get BPid
☑ get GUID
☑ last
☐ name
☐ position

☑ Show Names

Table 18 Nodes Methods

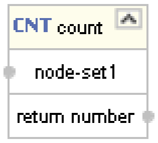
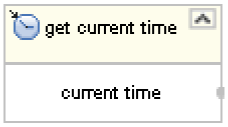
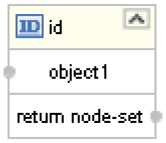
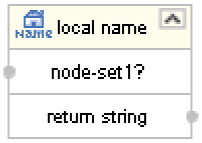
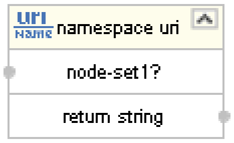
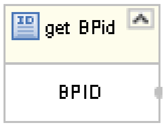
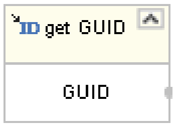
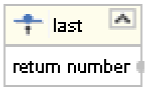
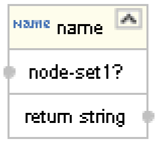
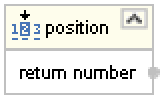
Symbol	Name	Function
	count	Returns the number of nodes in a node-set
	get current time	Gets the current time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
	id	Selects elements by their unique ID

Table 18 Nodes Methods (Continued)

Symbol	Name	Function
	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
	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

Appendix B

Conversational State in eVision Studio Web Applications

This appendix describes the support for conversational state in eVision Studio Web applications.

This Chapter Includes:

B.1 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 115 continues to the left boundary of Figure 116. The letters A and B and the numbers 1 through 13 are used in the explanation that follows.

Figure 115 Page Flow Example - Part 1

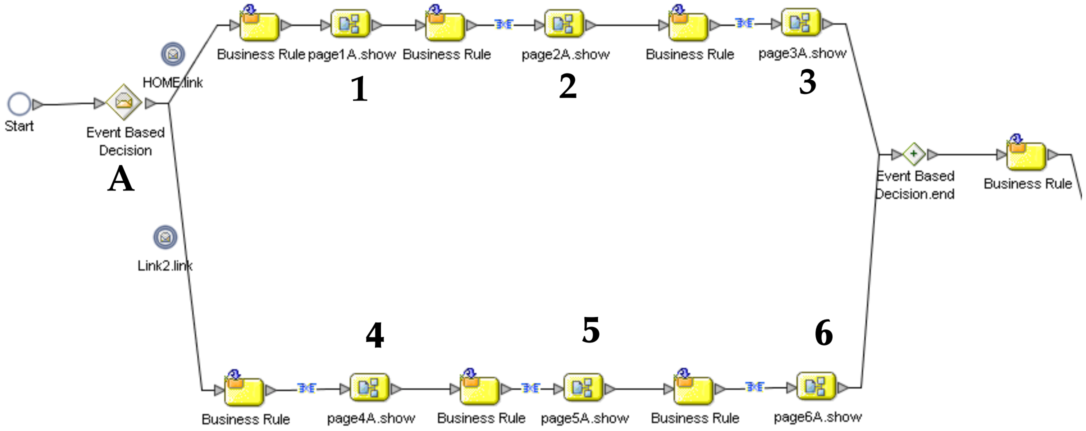
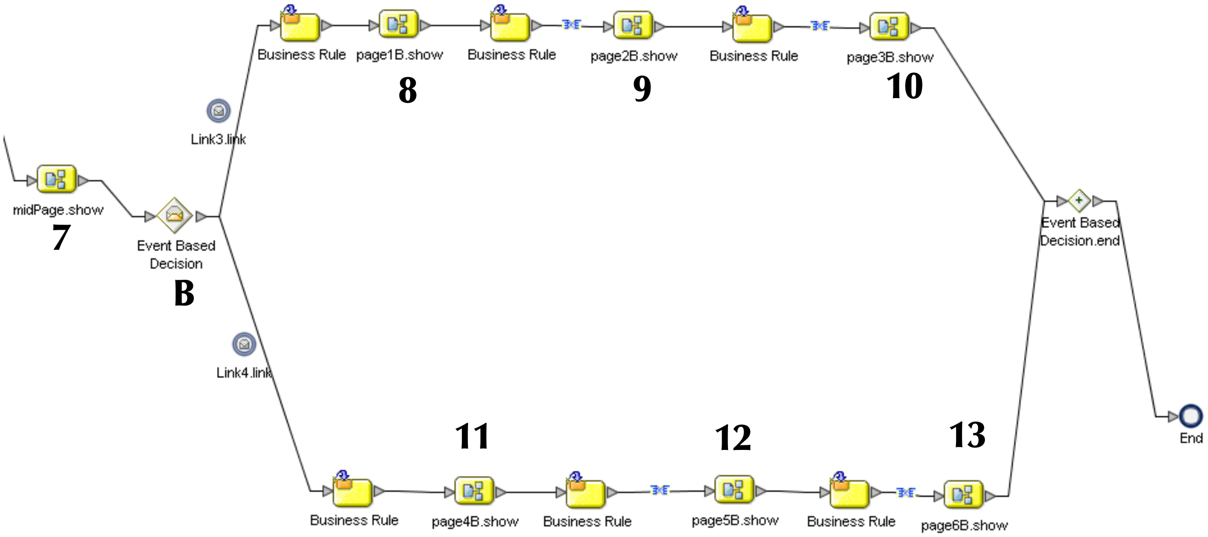


Figure 116 Page Flow Example - Part 2



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 Button 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