SeeBeyond ICAN Suite

eVision Studio User's Guide

Release 5.0

SEEBEYOND®

SeeBeyond Proprietary and Confidential

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20031015201457.

Contents

Chapter 6

Chapter 7

Chapter 8

List of Figures

List of Tables

Chapter 1

Introduction

eVision Studio (eVision) is a graphical design studio for the WYSIWYG 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.

Web applications enable the distribution of integrated Page Flows across the enterprise and allow real-time user interaction with those processes. eVision allows the organization to present a single, unified view of enterprise data and applications to employees, customers, and partners.

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. Web applications can be easily structured to allow employees to see what tasks are assigned to them, then to interact with, and complete the tasks.

1.0.1 Document Purpose and Scope

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

1.1 Intended Audience

This guide is intended for experienced PC users who have the responsibility of helping to set up and maintain a fully functioning ICAN Suite system. This person must also understand any operating systems on which eGate will be installed (Windows or UNIX) and must be thoroughly familiar with Windows-style GUI operations.

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 Web pages
- Instructions for moving and resizing GUI components
- Instructions for saving Web applications in the Repository
- Instructions for binding and deploying Web applications

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 more information, see **"Writing Conventions" on page 12**.

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

When downloading Acrobat Reader, make sure to download the version that includes the option for searching .pdf files—Acrobat Reader with Search. This version is required to view the searchable master index.

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

1.6

Chapter 2

About eVision Studio

This chapter provides an overview of eVision Studio.

This Chapter Includes:

2.1 eVision Web Applications

eVision Studio is a graphical design studio, which 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 text-based 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.

2.1.1 ICAN Integration

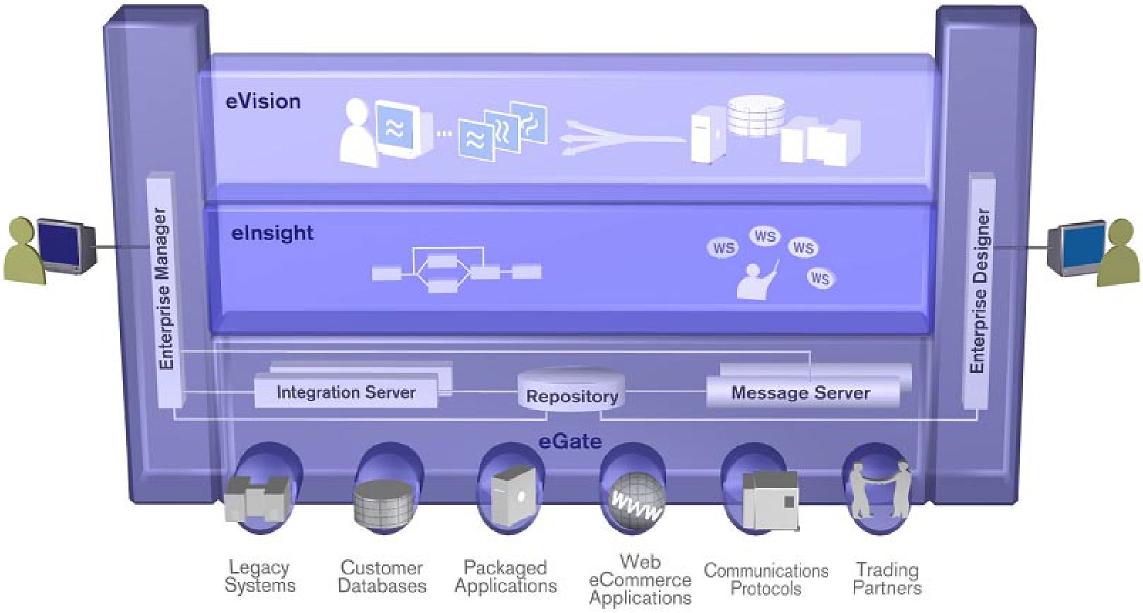
eVision is fully integrated with the ICAN Suite and runs as a component within the ICAN Suite environment. Page Layout Designer and Page Flow Designer run as a components within the Enterprise Designer GUI. eVision Web applications are stored in the SeeBeyond Repository.

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 the information necessary to activate eVision applications and associated components.
- Monitoring - Enterprise Manager lends 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.

An example of how eVision is integrated with the ICAN Suite is shown in **Figure 1 on page 16**.

Figure 1 eVision Studio integration with ICAN Suite components



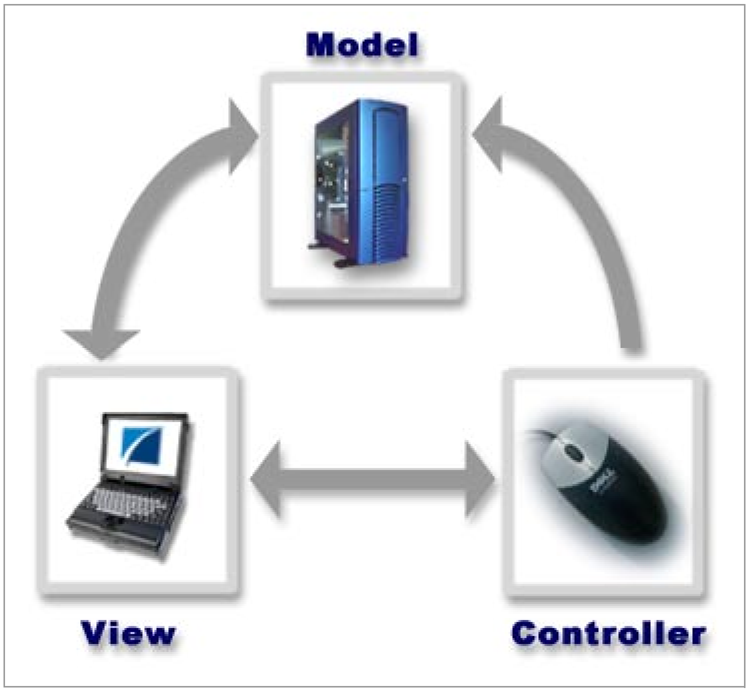
2.2 MVC Architecture

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 on page 17.

- 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.

Figure 2 Model/View/Controller (MVC) Architecture



2.2.1 MVC Application Architecture in eVision

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 **Page Flow Designer**. The Controller orchestrates the sequence of pages being sent to the browser in response to user actions.

2.3 eVision Page Flow Engine

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.

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 would be 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.

2.3.1 Page Flow Engine Run-time Architecture: Continuations

In a typical Web application, when the user sends a request to the backend system, it is handled through HTTP, which is a “stateless” transfer protocol, that is, HTTP processes requests from the browser and returns responses from the server; it does not perform computations, nor does it save data for later use. eVision’s Page Flow Engine leverage a processes called *Continuations*, which generate and store “states” when a user interacts with the application. Continuations make the stateless HTTP protocol behave like it is “stateful” where, in “conversations” with a user, simple or complex computations are held in suspension and stored for later use. This allows the user to return to any page in a Page Flow, and find that the Continuation has automatically captured and restored everything necessary to seamlessly resume the conversation.

Web Applications Without Continuations

Under HTTP, the typical Web application must maintain a conversational state with each client. This usually requires that the Web application be broken down into smaller pieces, such as JSP pages, servlets, etc., each of which is capable of handling a small number of requests. In addition to making an application harder to modify, this “distributed responsibility” solution can contribute to difficulty in maintaining a smooth flow when a user performs actions like cloning the window, submitting concurrent requests, inserting bookmarks, or clicking the Back button

Continuations are leveraged by eVision Studio’s Page Flow engine to add state to Page Flows. Whenever a Page is waiting for user input, it saves the Continuation associated with that user. When the user responds with an action, the Continuation is restored, and the user’s input is returned as the value of the Continuation.

Advantages of Continuations

A Continuation can be invoked any number of times, allowing users to leave a page and return to it as often as necessary. As long as the Continuation persists, the user can leave and then resume working with the application, and always return to the same point in the human/computer dialog. Using Continuations to create conversational state in a Page Flow is particularly effective in promoting smooth page transitions in unusual navigational patterns.

Continuations pass user actions forward by adding a Continuation argument. When a user sends a request to the server, rather than returning a value, the Page Flow continues with the value by presenting it as an argument to the Continuation. In a Page Flow, a function never *returns*; instead, it *continue*s. Based on their behavior, Continuations can be described as *gotos with arguments*.

2.4 eVision Web Design Components

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

2.4.1 Page Layout Designer

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 **Graphical User Interface Components** (Web page design objects) onto a blank page design canvas. Page Layout Designer provides the user with a comprehensive collection of these prebuilt GUI components, and a friendly, WYSIWYG Web page editor. Page Layout Designer components are described in detail in **"Design Tools" on page 28**

2.4.2 Page Flow Designer

Page Flow Designer is the component within the SeeBeyond ICAN Suite that facilitates the flow of Web-based business activities. Page Flow Designer function include Page Flow modeling, monitoring, and execution as well as the ability to analyze how data messages flow from Web page to Web page.

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 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 design and page flow development tools, the business analyst can enable enterprise-wide, internal and external access to Page Flows across an intranet, or the Internet.

Chapter 6

Installing eVision Studio

This chapter describes how to install eVision Studio on a client computer running the Windows 2000 or XP operating system.

6.1 System Requirements

This section lists the system requirements for each platform. The Readme.txt file on the CD-ROM contains the most up-to-date operating system requirements for the supported platforms.

eVision supports 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

  SeeBeyond only supports HPUX running on 9000/8xx machines. 9000/800 is 64-bit, but can also run in 32-bit mode. To determine if the system is 32 or 64 bit, on your workstation, type:

  ```
  getconf KERNEL_BITS
  ```

  This will return the bit mode, 32 or 64.
- IBM AIX 5.1 and 5.2 with required maintenance level patches
- Red Hat Enterprise Linux AS 2.1
- Red Hat Linux 8 (Intel Version)
- Sun Solaris 8 and 9 with required patches

6.1.1. Database Support

The eVision database is used to provide a runtime persistent store for recovery as well as a schema use for reporting purposes. eVision supports the following databases:

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

6.1.2. Installation Steps

The installation procedure for eVision Studio is the same as all other ICAN Suite products. You can find general product installation instructions in the *eGate Integrator Installation Guide,* which is available via the Enterprise Manager application's Documentation Web page.

Before You Begin

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

To install eVision Studio, follow these steps:

1. Open your browser.
2. On the **Address** line, start Enterprise Manager by typing the HTTP address for the Repository Server.

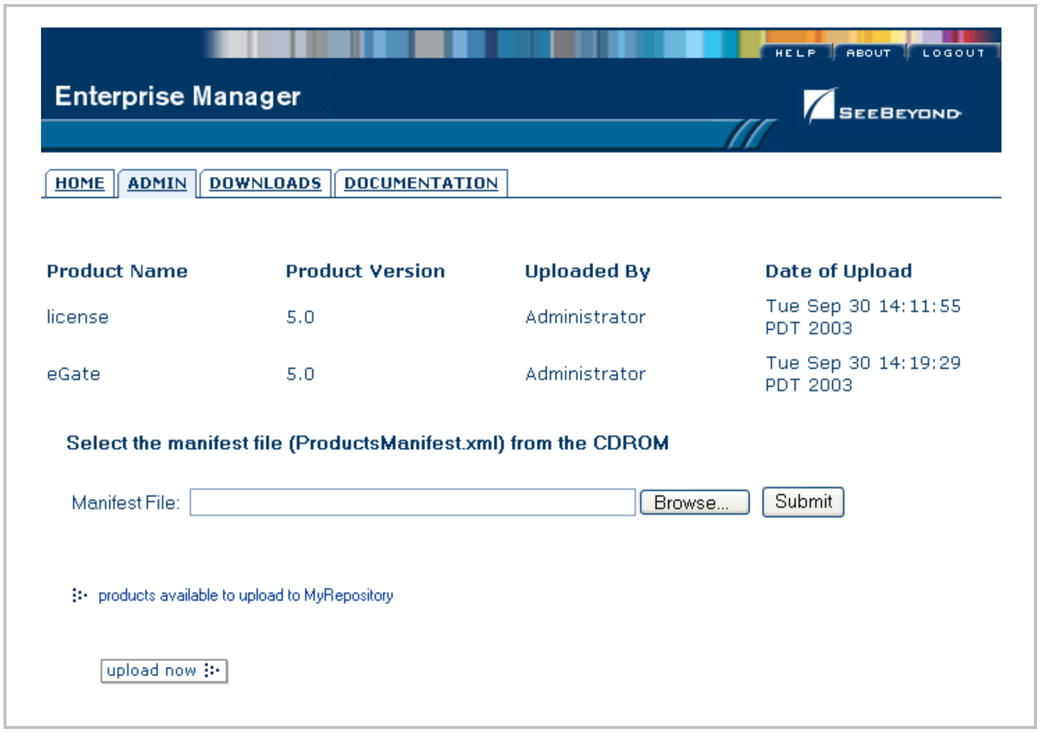
   The Repository server will have an HTTP address similar to the following:

   http://*<localhost>*/*<portnumber>*

   Where:

   <localhost> it the name of the machine running the Repository server, and *<portnumber>* is the access port you assigned to the Repository at installation time.
3. On the Enterprise Manager **Login** page, type your user user ID and password.
4. When Enterprise Manager starts, click the **ADMIN** tab. See **Figure 3 on page 22**.

Figure 3 Enterprise Manager ADMIN page



Product Name	Product Version	Uploaded By	Date of Upload
license	5.0	Administrator	Tue Sep 30 14:11:55 PDT 2003
eGate	5.0	Administrator	Tue Sep 30 14:19:29 PDT 2003

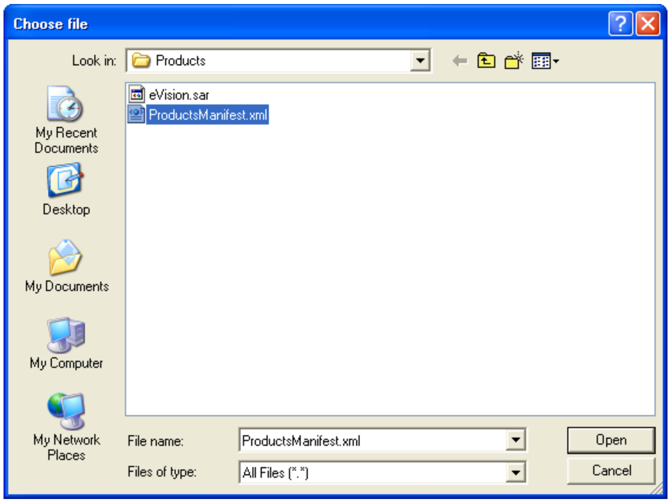
Select the manifest file (ProductsManifest.xml) from the CDROM

Manifest File:

5 On the ADMIN page, click **Browse**.

6 On the **Choose file** dialog box, select the **Product Manifest** file, **ProductManifest.xml**. See Figure 4

Figure 4 Selecting the Products Manifest file



Look in: Products

eVision.sar
ProductsManifest.xml

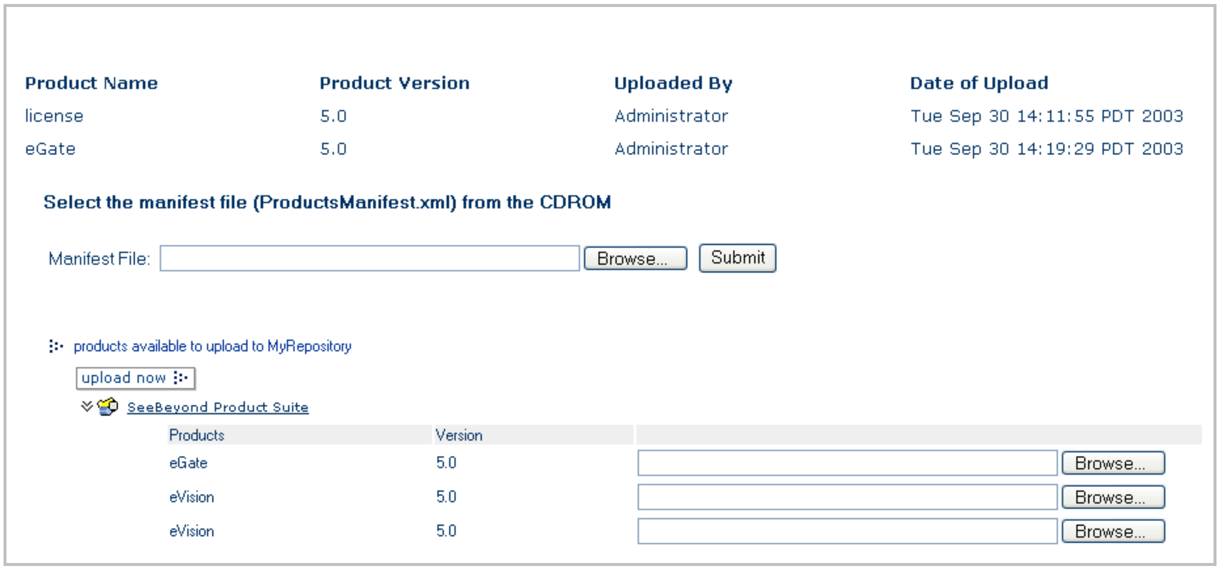
File name: ProductsManifest.xml

Files of type: All Files (*.*)

7 On the ADMIN page, click **Submit** to see the list of products available for you to upload.

The applications in the product manifest are set by the contents of your license agreement. See Figure 5.

Figure 5 ICAN Suite Products List



Product Name	Product Version	Uploaded By	Date of Upload
license	5.0	Administrator	Tue Sep 30 14:11:55 PDT 2003
eGate	5.0	Administrator	Tue Sep 30 14:19:29 PDT 2003

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

Manifest File:

products available to upload to MyRepository

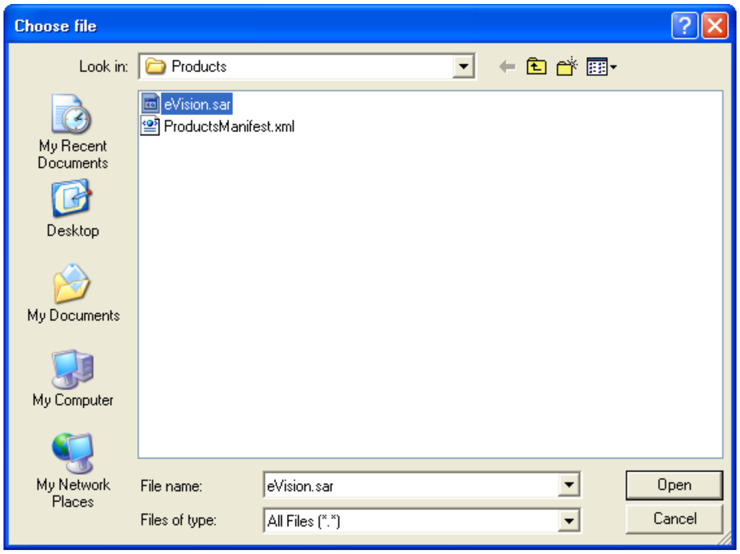
SeeBeyond Product Suite

Products	Version	
eGate	5.0	Browse...
eVision	5.0	Browse...
eVision	5.0	Browse...

8 Click **Browse** and locate the **eVision.sar** file.

9 On the **Choose file** dialog box, **eVision.sar**, and then click **Open**. See Figure 6,

Figure 6 eVision.sar file



Choose file

Look in: Products

eVision.sar
ProductsManifest.xml

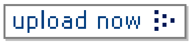
File name: eVision.sar

Files of type: All Files (*.*)

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

10 On the Enterprise Manager **ADMIN** page, click **Upload Now**. See Figure 7.

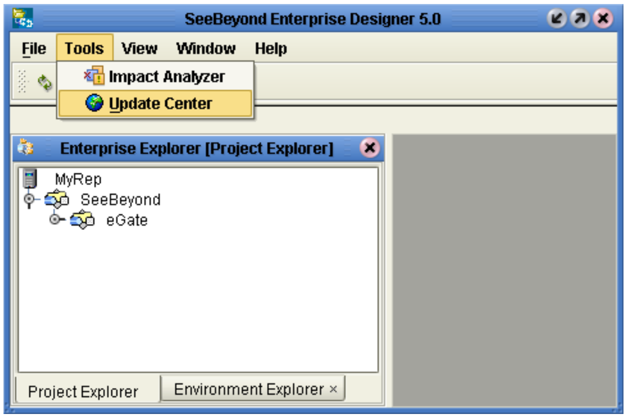
Figure 7 upload now button



11 Once the upload is complete, start Enterprise Designer.

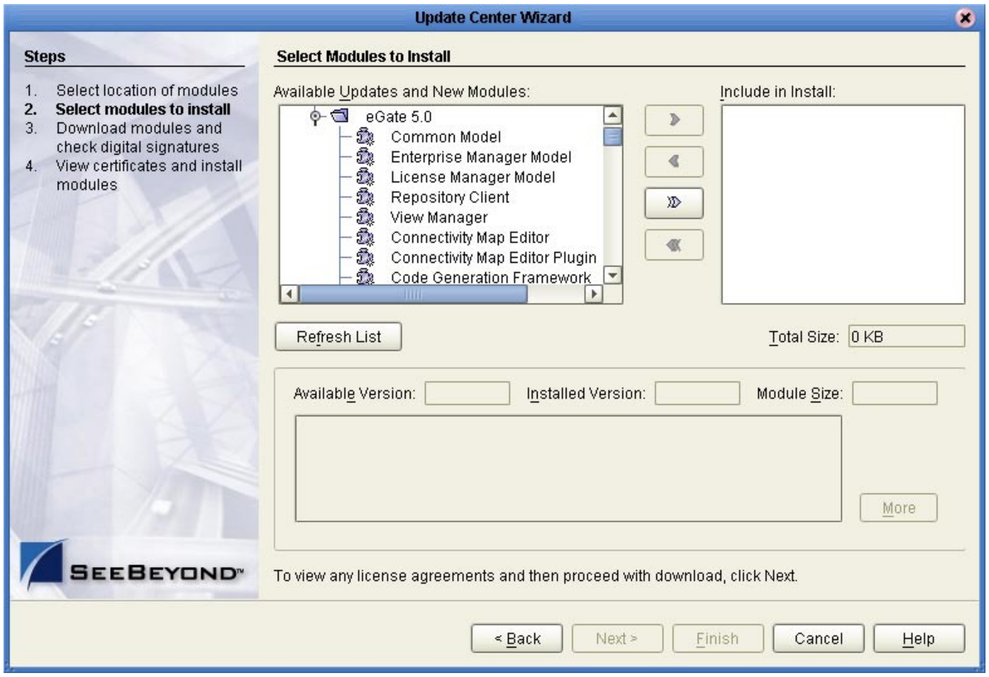
12 On the Enterprise Designer menu bar, click **Tools**, and then click **Update Center** to update the eVision Studio components. See Figure 8.

Figure 8 Updating eVision Studio



The Update Center appears with a list of components that are ready to update. See **Figure 9 on page 24**.

Figure 9 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

Total Size: 0 KB

Available Version:

Installed Version:

Module Size:

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

13 Select the eVision Studio module and the click action button with the dual, rightward-facing chevrons .

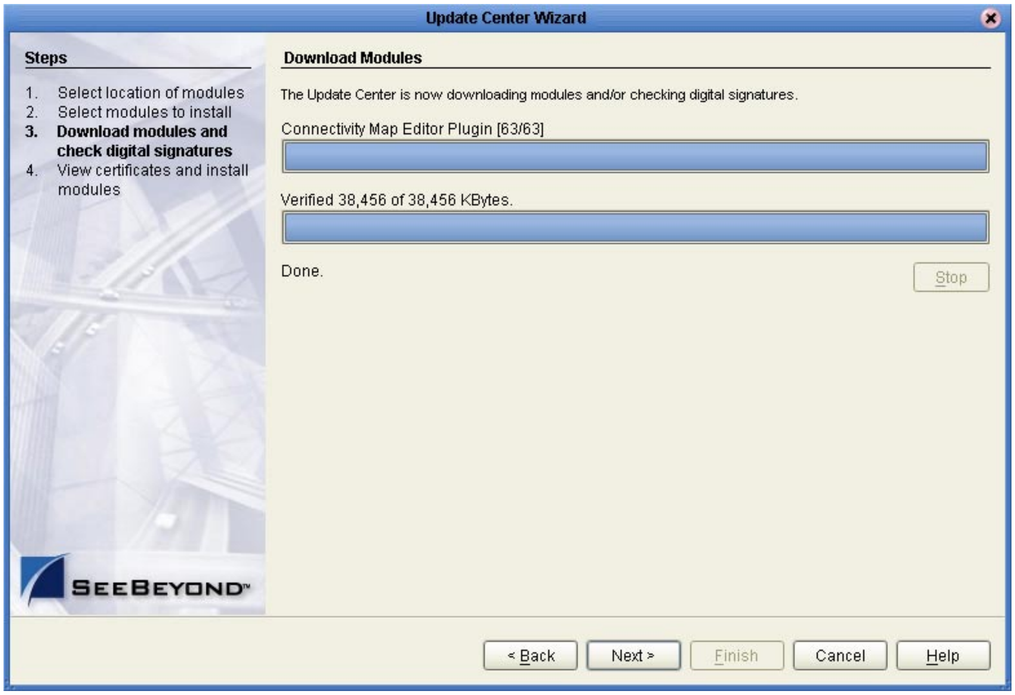
All modules are moved to the **Include in Install** section.

14 Click **Next.**

15 The files are downloaded.

Watch the progress bars as the files are downloaded. See Figure 10.

Figure 10 Update Center Wizard: Progress Bars



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

Download Modules

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

Connectivity Map Editor Plugin [63/63]

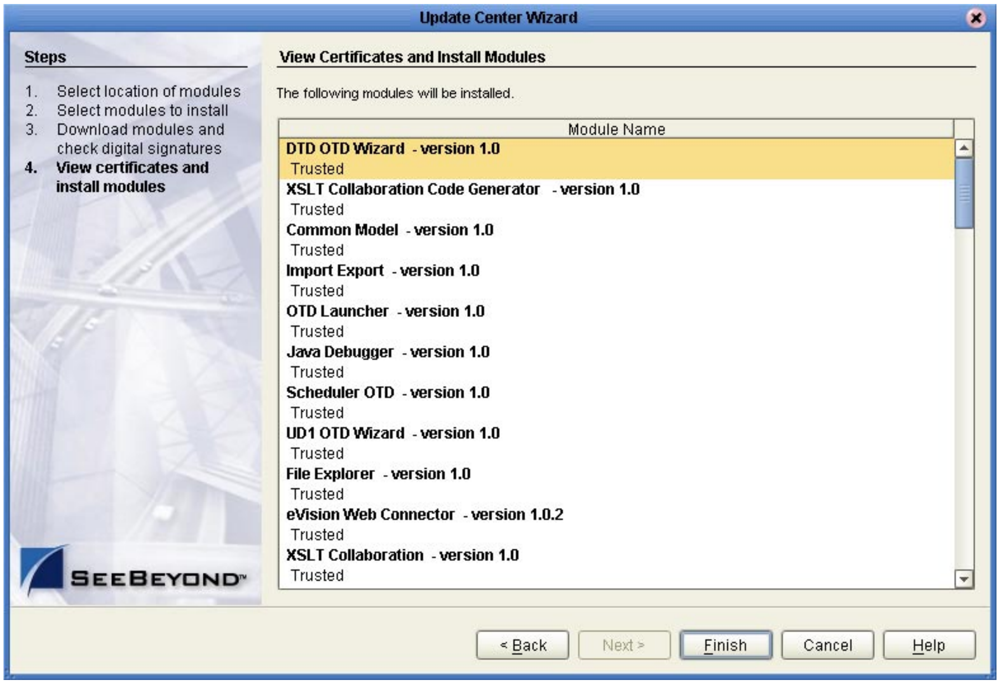
Verified 38,456 of 38,456 KBytes.

Done.

16 When the download process is complete, you will see **Done** displayed on the wizard.

17 The Update Center Wizard displays the list of Certificates and installed modules. See Figure 11.

Figure 11 Certificates and Installed Modules list



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**

View Certificates and Install Modules

The following modules will be installed.

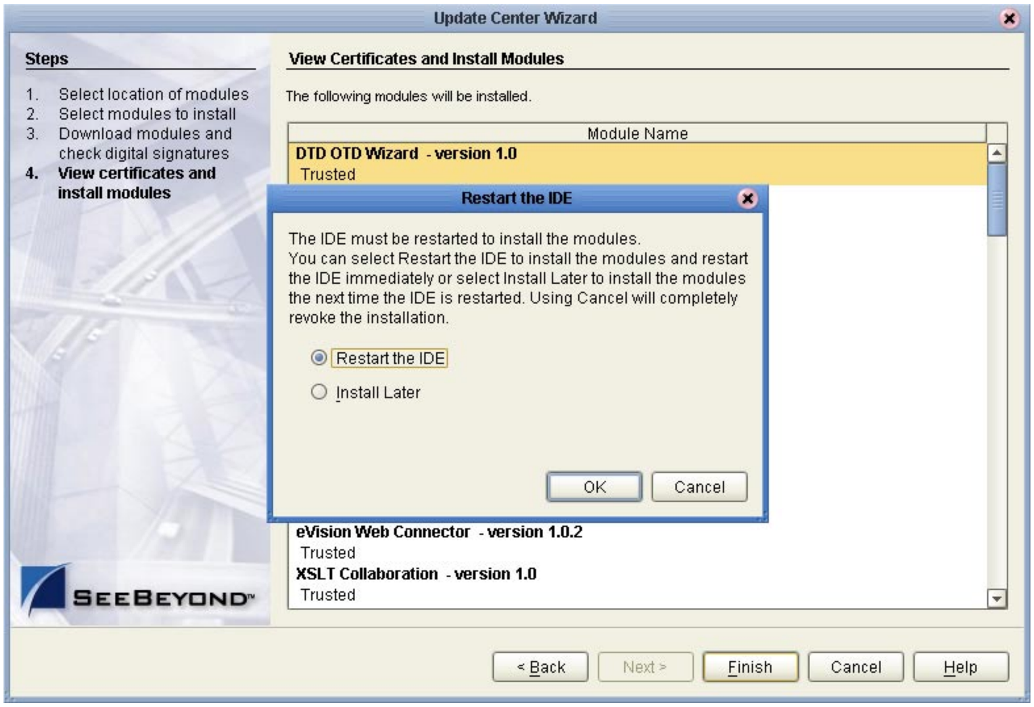
Module Name
DTD OTD Wizard - version 1.0 Trusted
XSLT Collaboration Code Generator - version 1.0 Trusted
Common Model - version 1.0 Trusted
Import Export - version 1.0 Trusted
OTD Launcher - version 1.0 Trusted
Java Debugger - version 1.0 Trusted
Scheduler OTD - version 1.0 Trusted
UD1 OTD Wizard - version 1.0 Trusted
File Explorer - version 1.0 Trusted
eVision Web Connector - version 1.0.2 Trusted
XSLT Collaboration - version 1.0 Trusted

18 Before continuing, briefly review what you have just installed.

19 Click **Finish**.

20 When the Update Center Wizard prompts you to do so, restart Enterprise Designer. See **Figure 12 on page 27**.

Figure 12 Update Center Wizard: Restart Enterprise Designer



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**

View Certificates and Install Modules

The following modules will be installed.

Module Name
DTD OTD Wizard - version 1.0 Trusted
eVision Web Connector - version 1.0.2 Trusted
XSLT Collaboration - version 1.0 Trusted

Restart the IDE

The IDE must be restarted to install the modules.
You can select Restart the IDE to install the modules and restart the IDE immediately or select Install Later to install the modules the next time the IDE is restarted. Using Cancel will completely revoke the installation.

● Restart the IDE

○ Install Later

When Enterprise Designer restarts, the installation of eVision Studio is complete. You are now ready to start using eVision Studio.

Chapter 7

eVision Studio Components

This chapter describes the components in the eVision Studio GUI.

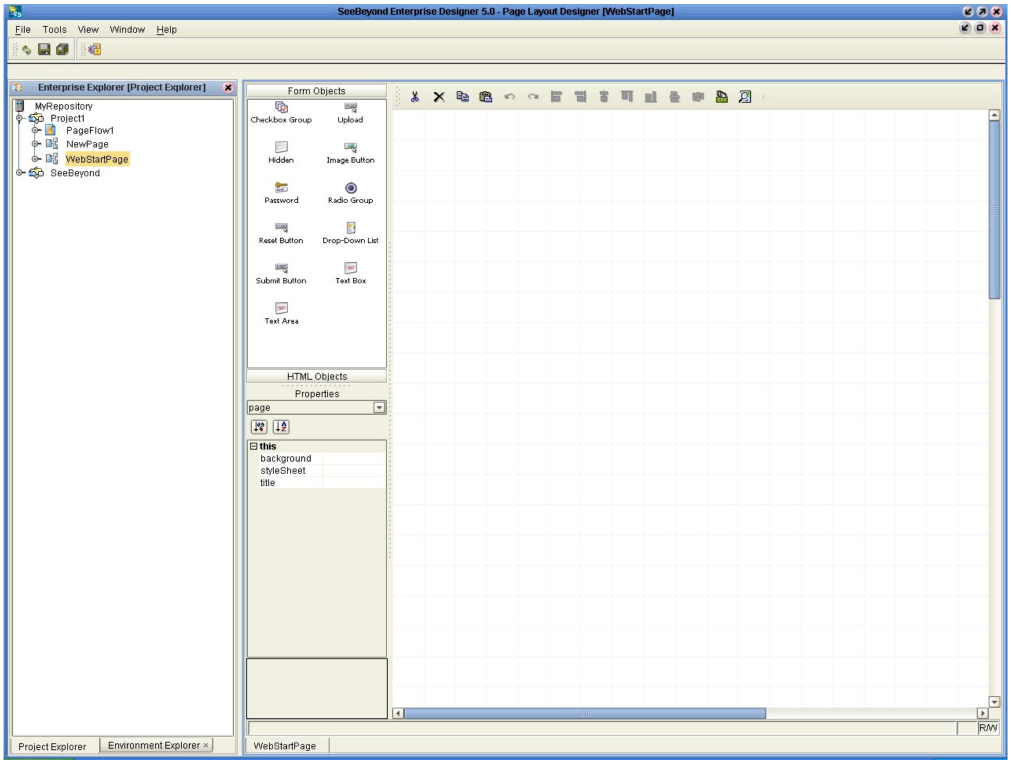
This Chapter Includes:

7.1 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. Pre-built components are dragged from a convenient component palette and positioned on the design canvas; the component's property sheet is opened automatically, allowing you to add attributes, such as the component name and the user-facing text to be displayed on Web page objects. As you lay out your Web pages, you can preview the design in your browser at any time throughout the process.

An example of eVision's GUI, **Page Layout Designer** with the **Form Objects** palette active is shown in **Figure 13 on page 29**.

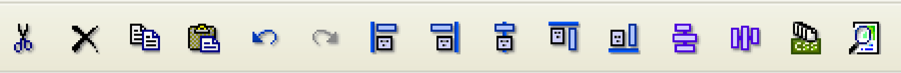
Figure 13 Page Layout Designer canvas at startup



7.1.1 eVision Studio Toolbar

The eVision Studio toolbar allows you to manipulate the objects you have placed on the **Page Layout Designer** canvas. This section describes the eVision tools and their functions. The eVision toolbar is shown in Figure 14. For a description of these tools, see **"eVision Studio Tools" on page 30**.

Figure 14 eVision Studio toolbar



7.1.2 eVision Studio Tools

eVision Studio's tools are described below.

Table 2 eVision Studio Web page design tools

Tool	Name	Purpose
✂	Cut Component	Cuts a component from the canvas a 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 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.
	Horizontal Spacing	Creates equal horizontal spacing between objects in a group of throw or more, based on an averaging algorithm.
	Vertical Spacing	Creates equal vertical spacing between objects in a group of three or more, based on an averaging algorithm.

Tool	Name	Purpose
CSS	CSS	Launches the **Choose CSS to Edit** dialog box, which allows you to select an imported style sheet to edit with the **Style Sheet Editor** tool.
🔍	Preview	Previews the Web page layout in your browser.

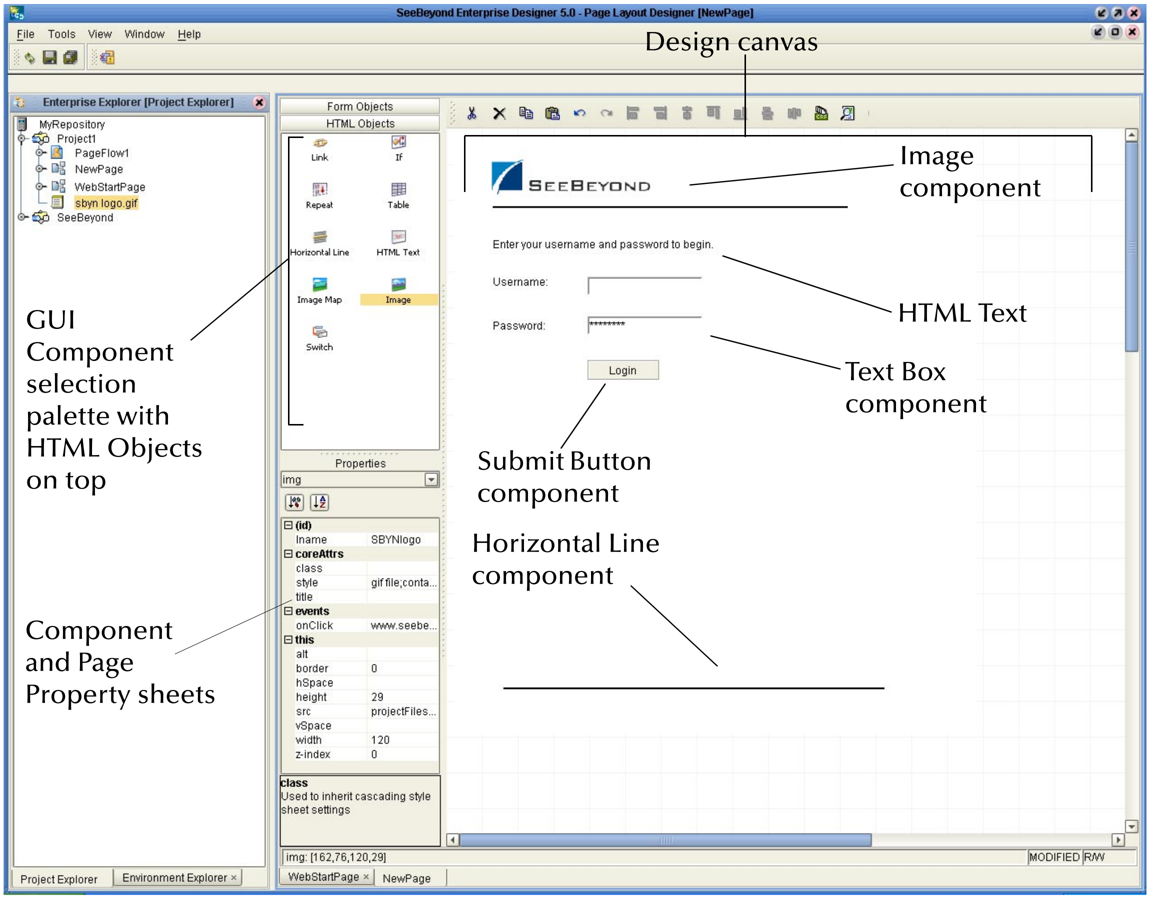
7.2 Graphical User Interface Components

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 either the **HTML Objects** or **Form Objects** palette, drag it onto the canvas, and release the mouse button to set the component's location on the page. GUI components (HTML and Form objects) comprise familiar Web-centric design elements including check boxes, check box groups, text boxes, 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, it'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 users in a browser.

In Page Layout Designer. the upper left pane contains the Graphical User Interface (GUI) Component libraries. See **Figure 15 on page 32**.

Figure 15 Page Layout Designer components



Design canvas

Image component

GUI Component selection palette with HTML Objects on top

Enter your username and password to begin.

Username:

Password: ********

HTML Text

Text Box component

Submit Button component

Horizontal Line component

Component and Page Property sheets

Properties

img

⊟ **(id)**	
lname	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

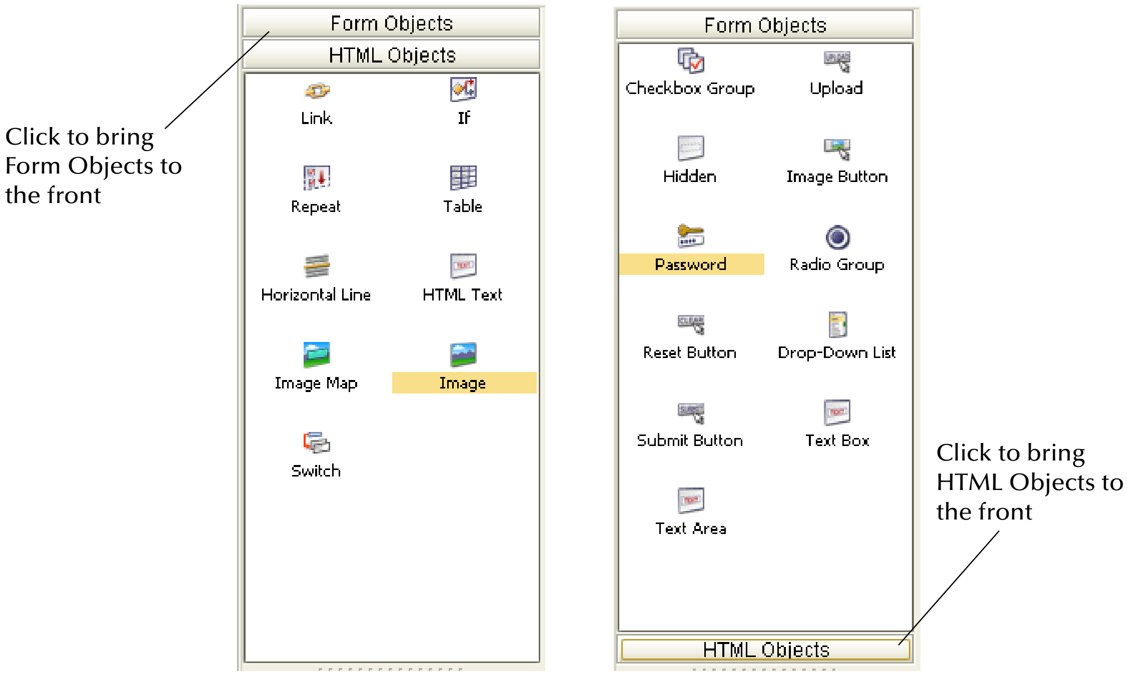
class
Used to inherit cascading style sheet settings

img: [162,76,120,29]

7.3 GUI Component Selection Tables (object palettes)

eVision Studio provides component selection tables (object 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, you then follow up by customizing its functional and visual presentation properties in the **Component Properties** window.To access HTML Objects, select the **HTML Objects** title bar. To access Form Objects, select the **Form Objects** title bar. See Figure 16 on page 33.

Figure 16 HTML and Form Objects palettes



Click to bring Form Objects to the front

Click to bring HTML Objects to the front

7.3.1 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 to an external Web site. Page links are created using the **Page Link Wizard**. For more information, see **“Linking Pages” on page 51**.
	Repeat	A repeating element is used when data needs to collected to dynamically populate a table or field with values based on recurring runtime data.
	Horizontal Line	Places a horizontal line on the canvas. The line is initially of a fixed length. To extend the line, place the pointer on the end you want to change (right or left). The pointer will change to a “resize” arrow (**<-->**). Drag the line end to modify the length. Use the Horizontal Line object to create visual separations in your layouts.

Component	Name	Purpose
	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 on image maps, see**"About Image Maps" on page 35**.
	Switch	Allows users 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.
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 selection palettes. A table can contain both *static* and *dynamic* rows. For more information, see **"About Dynamic Tables" on page 35**.
HTML	HTML Text	Creates a field on the canvas to hold HTML (static) text, or can act as a place holder for dynamic text. You use this component for labels and general information on a page.
	Image	Holds a static graphic image. To place a graphic image in the Image place-holder, open your Project folder in Enterprise Explorer, navigate to the image you want to import, and select the image. **Note:** *Images must reside in your Project in the Repository before you can access them.* For more information on importing images, see **"Importing Images" on page 38**

7.3.2 About 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 (it will be highlighted in blue).
2. Right-click to display the table context menu.
3. On the context menu, click **Set as Dynamic Row**.

The row will be clearly marked with bold text as being dynamic.

7.3.3 About Image Maps

Image maps are created 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 within the Page Flow. The following section describes how to create and add a link to an image map.

Important: *Before you begin, you must import the images that you want to use into your Project. For more information, see* **"Importing Images" on page 38**.

To Create an Image Map

1. To create an **Image Map** GUI component, from **HTML Objects** selection palette, drag the **Image Map** icon onto the canvas.

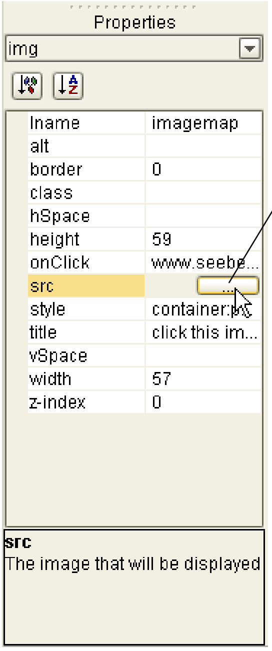
   When the icon transfers onto the canvas it changes into a dashed-line rectangle.

2. Drop the rectangle onto the canvas.

   The rectangle will change to a blue box with a gold outline, which represents the outside boundaries of the image container.

3. In the **Image Map** Properties sheet, click the **Command** button (**...**). See **Figure 17 on page 36**.

Figure 17 Image Map Properties sheet



Properties

img

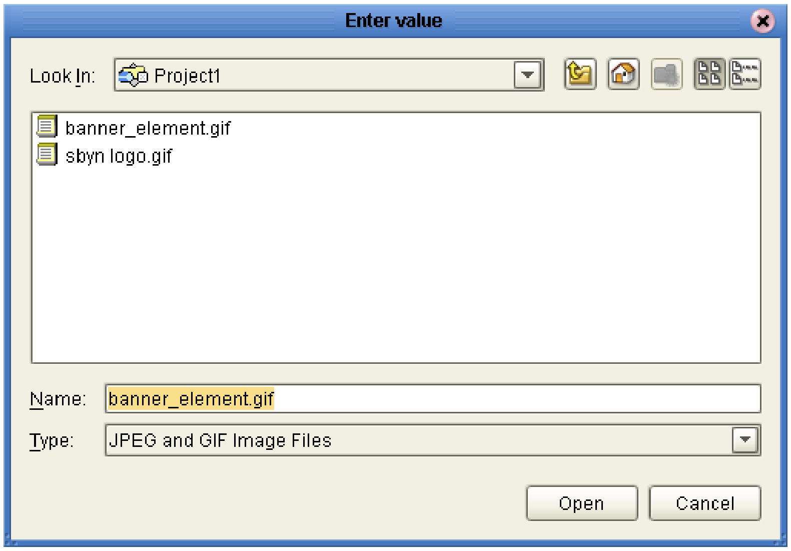
lname	imagemap
alt	
border	0
class	
hSpace	
height	59
onClick	www.seebe...
src	...
style	container:pc
title	click this im...
vSpace	
width	57
z-index	0

src
The image that will be displayed

Click to explore to the image

The **Enter value** dialog box appears. See Figure 18.

Figure 18 Selecting an image for an Image Map



Enter value

Look In: Project1

banner_element.gif
sbyn logo.gif

Name: banner_element.gif

Type: JPEG and GIF Image Files

4 Select the image you want to place in the container.

Remember that images are available only if they have been previously imported into the Project. See **"Importing Images" on page 38**.

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

A Under the **Value** heading, select the **lname value** field (the default name will be **img(*n*)**).

B In the **Value** field, delete the default name, and type a name for the component (for example **imagemap1**).

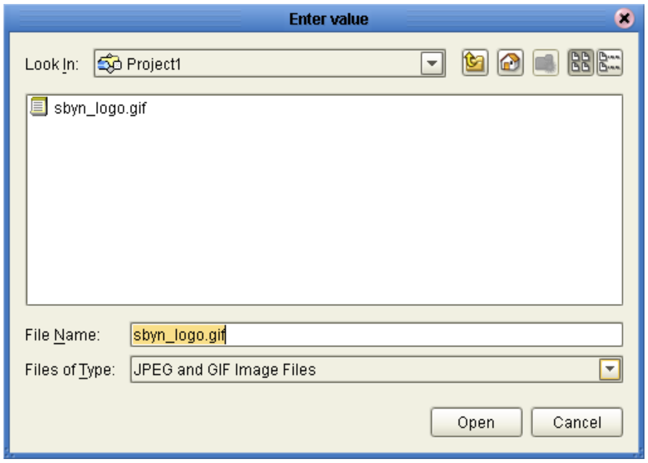
lname is the internal, logical name for the component, which identifies the component in the Page Flow.

6 In the **Component Properties** sheet, define the value for **src** (the image source file) by doing the following:

A Select the blank value field for the **src** property to start the search.

B Click the **Command** button (**...**) to launch the **Enter value for "src"** dialog box. See Figure 19.

Figure 19 Searching the Project for an image file



C Using the **Enter value for "src"** dialog, explore to the image file you want to add to the image map container.

D To link the image to the container, double-click the image, or select the image and click **Open**.

The image will appear in the image container.

7 To resize the container, click and drag a highlighted boundary element. The shape of the bidirectional arrow cursor will tell you which way the container will be resized. To move the container, click the container in the center (away from the edges), and drag it to the new location.

Note: *To be accessible, all images must be imported into the Project. See* **"Importing Images" on page 38** *for instructions.*

8 To add links to the image, do any of the following:

- In the **Enterprise Explorer** tree, drag a link and drop it onto the image. This link will have predefined parameters and a target location.

- From the **HTML Objects** palette, drag a link icon onto the image. This link will require that you specify link parameters and a target location.

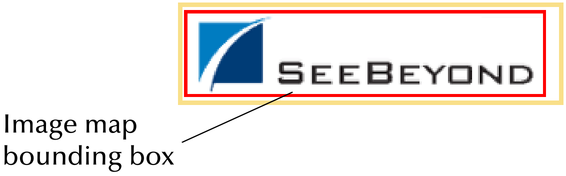
  When you drop a link onto the image, a highlighted rectangle appears.

Note: *An image map holds a link area. The link area enables you to link-edit and attach a target destination to theimage map.*

9. Modify the image map area, if necessary. See Figure 20.

   - To resize a mapped area, right-click an edge element on the rectangle and drag it in the desired direction.
   - To move a mapped area, right-click the approximate center of the rectangle and drag it to the desired location.

Figure 20 Resized bounding box for Image Map



10. When you have finished, click **OK**.
11. On the **eVision** toolbar, click **Preview** to test the link on the Preview page.

    In the preview image, when you pass the pointer over the mapped area, the cursor will change to the “hand” icon.
12. Click the defined image area to execute the link code.

Importing Images

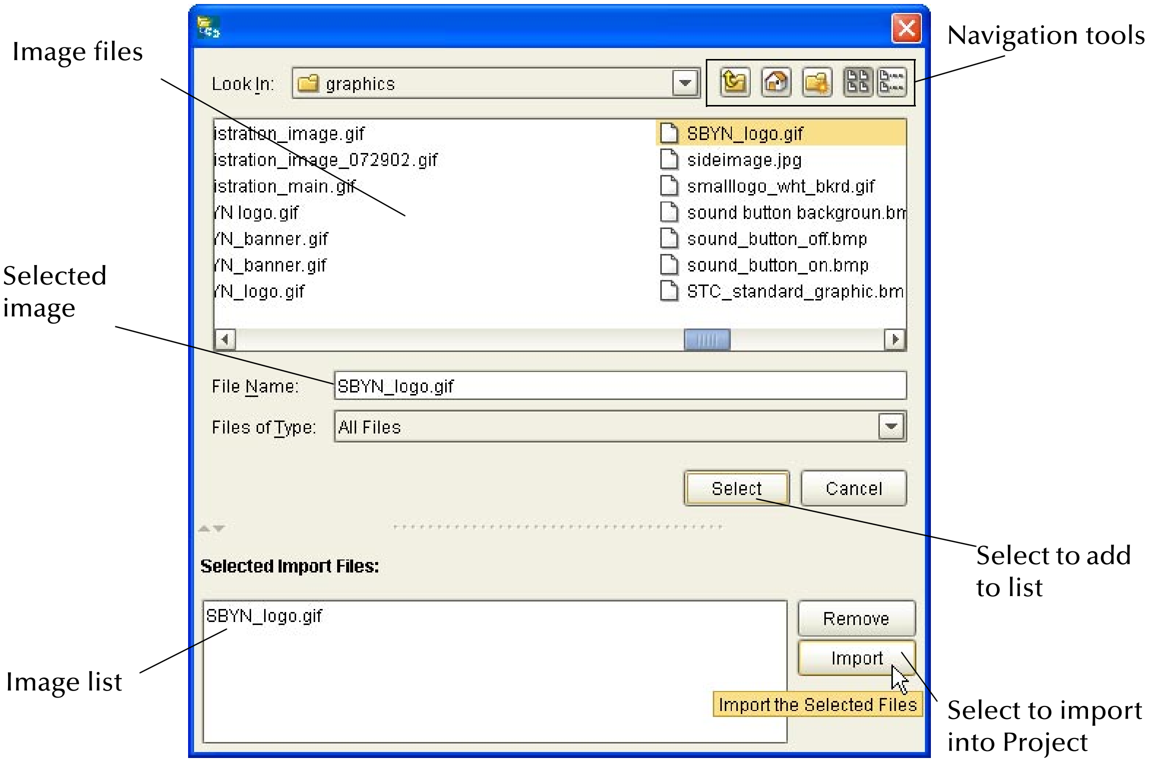
To make graphic images accessible to your Web page, they must first be imported into your Project from their location in the Repository.

To Import a Graphic Image

1. In **Enterprise Explorer**, right-click the **Project** icon.
2. On the context menu, click **New**, and then point to **New File**.
3. In the dialog box that appears, explore to the location of the file or files that you want to import into your Project.

   The image files can reside anywhere on your machine or a network. See Figure 21.

Figure 21 Importing images into a Project



Look In: graphics

File Name: SBYN_logo.gif

Files of Type: All Files

Selected Import Files:

SBYN_logo.gif

4 Select the image file or files that you want to import.

5 On the dialog box, click **Select**.

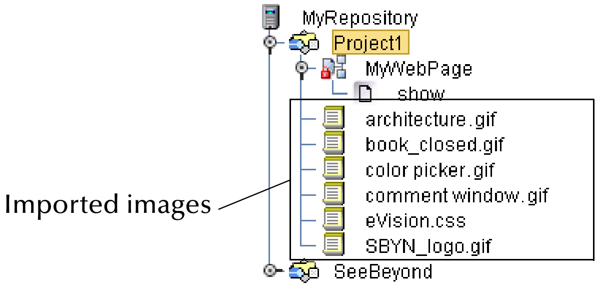
6 Select additional images, if desired.

Click **Select** add an image to the list.

7 To remove an image from the list, select it and then click **Remove**.

8 To import the image or images into your Project, click **Import**.

Figure 22 Imported images in Enterprise Explorer



The images will be displayed under your Project in the **Enterprise Explorer** tree. See Figure 22.

7.3.4 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
☑	Check Box Group	Creates a group of check boxes. Indicates inclusive user selection at run-time
▭	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.
••••	Password	Creates a password input box with asterisks that mask the password.
CLEAR	Reset Button	Allows the user to cancel an operation or reset values to a default condition.
SUBMIT	Submit Button	Allows the user to launch an operation or submit text to the application. Passes accumulated values to the eGate system.
TEXT	Text Area	Allows the user to type an extensive body of text and place it anywhere on the Web page without disturbing formatting.
UPLOAD	Upload Button	Allows the user to upload files and data to an application.
▣	Image Button	Creates an image “hotspot” that the user can click to perform and action, such as launching an application or jumping to another internal Web page or an external Web site.
◉	Radio Button	Switches an attribute or condition on or off. **Note:** *The Value property cannot be empty. This object requires that a value be assigned in order to pass the true/false ondition.*
≡	Drop-Down List	Allows the user to select an entry from a drop-down list.
SUBMIT	Submit Button	Allows the user to launch an operation or submit text to the application. Passes accumulated values to the eGate system.

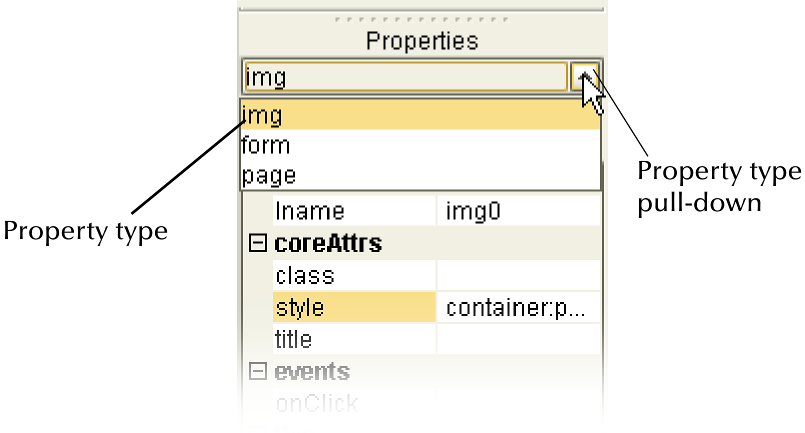
Component	Name	Purpose
TEXT	Text Box	Allows the user to type text that will be displayed on the Web page surrounded by a bounding box.

7.4 Defining Component Properties

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

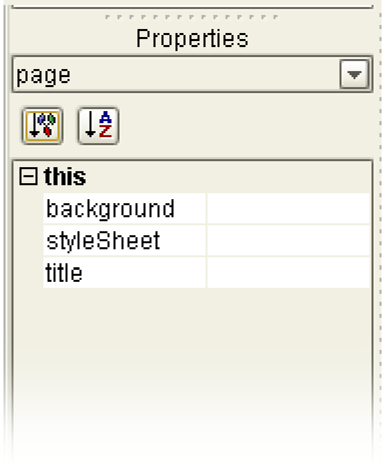
When you click an HTML or Form Object, its properties are automatically displayed in the Properties window. If you want to jump from one property type to another without clicking an object on the canvas, you can click the Properties drop-down arrow and click the property type you want to access from the drop-down menu. (A property type would typically be Page properties, image properties (img) , or HTML/Form Object properties.) See Figure 23.

Figure 23 Properties drop-down menu



For example, when you click the Page entry from the Properties drop-down, the Page Properties sheet pops to the front. See **Figure 24 on page 42**.

Figure 24 New Page properties

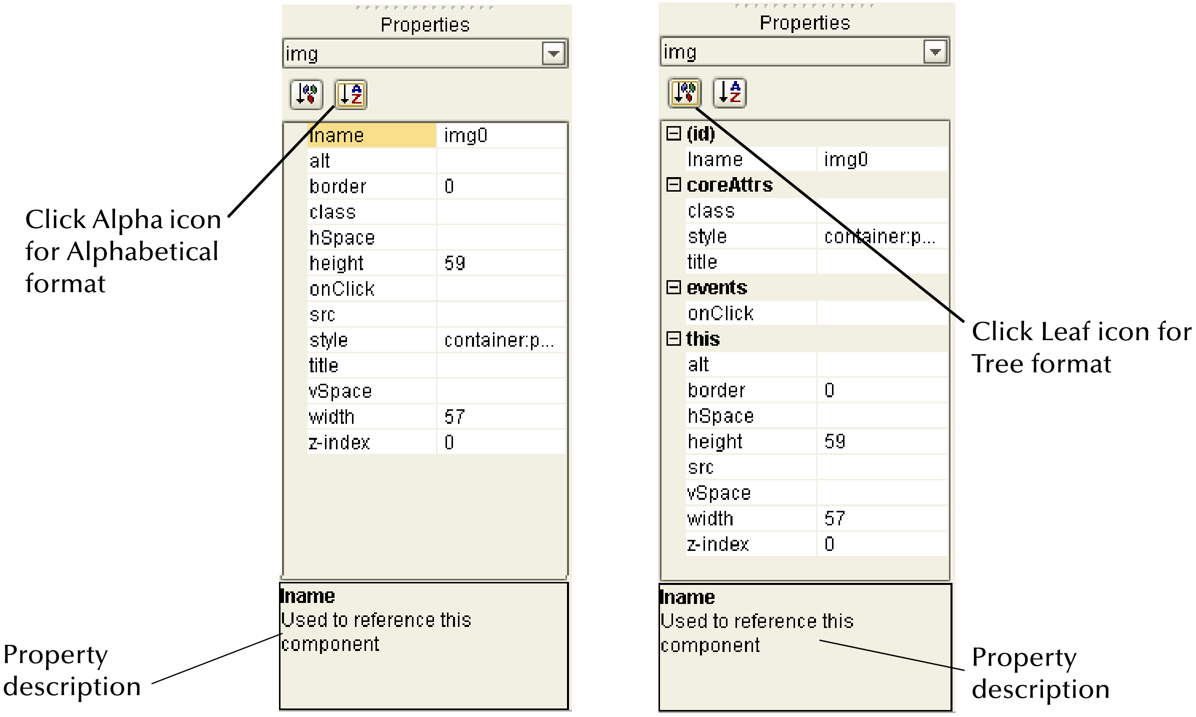


this	
background	
styleSheet	
title	

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, for example, a text string to be presented to Web application users at run-time.

A description of each property is provided in two ways: In a property description box at the bottom of the Properties window, and as a tooltip when you roll the pointer over a property name. You can also specify how the property list is displayed: In either alphabetical or Explorer (tree) formats. For alphabetical format, click the **alpha** icon. For tree format, click the **leaf** icon. See Figure 25.

Figure 25 Swapping Component Properties formats



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

Click Alpha icon for Alphabetical format

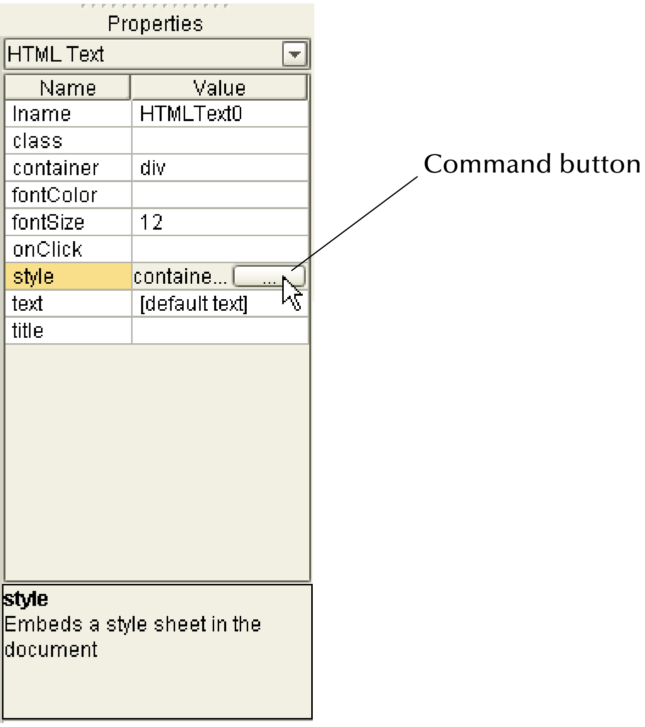
Click Leaf icon for Tree format

Property description

Property description

To define a property, select any field under the **Value** heading 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. See Figure 27 on page 43.

Figure 26 Command button

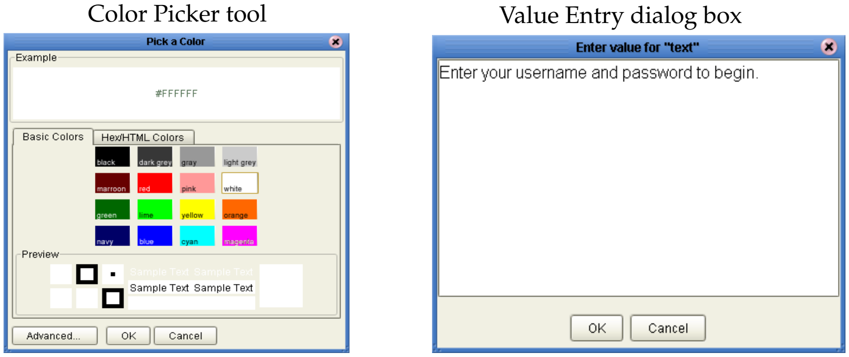


Name	Value
lname	HTMLText0
class	
container	div
fontColor	
fontSize	12
onClick	
style	containe...
text	[default text]
title	

style
Embeds a style sheet in the document

A **Command** button indicates that another window, tool, or dialog box will be triggered when the button is clicked. See Figure 27.

Figure 27 Interface elements displayed via a **Command** button



Color Picker tool

Value Entry dialog box

Enter your username and password to begin.

7.5 Customizing a Style Sheet

eVision Studio provides two style sheets for immediate use: *eVision-default.css* and *eVision.css*. To apply a style sheet, see **“Applying a Style Sheet to Your Web Application” on page 49**.

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 48**.

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

- You can edit one of the default style sheets.
- You can import a style sheet from an external source and modify it.

See **“Customizing a Style Sheet” on page 44**.

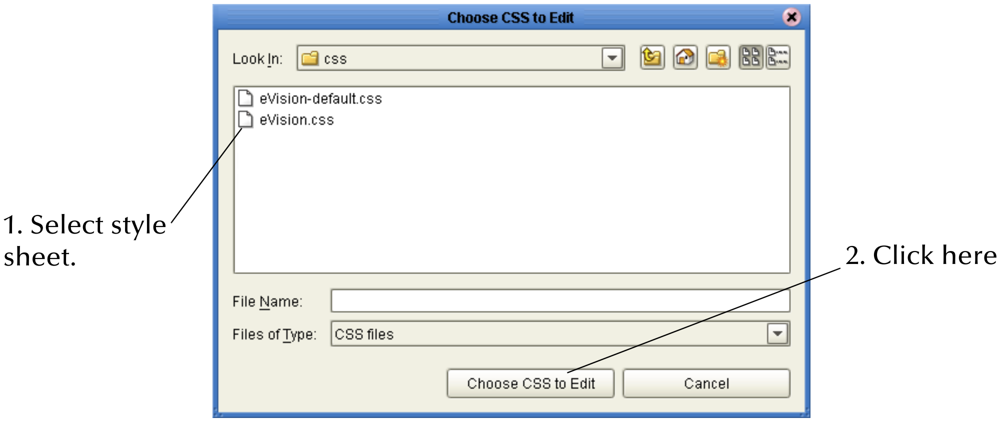
Creating a new style sheet from eVision.CSS

To create a custom style sheet you can start by modifying one of the default style sheets (for example, *eVision.css)* and use it as a starter template.

To create a new style sheet from the eVision.CSS style sheet, do the following:

1. On the eVision toolbar, click the **CSS** icon.
2. On the **Choose CSS to Edit** dialog box, click **eVision.css**
3. Click **Choose CSS to Edit**. See Figure 28.

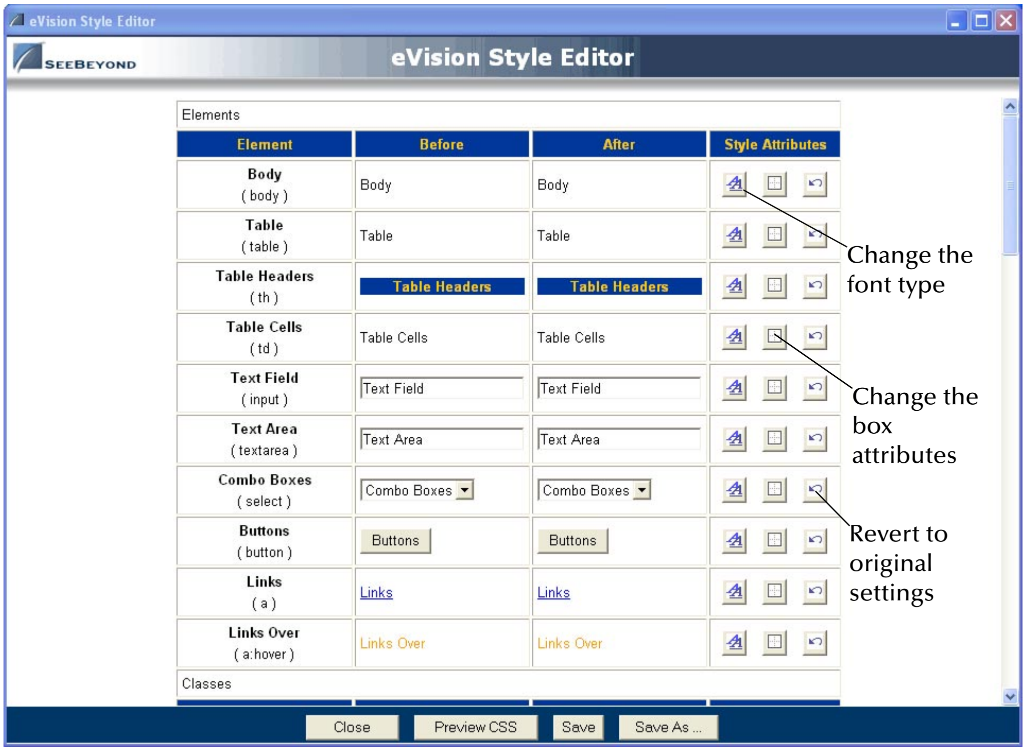
Figure 28 Choosing a style sheet to edit



eVision-default.css
eVision.css

The **eVision Style Editor** is displayed. This editor allows you to modify a host of properties that will be used to manage the format of a Web page. See **Figure 29 on page 45**.

Figure 29 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

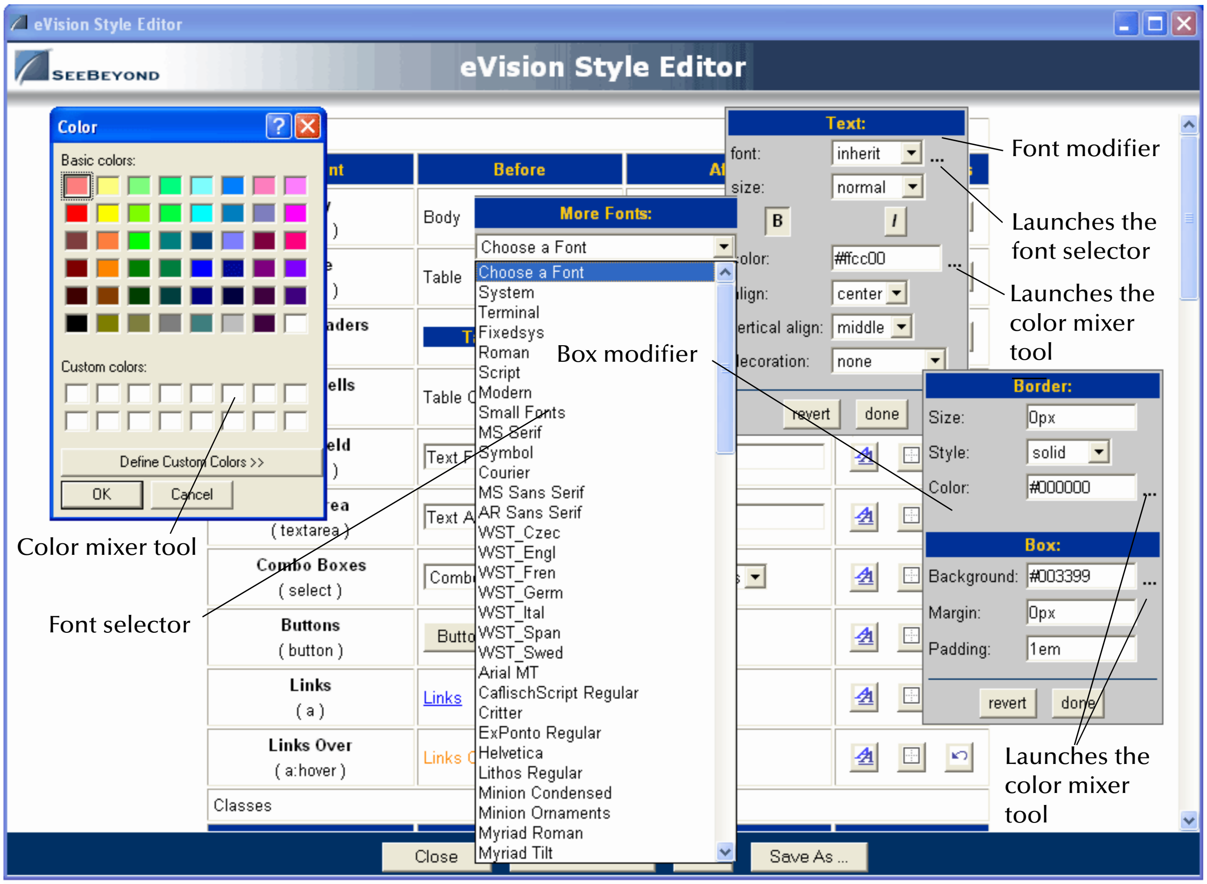
Revert to original settings

The **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 **Style Editor** tools, see **Figure 30 on page 46**.

Figure 30 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

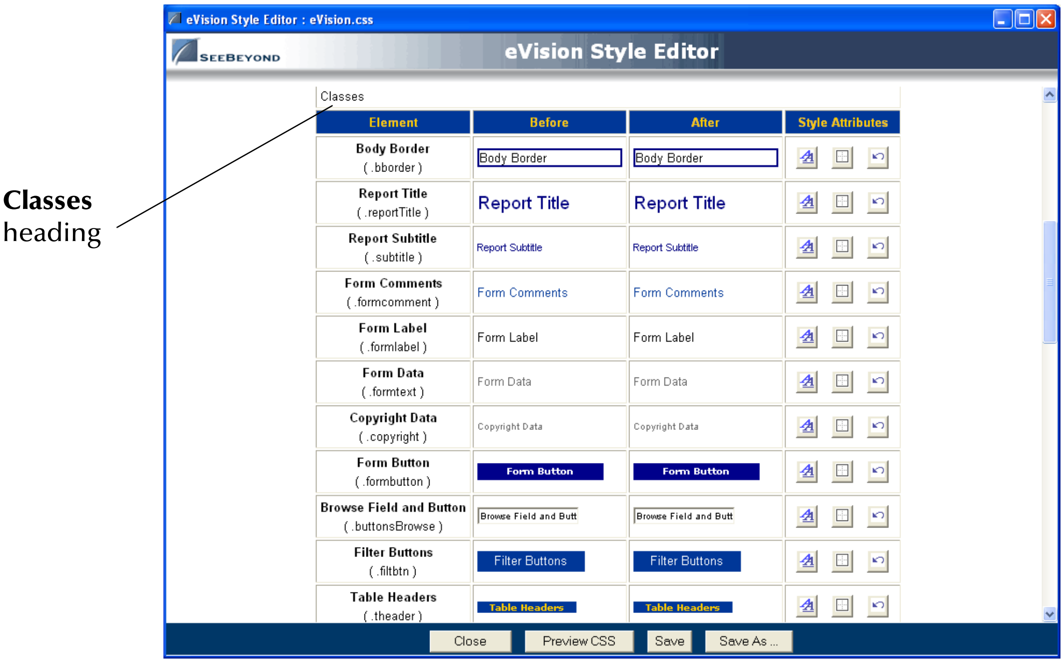
The **Style Editor** retains the original specifications for the elements you have changed; the original element configuration is shown under the "**Before**" heading. A real-time rendering of the elements you have modified is displayed under the "**After**" heading.

- To change back to the original configuration (reverse your changes), click **Revert**.
- To view the code you have generated, 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.

7.5.1 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 30 on page 46**.

Figure 31 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. After importing, follow the instructions in **"Applying a Class to an Object"**.

Applying a Class to an Object

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

1. On the design canvas, select the **HTML** or **Forms Object** to which you want to add the attributes of a class.

   The **Component Properties** sheet will open showing the object's properties.

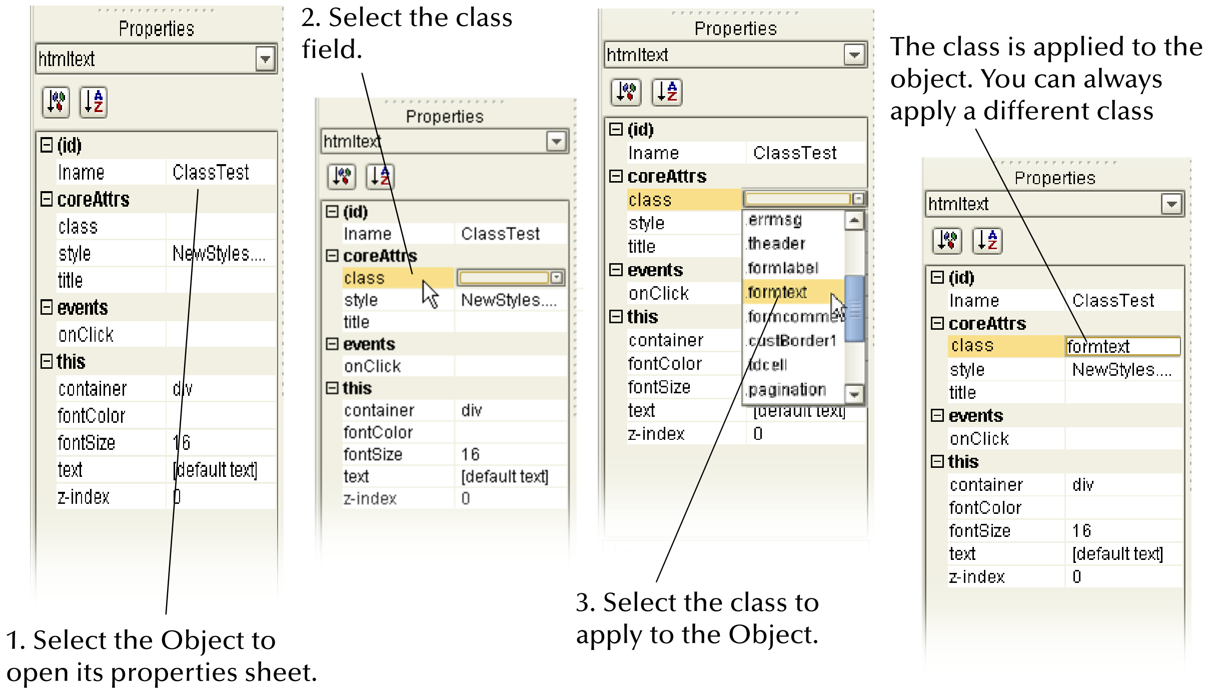
2. In the **Component Properties** sheet, select the **Class** field.

   A drop-down menu will appear listing 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 32 on page 48**.

Figure 32 Mapping a class to a component



1. Select the Object to open its properties sheet.

2. Select the class field.

3. Select the class to apply to the Object.

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

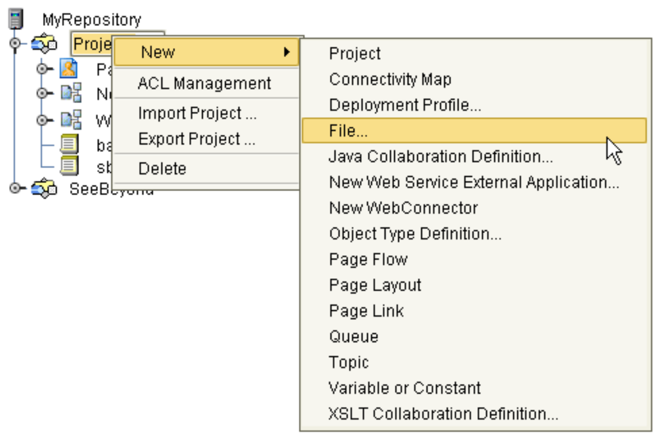
7.6 Importing a Style Sheet Into Your Project

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 **Enterprise Explorer**, right-click the **Project** icon.
2. On the context menu, point to **New**, and then click **New File**. See Figure 33.

Figure 33 Importing a style sheet as a new file



3 Using the **Explorer** window in the **Select Import Files** dialog box, explore to <*path*>**edesigner**, where:

<*path*> is the path to your **eGate** root directory, for example, ***c:\eGate50***, and **edesigner** is the folder containing the **Enterprise Designer** code.

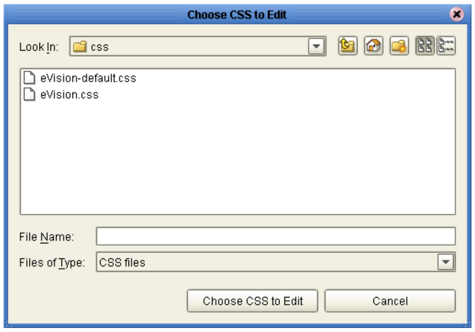
4 Double-click the **edesigner** folder.

5 Double-click your way through the following path to the **css** folder:

usrdir > modules > ext > stc > evision_core > tigris > css

See **Figure 33.**

Figure 34 CSS Chooser dialog box



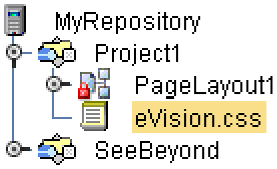
6 Double-click the **css** folder.

7 In the **css** folder, select one or more style sheets.

8 On the import dialog box, click **Choose CSS to Edit**, and then click **Import**.

The style sheet(s) will appear in your Project in **Enterprise Explorer**. See Figure 35.

Figure 35 Style sheet in Enterprise Explorer



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

7.6.1 Applying a Style Sheet to Your Web Application

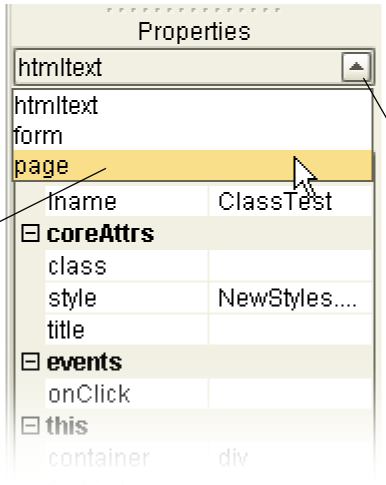
After importing the style sheet, apply it to the current Page.

1 In the **Component Properties** window, click the **Properties** drop-down arrows.

2 On the drop-down list, click **page**.

The **Page Properties** window will come to the front. See **Figure 36 on page 50**.

Figure 36 Changing the Properties window



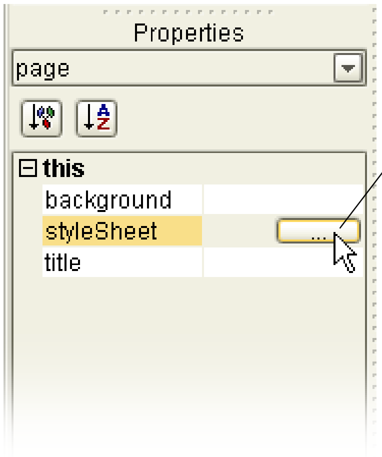
name	ClassTest
class	
style	NewStyles....
title	
onClick	
container	div

Select to swap to Page Properties

Click to expose property types

3 To launch the **Style Sheet Explorer** dialog box, click the **Command** button (**...**) in the **Page Properties** window. See Figure 37.

Figure 37 Exploring for a style sheet

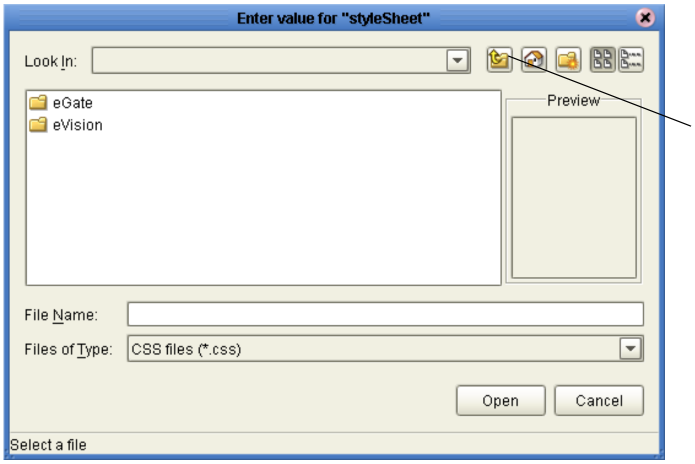


background	
styleSheet	
title	

Click to explore to the style sheet to apply to the page

4 In the **Style Sheet Selector** dialog box, click the “up” icon to move up one level in the file system. See **Figure 38 on page 51**.

Figure 38 Moving up a level in the CSS Explorer



Files of Type: CSS files (*.css)

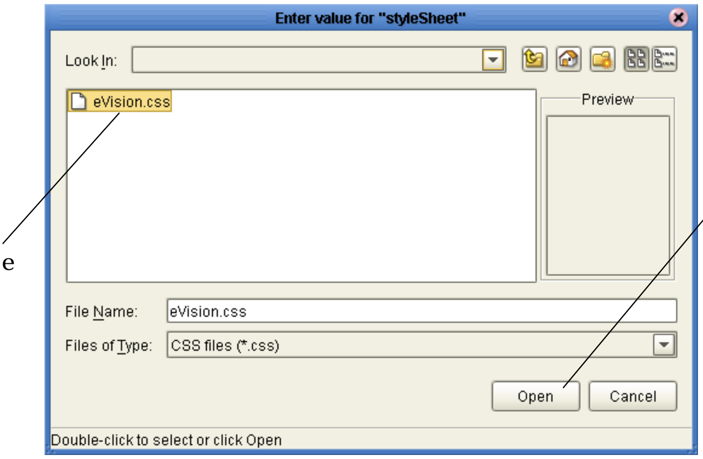
Select a file

Click to move up one level in the directory

5 Double-click the **Project** folder.

6 In the style sheet file list, select the style sheet that you want to apply, and then click **Open**. See Figure 39.

Figure 39 Selecting a style sheet to import into the Project



File Name: eVision.css

Files of Type: CSS files (*.css)

Double-click to select or click Open

Select imported style sheet file

Click Open

The style sheet is applied to the page.

7.7 Linking Pages

In eVision Web applications, links that point to internal application resources (other pages, applications, etc.), 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

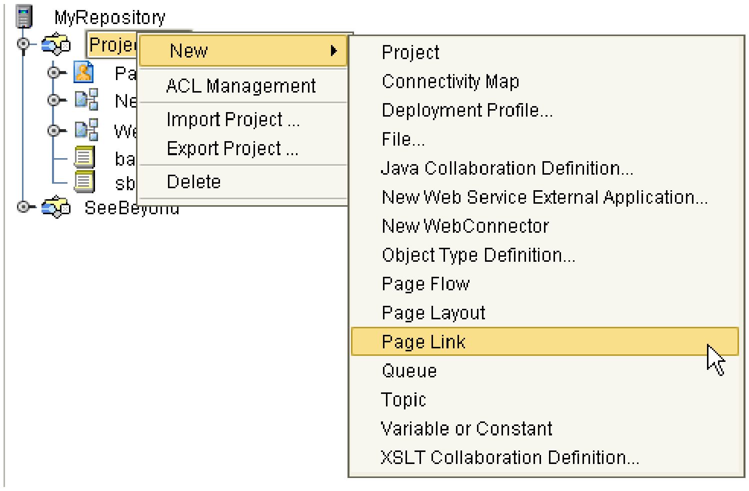
- Specify parameters to pass to other objects on destination pages

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. For more information, see **"About Event-based Linking" on page 97**.

To Create an Application Page Link

1. In **Enterprise Explorer**, right-click your **Project** icon.
2. On the context menu, point to **New**, and then click **Page Link**. See Figure 40.

Figure 40 Creating a Page Link



The **Page Link Wizard** appears. See **Figure 41 on page 53**.

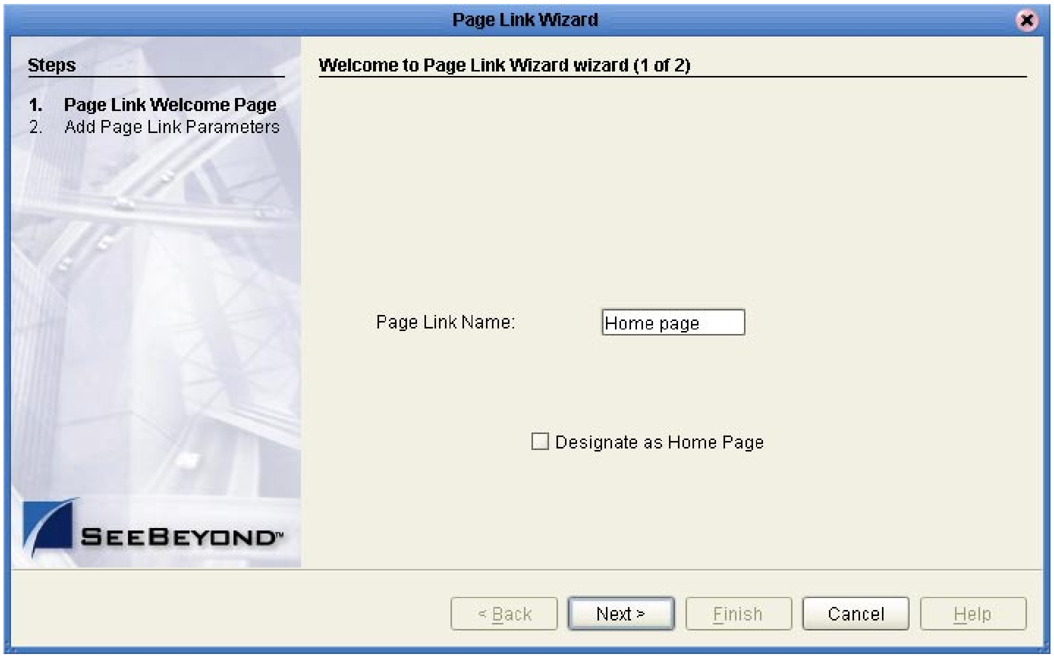
3. On the **Page Link Wizard**, in the **Page Link Name** field, type a descriptive name for the link.

   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 Homepage**.

   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.

Figure 41 Page 1 of 2 in the Page Link Wizard



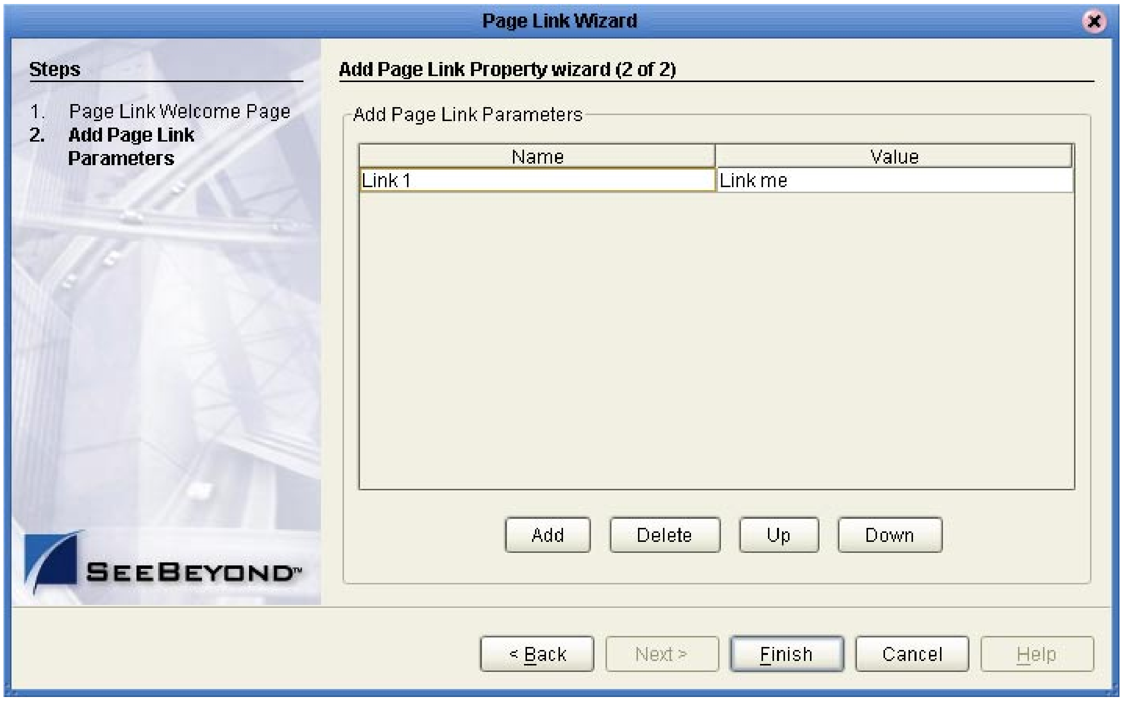
Welcome to Page Link Wizard wizard (1 of 2)

Page Link Name: Home page

□ Designate as Home Page

5 Click **Next**. See Figure 42.

Figure 42 Page 2 of 2 in the Page Link Wizard



Add Page Link Property wizard (2 of 2)

Add Page Link Parameters

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 off parameters.

On the 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.

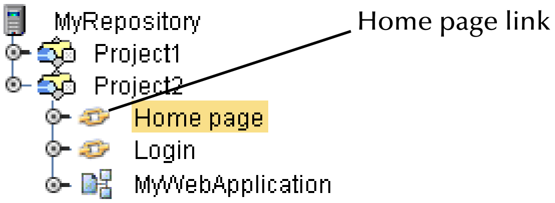
7 Modify the Page Link parameter fields, if necessary:

- 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 **Enterprise 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 78**.

Chapter 8

Using Page Layout Designer

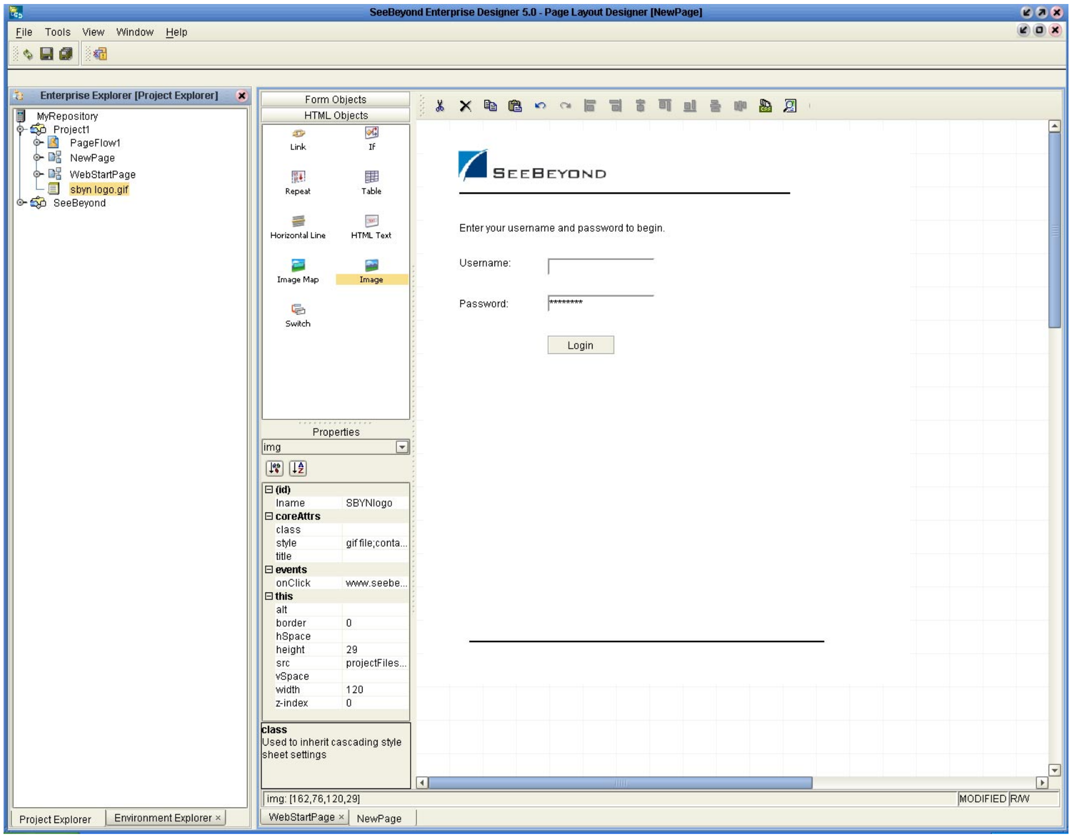
This chapter explains how to start eVision, start a new page, place objects on the canvas, and change the size and location of objects individually or in a group.

This Chapter Includes:

8.1 Laying Out a Web Page

This section describes how to build a simple Login page using eVision's GUI Components. 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. An example of the finished page is shown in **Figure 44 on page 56**.

Figure 44 Login Page Layout Designer project



Enter your username and password to begin.

Username:

Password: ********

(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

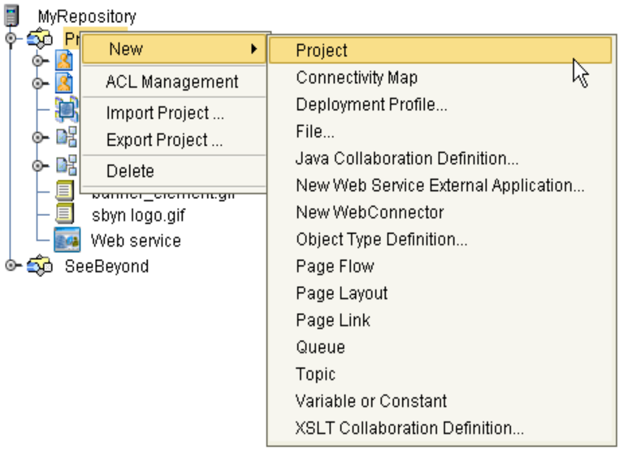
class
Used to inherit cascading style sheet settings

img: [162,76,120,29]

8.1.1 Starting a New Page Layout

1. Start **Enterprise Designer**.
2. In **Enterprise Explorer**, click the **Repository** icon and right-click the mouse to expose the context menu.
3. On the context menu, click **Project**. See **Figure 45 on page 57**.

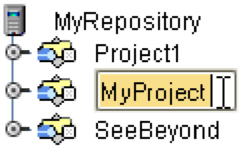
Figure 45 Project context menu



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 46.

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

Figure 46 Naming a new Project

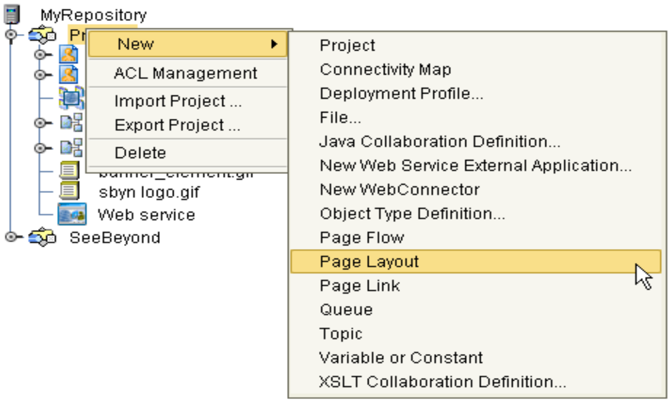


You do not need to press **Enter**; the next mouse click outside the name box will set the Project name.

5 In the **Enterprise Explorer** tree, select the **Project** icon for your new Project, and right-click the mouse to expose the **New** context menu.

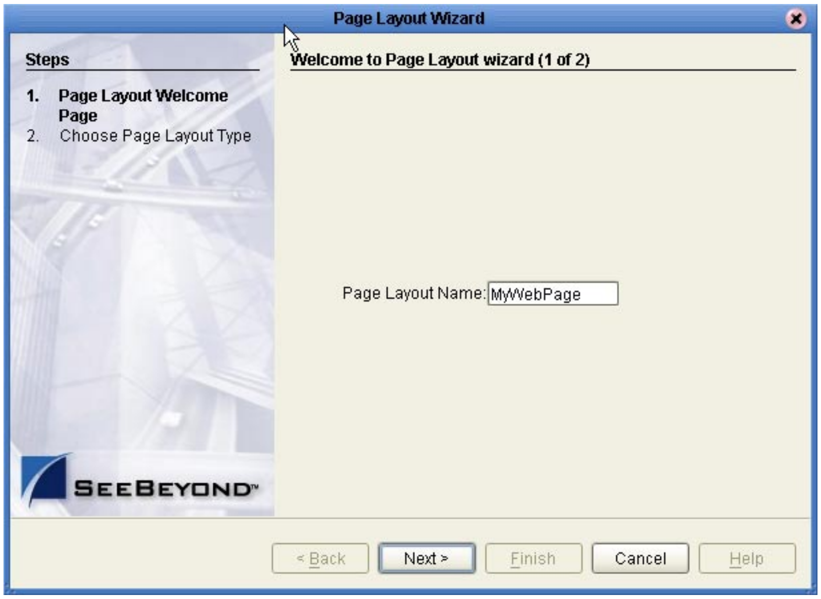
6 On the context menu, click point to **New**, and then click **New Page Layout**. See **Figure 47 on page 58**.

Figure 47 New Page Layout context menu



The Page Layout Wizard appears. See Figure 48.

Figure 48 Page Layout Wizard page 1



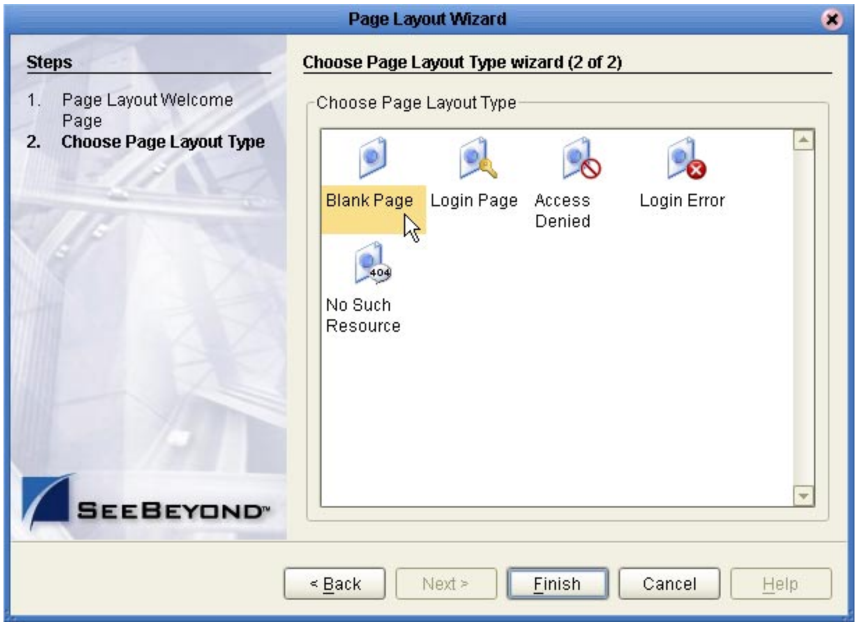
Welcome to Page Layout wizard (1 of 2)

Page Layout Name: MyWebPage

7 In the **Page Layout Name** box, type a unique name for the new Page Layout.

8 Click **Next**.

Figure 49 Page Layout Wizard page 2



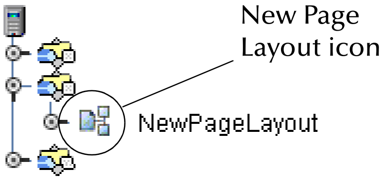
On the Wizard page shown in Figure 49, you could select any one of the preconfigured Web pages. Because you are creating a Login Page, do not select a preconfigure page.

9 In the **Choose Page Layout Typ**e window, select the **Blank Page** icon.

10 Click **Finish**.

A new **Page Layout** icon is added to the **Project Explorer** tree. See Figure 50.

Figure 50 Page Layout icon in Project Explorer



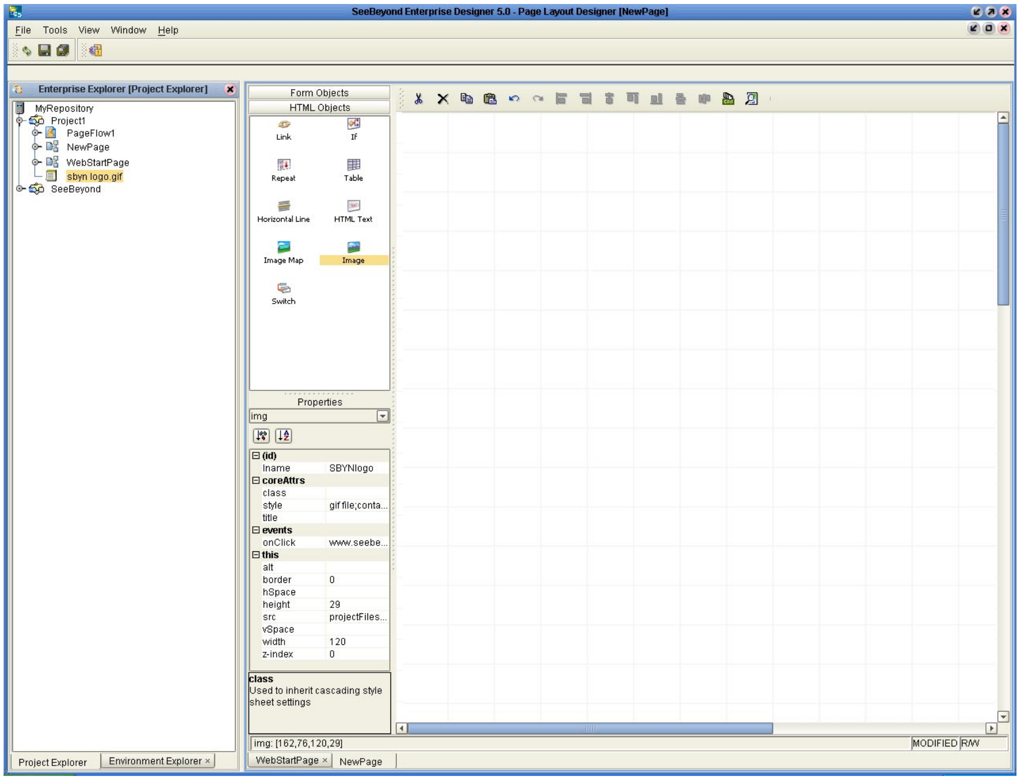
The name you provided in the Page Layout Wizard is appended to the eVision application icon name. The **eVision Studio** GUI is appears in the work area of Enterprise Designer with a blank canvas. See **Figure 51 on page 60**.

You have just created a new, blank Web page and started Page Layout Designer. The following section describes how to create a simple “login” Web page using the Page Layout Designer tools and pre-built GUI components.

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 48. *Henceforth, Cascading Style Sheets are referred to simply as "style sheets."*

Figure 51 eVision GUI with a blank canvas



img

⊟ **(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]

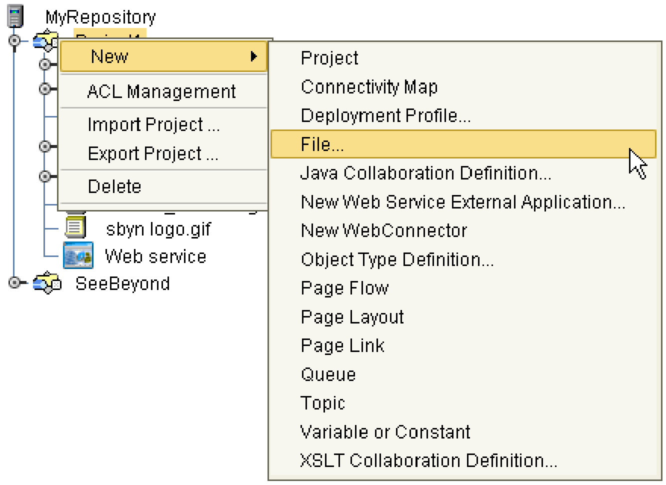
8.1.2 Importing Images

The sample Web page described in this section requires the use of a graphical image (a logo) as part of the page layout process. Before graphic images can be used in an eVision Web page, they must first be imported into the Repository. In this section, the page layout starts with importing a graphic logo image.

To import a graphic image

1. In **Enterprise Explorer** tree, right-click the **MyProject** icon.
2. On the context menu, point to **New** and then click **File**. See **Figure 52 on page 61**.

Figure 52 Importing an image into the Repository

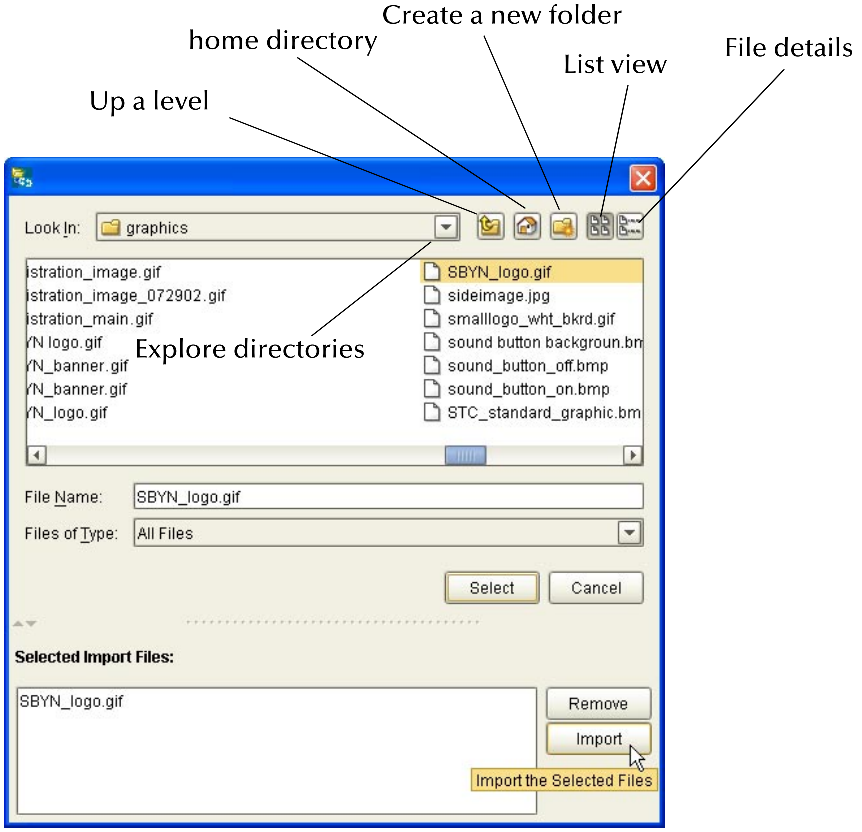


3 In the **Selected Import Files** dialog box, explore to the graphic image file that you want to add to the Repository. (To explore to the location, double-click the folders.) Externally-acquired graphic images can be in any directory that is accessible to your computer, locally or on a network. See Figure 53.

4 Select the image file, and then click **Select**.

5 Click **Import**.

Figure 53 Image source dialog box



istration_image.gif
istration_image_072902.gif
istration_main.gif
'N logo.gif
'N_banner.gif
'N_banner.gif
'N_logo.gif
SBYN_logo.gif
sideimage.jpg
smalllogo_wht_bkrd.gif
sound button backgroun.bm
sound_button_off.bmp
sound_button_on.bmp
STC_standard_graphic.bm

File Name: SBYN_logo.gif

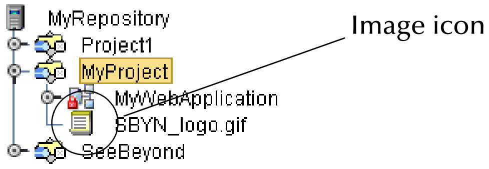
Files of Type: All Files

Selected Import Files:

SBYN_logo.gif

The graphic image's name will appear nested within your Project in the **Enterprise Explorer** tree. See Figure 54.

Figure 54 Graphic image in Enterprise Explorer

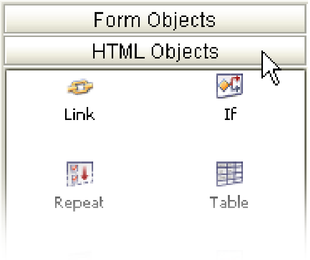


The image is now a part of your Project and can be imported into a graphic container on the Page Layout Designer canvas.

8.1.3 Placing GUI Components on the Canvas

This section shows you how to place basic components for the login page on the canvas using the Object selection palettes. Before you begin, make sure that the **HTML Objects** selection palette is in front. See **Figure 55 on page 63**.

Figure 55 HTML Objects palette in front



Create the Page Banner GUI Component

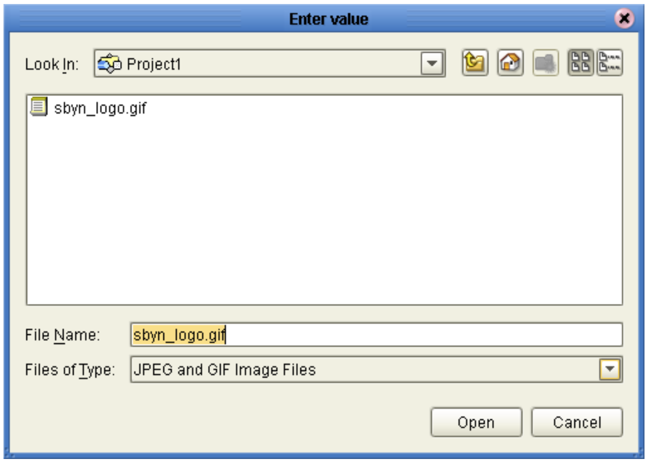
1 To create the **Page Banner** GUI component, from **HTML Objects** selection palette, drag the **Image** icon onto the canvas.

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

2 Drop the rectangle onto the top left of the canvas.

The rectangle will change to a blue box with a gold outline. This represents the outside boundaries of the graphic container. The **Enter value** dialog box appears. See Figure 56.

Figure 56 Selecting a graphic for the Image object



Look In: Project1

sbyn_logo.gif

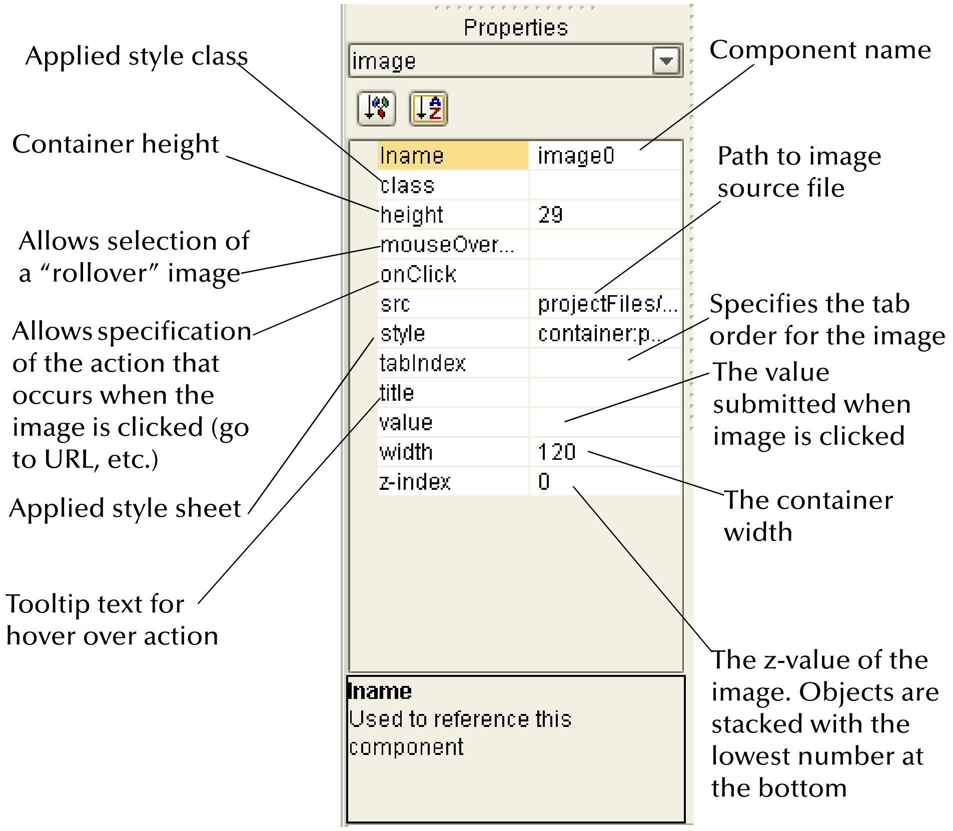
File Name: sbyn_logo.gif

Files of Type: JPEG and GIF Image Files

3 In the dialog box, select the image you want to place in the container and click **Open**.

The image is linked and displayed in the container. The **Component Properties** sheet for the **Image** component is automatically displayed. See **Figure 57 on page 64**.

Figure 57 Component Properties sheet for Image



Properties

image

lname	image0
class	
height	29
mouseOver...	
onClick	
src	projectFiles/...
style	container:p...
tabIndex	
title	
value	
width	120
z-index	0

lname
Used to reference this component

Applied style class
Container height
Allows selection of a “rollover” image
Allows specification of the action that occurs when the image is clicked (go to URL, etc.)
Applied style sheet
Tooltip text for hover over action
Component name
Path to image source file
Specifies the tab order for the image
The value submitted when image is clicked
The container width
The z-value of the image. Objects are stacked with the lowest number at the bottom

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

A In the left column, select the **lname** field (the default name will be **img(*n*)**).

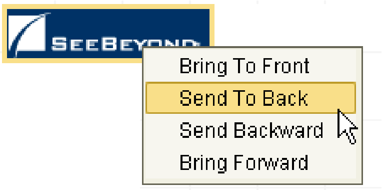
B In the value field, delete the default name, and type a name for the component (for example **image1**).

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

- To resize the container, click and drag a highlighted boundary element. The shape of the bidirectional arrow cursor will tell you which way the container will be resized.
- To move the container, click the container in the center (away from the edges), and drag it to the new location.
- To modify the z-value of the image, right click the image container, and in the context menu that appears, do the following:
  - To move the image to the top (front) of the stack, click **Bring to Front**.
  - To send the image to the bottom (back) of the stack, click **Sent to Back.**
  - To move the image down (back) one level, click **Send Backward**.
  - To move the image up (foward) on level, click **Bring Forward**.

See Figure 58.

Figure 58 Adjusting the z-value for an image



Create the Top Horizontal Line GUI Component

1 To create the **Horizontal Line** GUI component, from **HTML Objects** selection palette, drag the **Horizontal Line** icon onto the canvas.

2 Drop it just under the logo image.

The **Component Properties** sheet for the **Horizontal Line** component is automatically displayed.

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

A Select the **lname** field (the default name will be **hr(*n*)**).

B In the **lname** field, delete the default name and type a name for the line, for example, **hzline1**.

4 To adjust the line length:

A Click the right end of the line.

The cursor will change to a diagonal, bidirectional arrow.

B Drag the line end to the right to stretch the line across the canvas.

This line will be a visual separator displayed to users under the logo image.

Note: *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.*

Create the User Prompt GUI Component

1 To create the **User Prompt** GUI component, from **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

2 Place the component underneath, and left-aligned with the **Horizontal Line** component.

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

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

A Under the **Value** heading, select the **lname** field. (The default name will be **htmltext(n)**).

B In the **lname** field, delete the default name, and type: **UserPromptLabel**.

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

A Under the **Value** heading, click the **[default text]** field.

B Click the **Command** button (**...**) to launch the **Enter value for "text"** window.

C In the **Enter value for "text"** window, type: **Enter your username and password to begin**.

D Click **OK**.

This text string will be displayed to users prompting them to enter their user name and password to log into the Web application.

Initially, on the canvas, the text string will appear truncated. To see the entire text string, click and drag a bounding box around the text, then drag the right vertical bar of the bounding box to the right. To move the component, bound it again, then click inside the rectangle, and drag it to the desired location. This component should be left-aligned with the other components on the page.

Create the User Name Label GUI Component

1 To create the **User name** GUI component, from **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

2 Place it under, and left-aligned with the **User Prompt** component.

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

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

A In the left column, click the **lname** field. The default name will be **htmltext(*n*)**).

B In the **lname** value field, delete the default name, and type: **UserNameLabel**.

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

A Under the **Value** heading, click the **[default text]** field.

B Click the **Command** button (**...**) to launch the **Enter value for "text"** window. The **[default text]** string will be displayed in the window.

C In the **Enter value for "text"** window, delete the **[default text]** string, and type: **Username:**. Make sure you close the string with a colon (:).

5 Click **OK**.

This text string will be displayed to users and will label the **User ID** input field.

Create the Password Label GUI Component

1 To create the **Password Label** GUI component, from **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

2 Place the component under, and left-aligned with the **User Name** component.

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

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

A In the left column, click the **lname** field. The default name will be **htmltext(*n*)**).

B In the **lname** value field, delete the default name, and type **PasswordLabel**

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

A Under the **Value** heading, click the **[default text]** field.

B Click the **Command** button (**...**) to launch the **Enter value for "text"** window. The **[default text]** string will be displayed in the window.

C In the **Enter value for "text"** window, delete the **[default text]** string, and type: **Password:**. Make sure you close the string with a colon (:).

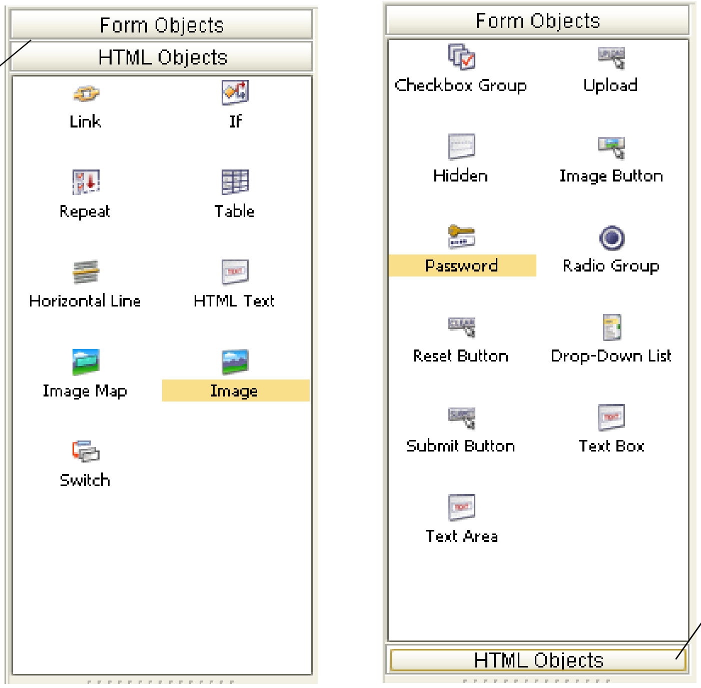
D Click **OK**.

This text string will be displayed to users and will label the **Password** input field.

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 59.

Figure 59 HTML and Form Objects palettes



Click to bring Form Objects to the front

Click to bring HTML Objects to the front

Create the User ID Text Field

1 To create a **User ID** GUI component, from the **Form Objects** selection palette, drag the **Text Box** icon onto the canvas.

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

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

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

A In the left column, click the **lname** field. (The default name will be **text(*n*)**).

B In the **lname** value field, delete the default name, and type: **UserName**.

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

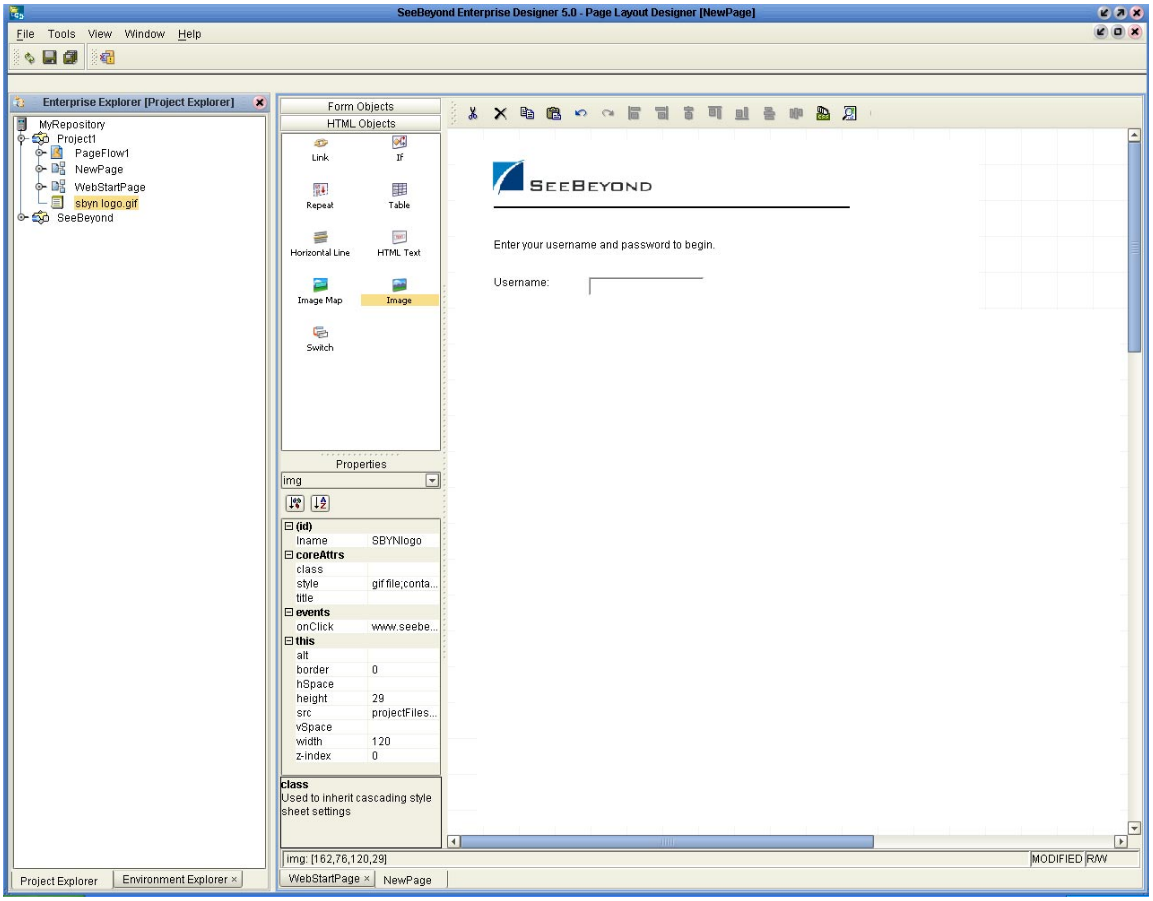
A Under the **Value** heading, click the **[default text]** field.

B Delete the **[default text]** string. (You want the **User ID** input field to be initially blank, to receive input from users.)

This component will be displayed to users as an empty input field with the label: **Username**

At this point, the unfinished canvas should look like the example shown in **Figure 60 on page 69**.

Figure 60 Unfinished Login Page canvas



Enter your username and password to begin.

Username:

img

(id)	
lname	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]

Create the Password Text Field

1 To create a **Password Text Box** GUI component, from the **Form Objects** selection palette, drag the **Password** icon onto the canvas.

Align this component horizontally with, and to the right of the **PasswordLabel** component. Make sure the components' vertical edges do not overlap.

The **Component Properties** sheet for the **Password** component is automatically displayed.

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

A Under the **Value** heading, click the **lname** field. (The default name will be **password(*n*)**).

B In the **lname** value field, delete the default name, and type: **Password**.

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

C Under the **Value** heading, click the **[default text]** field.

D Delete the **[default text]** string. (You want the **Password** input field to be initially blank, to receive input from users.).

This component will be displayed to users as an empty input field with the label: **Password**

Create the Submit Button

1 To create the **Submit Button** GUI component, from the **Form Objects** selection palette, drag the **Submit Button** icon onto the canvas.

2 Align this component below, and left-aligned with the **Password text field** component.

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

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

A Under the **Value** heading, click the **lname** field. (The default name will be **submit(*n*)**).

B In the **Value** field, type **SubmitLogin**.

4 In the **Component Properties** sheet, define the component's **value** by doing the following:

A Under the **Value** heading, select the **value** field.

B In the **value** field, type: **Login**.

This component will be displayed to users as a button with the label: **Login**

Swap the HTML Objects and Form Objects Palettes

To bring the **HTML Objects** palette to the front, click the **HTML Objects** title bar.

Add a Horizontal Line to the Bottom of the Page

1 To create the bottom **Horizontal Line** GUI component, click the horizontal line at the top of the canvas.

2 On the Page Designer toolbar, click the **Copy Components** icon.

The line is copied to the clipboard.

3 To place the copy on the canvas, click the **Paste Components** icon.

The copy will appear just below the parent horizontal line.

The **Component Properties** sheet for the **Horizontal Line** component is automatically displayed.

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

A Under the **Value** heading, click the **lname** field (the default name will be **hr(*n*)**).

B In the **lname** field, delete the default name and type a name for the line, for example, **hzline2**.

Do not adjust the line length.

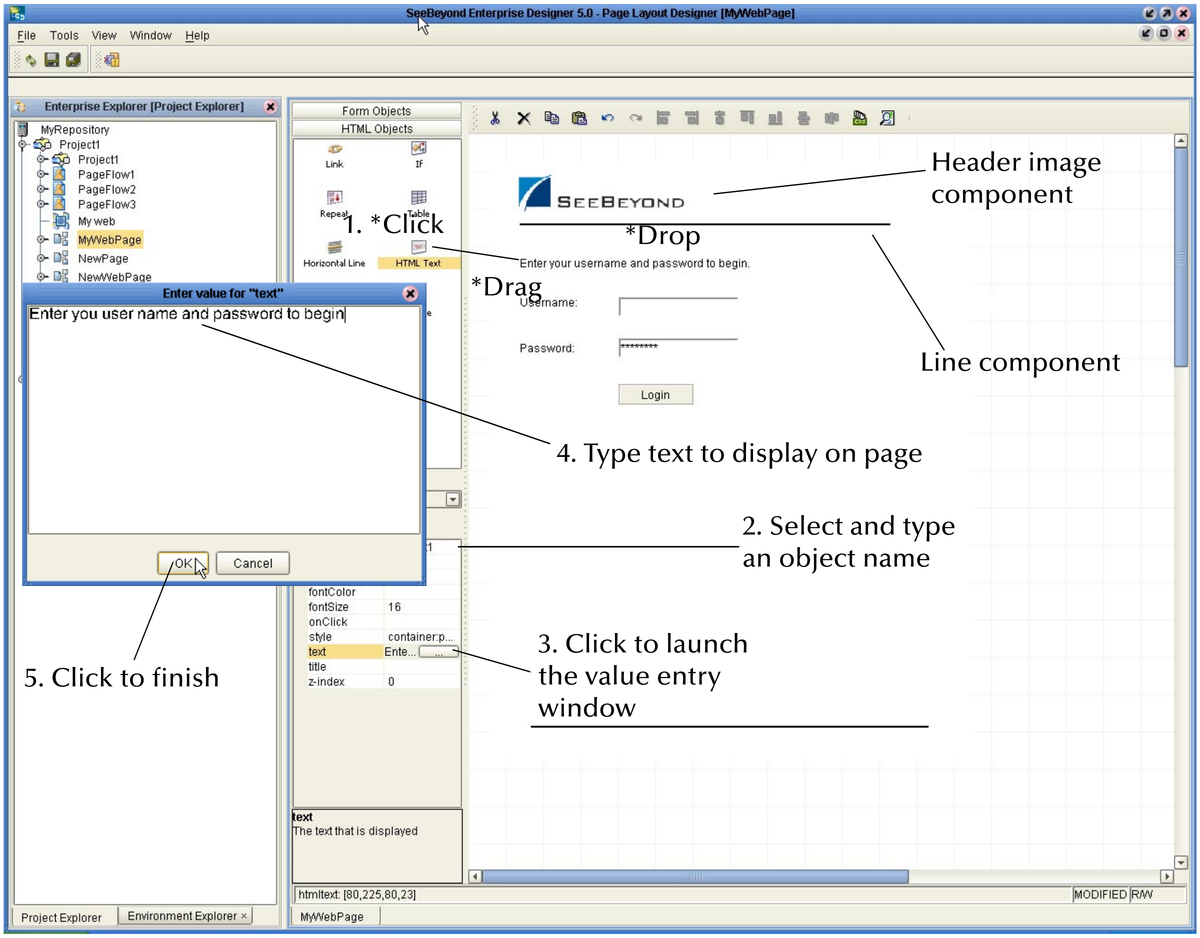
5 Move the line to the lower half of the page, well below the **Submit** (Login) button.

To move the line, click the line's mid-point (avoid the end points), and drag it to a new location. This line will be a visual separator displayed to users near the bottom of the page.

8.1.4 Reviewing the Page Layout

This completes the page construction activity. For a brief, visual summary of the process, see Figure 61.

Figure 61 Review adding GUI components to the canvas



Enter you user name and password to begin

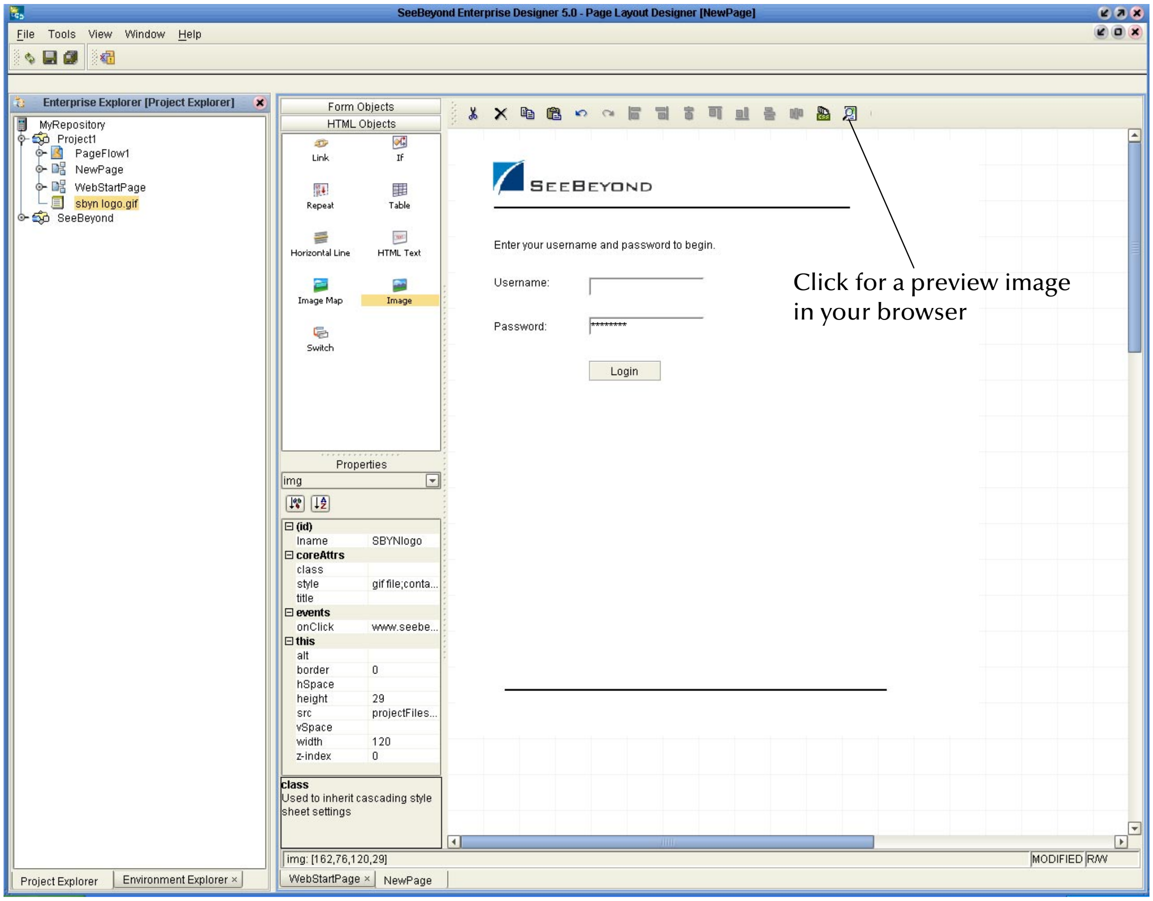
fontColor	
fontSize	16
onClick	
style	container:p...
text	Ente...
title	
z-index	0

text
The text that is displayed

htmltext: [80,225,80,23] MODIFIED R/W

On the design canvas, the finished **Login** page should look like the example shown in **Figure 62 on page 72**.

Figure 62 Finished Login page



⊟ (id)	
lname	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]

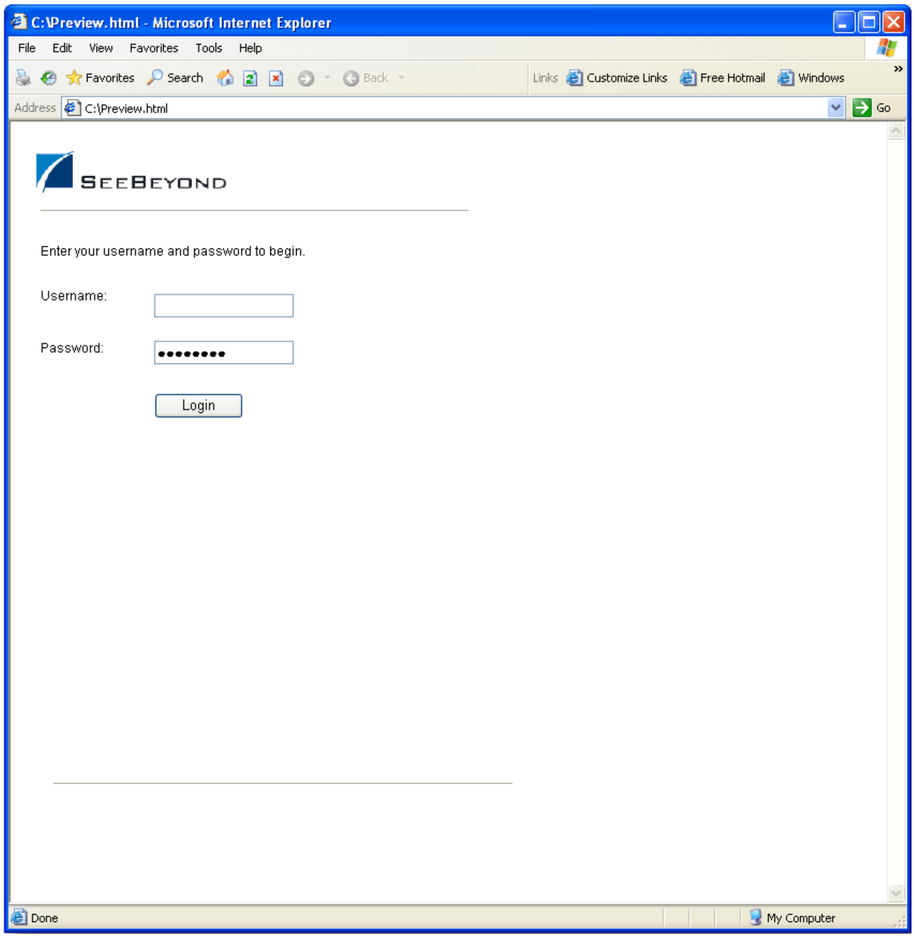
Click for a preview image in your browser

The arrangement of objects on the screen may not be what you want them to be. If so, you can move any object to any location on the canvas. See **"Manipulating Objects on the Canvas" on page 73**.

8.1.5 Previewing the Finished Web Page

To preview the finished **Login** page in your browser, on the **Page Layout Designer** toolbar, click the **Preview** icon. The page will display in a new browser window. It should look like the example shown in **Figure 63 on page 73**.

Figure 63 Web page preview


C:\Preview.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Favorites Search Back
Links Customize Links Free Hotmail Windows
Address C:\Preview.html Go
SEEBEYOND
Login
Done
My Computer

Enter your username and password to begin.

Username:

Password: •••••••••

8.1.6 Saving the Project in the Repository

To save the page layout to the Repository, on the **Enterprise Designer** menu, select **File,** and click **Save All**. The system will save the Project in the Repository.

Note: *Before you log out, the Project must be checked into the Enterprise Designer Version Control system. See* **"Checking the Page Layout in to Version Control" on page 75***.*

8.2 Manipulating Objects on the Canvas

On the preview page, some of the components may not be located exactly where you want them, or they may not be aligned evenly. For example, if the horizontal lines are not even in length or alignment, it detracts from the composition of the page layout. This section describes the various ways you can manually move and resize page components.

8.2.1 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.

8.2.2 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 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.

8.2.3 Aligning Objects Using the Alignment Tools

Aligning objects requires the involvement of two or more objects.

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

- Select the canvas in a blank area adjacent to the objects you want to align, and drag the cursor over the objects. Your mouse movement will trace a bounding box for visual reference. You can surround the objects you want to align, 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 select the objects you want to align.

When the objects you want to align are selected, on the **Page Layout Designe**r toolbar, click the appropriate alignment tool. 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.

Tool	Name	Purpose
	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.
	Horizontal Spacing	Creates equal horizontal spacing between objects in a group based on an averaging algorithm.
	Vertical Spacing	Creates equal vertical spacing between object in a group based on an averaging algorithm.

8.2.4 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 74**, and **"Moving a Temporary Group of Objects" on page 74**. 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.

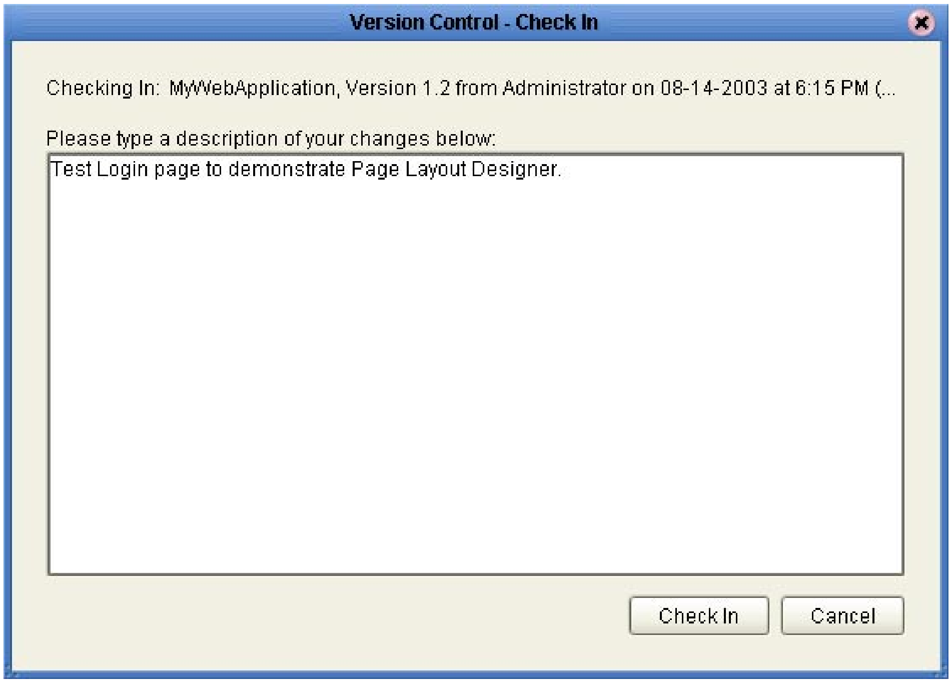
8.2.5 Checking the Page Layout in to Version Control

All new Projects must be checked in to the Enterprise Designer Version Control system. This system allows you to track changes and prohibit simultaneous check-out of a Page Layout (when a page is checked out, it is locked and unavailable to other developers).

To check in the Page Layout

1. In the **Enterprise Explorer** tree, right-click the **Page Layout** name (for example, **MyWebApplication**), and on the context menu, click **Check In**.
2. In the **Version Contro**l dialog box, type a comment to accompany the Page Layout. See Figure 64.

Figure 64 Version Control check in procedure



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

Please type a description of your changes below:

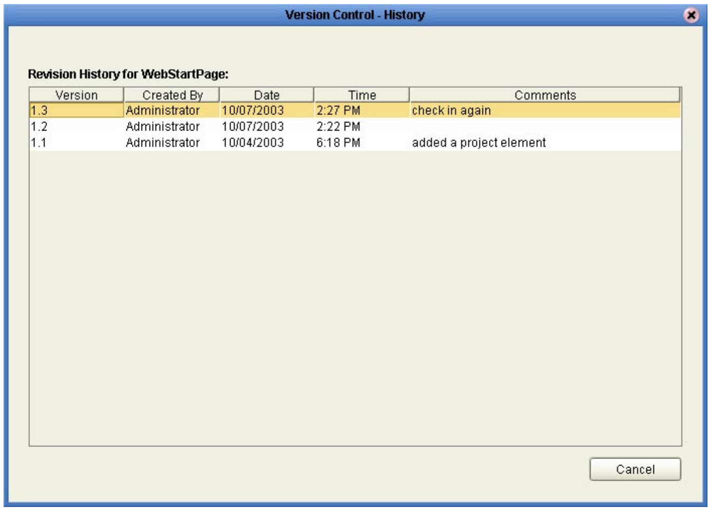
Test Login page to demonstrate Page Layout Designer.

3. Click **Check In**.

To check out a Page Layout

1. In the **Enterprise Explorer**, right-click the name of the Page you want to checkout.
2. To review version history of the Page, on the context menu, click **Version History**. See **Figure 65 on page 77**.

Figure 65 Version History



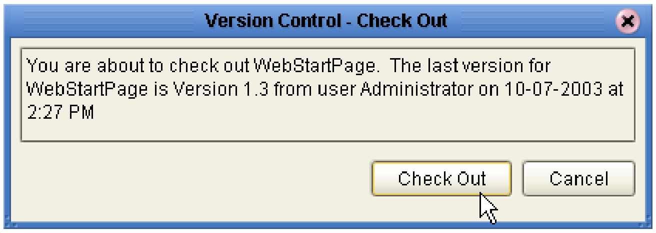
Revision History for WebStartPage:

Version	Created By	Date	Time	Comments
1.3	Administrator	10/07/2003	2:27 PM	check in again
1.2	Administrator	10/07/2003	2:22 PM	
1.1	Administrator	10/04/2003	6:18 PM	added a project element

3 Close the **Version Control- History** dialog box.

4 On the context menu, click **Check Out**. Figure 66.

Figure 66 Check out a Version



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

5 In the **Version Control - Check Out** dialog box, click **Check Out**.

Chapter 9

Page Flow Designer

To build a functioning Web application, you first develop the individual user-facing pages that you want users to interact with. When the page-creation task is finished, Vision's Page Flow Designer allows you to link the Web pages together and create a logical chain called a Page Flow.

This Chapter Includes:

9.1 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.

A sample Page Flow is shown in Figure 67.

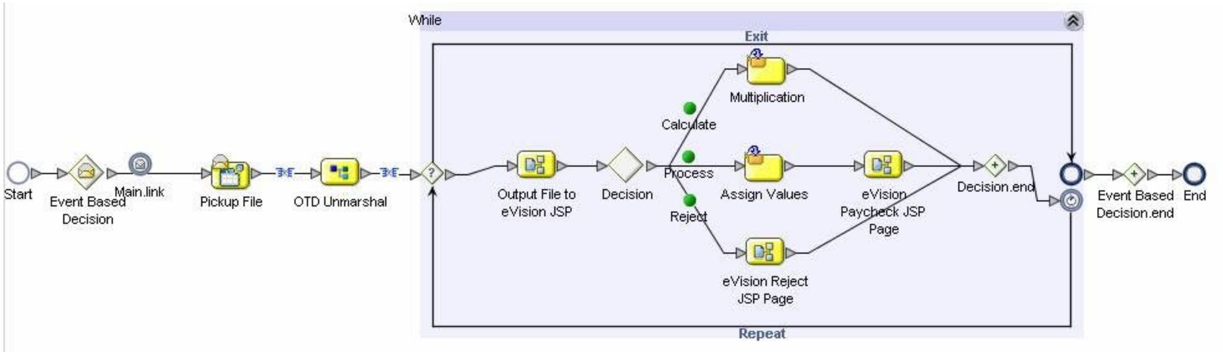


Figure 67 Sample Page Flow

Adding a Page Flow to a Project

When you add a Page Flow to a Project, Page Flow Designer opens an empty Page Flow canvas for placing pages and other design (sub-process) elements and then connecting them together in a sequence. Before you can start building a Page Flow, you must first add the Page Flow to your Project.

1. From the Enterprise Explorer pane, right-click **Project**.
2. From the Context menu, point to **New** and then click **Page Flow**.
3. In the Explorer tree, type a name for the project.

9.1.1. Adding Web Pages to a Page Flow

To create a Page Flow, you drag and drop Web pages and other sub-process elements onto the Page Flow Designer canvas, and then link the pages with other elements to create a logical page flow for a Web application. Page Flow Designer provides the tools to lay out and connect the Web pages and other specialized sub-process elements, including graphic editing tools that allow you to adjust, resize, and align Web page and sub-process components.

Creating a Page Flow

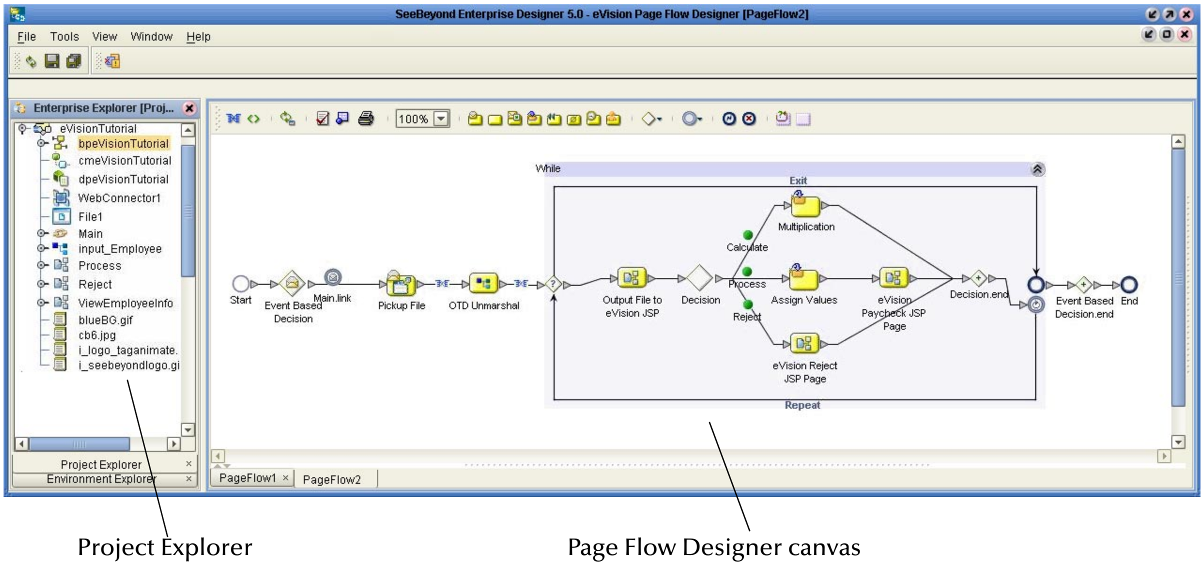
When you start a new Web application project, the **Start** and **End** Page Flow icons automatically appear on the blank Page Flow Designer canvas. (There can be only one starting point for a Page Flow, however, there can be multiple end points.)

1. Drag the desired Web pages onto the Page Flow Designer canvas.
2. Create links between the pages (and other elements as necessary) to create the Page Flow.
3. Click **Save** to save your changes to the SeeBeyond 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.

See **Figure 68 on page 80**.

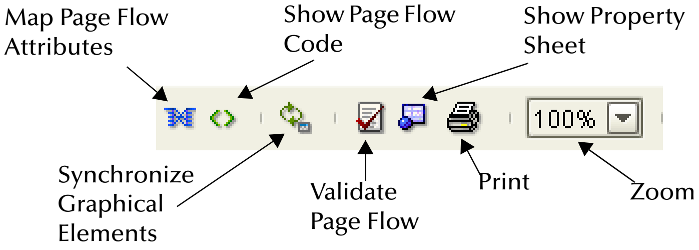
Figure 68 Page Flow in the Page Flow Designer GUI



9.2 Using the Page Flow Designer GUI

The Page Flow Designer toolbar is shown in Figure 69.

Figure 69 Page Flow Designer Toolbar



The Page Flow Designer tools are described in Table 6.

Table 6 Page Flow Designer GUI tools

Tool	Purpose	Description
	Map Page Flow Attributes	Shows that rule mapping exists between two Page Flow entities.
<>	**Show Page Flow Code**	Opens a window in the lower portion of the Page Flow Designer screen and displays the generated code for the Page Flow.

Tool	Purpose	Description
⟳	**Synchronize Graphical Page and Page Flow Code**	Synchronizes the Page Flow on the canvas with the underlying code generation.
☑	**Validate Page Flow**	Check 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 to decreases the Page Flow image on the screen.

9.3 Page Flow Elements

Page Flow Designer provides a palette of Page Flow Design 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 sub-process elements, branching elements, and intermediate events.

Sub-process Elements

There are several different kinds of sub-processes you can include in a Page Flow. See Table 7.

Table 7 Connecting and Sub-process Elements

Button	Command	Function
○	Start Node	The Start Node element 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 only link to a **Page Link**.
▷—▷	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. Page Flow Designer ensures the Page Flow is correctly linked by rejecting invalid links. Links can also accept mapped values. A link with mapped values is highlighted in blue.
○	End Node	The End Node indicates the completed state of a Page Flow. The End node is automatically added to the canvas when new Page Flow is initialized.

Table 7 Connecting and Sub-process Elements

Button	Command	Function
	Receive	The Receive element is the only element that a **Page Link** can connect to. 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 their 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 upon 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.
	Reply	The Reply Activity 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 activity 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 elements 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 activity allows data to pass through without changes.
	Wait	The Wait activity acts as a timer. The user will build a Page Flow in which there are two simultaneous paths within a set scope, one for the business process and 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.

Branching Elements

Branching Activities are objects you add to your Page Flows to specify the logical flow of information. eVision provides three different kinds of branching activities—Decisions, Event Based Decisions, and Flow. See Table 8.

Adding a Branching activity

To add a Branching activity to the Page Flow Designer canvas

1. On the Page Flow Designer toolbar, click Branching Activity, and the select the type of Branching activity you would like to add.
2. Click your choice and drag it to the Business Page Flow Designer canvas.

The selected Branching activity appears on the Page Flow Designer canvas.

Branching elements are described inTable 8.

.

Table 8 Branching Elements

◇	Decision	A Decision 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. Decisions allow you to define expressions that are evaluated to determine the proper Page Flow flow. Expressions are built using the mapping interface and Page Flow Attributes. ***Note:*** *This 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	Multiple possible messages can be juxtaposed against a time-out condition to allow the type of message received to determine the appropriate Page Flow path.
◈	Flow	Specifies that one or more pages are to flow concurrently.

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.

Adding an Intermediate event

To add an **Intermediate event** to the Page Flow Designer canvas:

1. Click on the **Intermediate event** toolbar icon and select the type of **Intermediate event** you would like to add.

Click on your choice and drag it from the menu to the Page Flow Designer canvas.

Intermediate Events are described in Table 9.

Table 9 Intermediate Events

	Timer Event	A time-out condition is set for 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.
	Message Event	This 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.
	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 things go wrong and the exception handler will automatically initiate the appropriate process to handle the problem.
	Catch All Exceptions	This exception handler is configured to handle all exceptions.
	Compensation Handler	Used when something in a Page Flow fails and requires a rollback or upstream activities (like money has to be returned to the customer). 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.
	Throw	Throws an exception.
	Terminate	Ends the Page Flow.

Adding an Intermediate event

To add an **Intermediate event** to the Page Flow Designer canvas:

1 On the Page Flow Designer toolbar, click the **Intermediate event** icon, and then select the type of **Intermediate event** you would like to add.

Click your choice and drag it from the menu to the Page Flow Designer elements.

While

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

The While looop is described in **Table 10 on page 85**.

Table 10 While loop

↻	While	Creates and maintains a looping process within a Page Flow. For example, a negotiation may consume several weeks, but management wants to review the status daily. The loop continues until the negotiation is complete, and then the Page Flow continues.

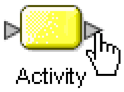
9.3.1. Links

In a Page Flow, you can connect Web pages using either orthogonal and diagonal link styles – this setting applies to all links in a Page Flow and is an automated application of the style.

To link Web pages and other Page Flow elements, do the following:

1. Move your cursor over the connector portion of the page element.
2. Hold the cursor over the outside edge of the page until it changes from the arrow pointer to a hand. See Figure 70 and Figure 71.

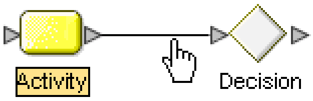
Figure 70 Starting a Link



3. Drag a line from the first Web page to the connector of the second Web page, and then release the mouse.

Links are used to create the flow between Pages. See Figure 71. You can also add logic to a link, by adding an Assign.

Figure 71 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.*

9.3.2. 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, on the Page Flow Designer toolbar, click **Validate Page Flow.**

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**.**

9.3.3. 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**.

9.4 Configuring Page Flow Designer Elements

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

9.4.1. Assigning an Activity

Adding an Inline Assign

1. Right click on a link that you have created.
2. Select **Add Assign**.
3. On the Page Flow Designer toolbar, the **Map Page Flow** Attributes icon .

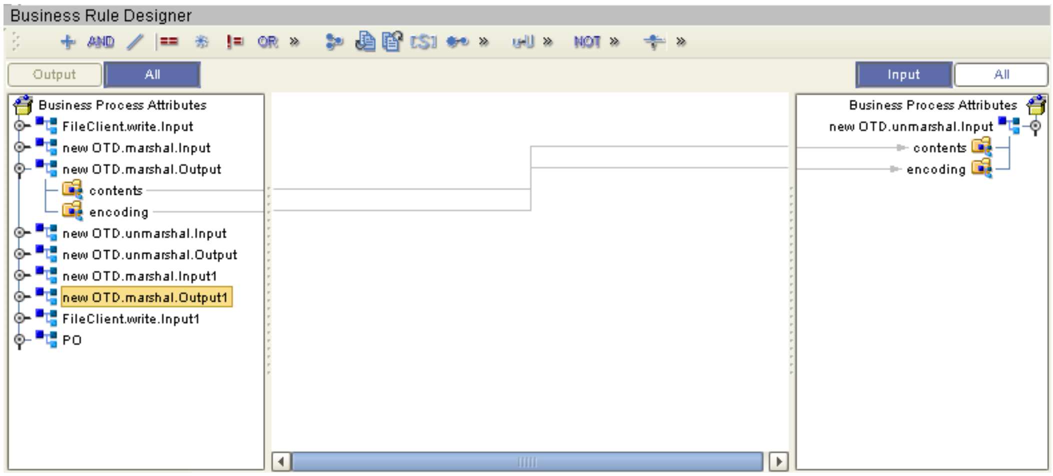
   The **Business Rules Designer** appears in the lower part of the Page Flow Designer.

9.4.2. About Business Rules Designer

The **Business Rules 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 Rules Designer** is called the *Mappe*r. See Figure 72.

The Business Rules Designer appears whenever you click **Map Page Flow Attributes** icon (shown in Figure 72). Also, it is automatically activated when you add or select an inline Assign, and add or select an Assign activity.

Figure 72 Business Rules Designer



Method Palette

Use the Method Palette in the Business Rules Designer to configure data passed between input and output Pages. You can drag and drop a method from the method palette to the Business Rules Designer and then configure the method.

See **"Method Palette" on page 90** for more information about the methods in the Method Palette.

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.

To use a sub-process, do the following:

1. Create the parent process – add an empty Activity in as a placeholder for the sub-process.
2. Create the child process – add an empty Receive at the beginning and reply at the end.
3. Create a WSDL file that will represent the inputs and outputs of the process.
4. In the child sub-process:
   - A Add the WSDL to the Page Flow using the WSDL tab.
   - B Create a partner using the WSDL tab.
   - C Open the properties of the initial Receive and select the appropriate partner, portType, and operation.
   - D 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.*

5 Generate the code and save the child process.

6 Return to the parent process – from the Project Explorer, drag the 1st operation under the child process to the sub-process placeholder in the parent process.

Note: *It is not always the 1st operation - it is the operation assigned to the Receive/Reply in the sub-process.*

7 Perform any appropriate mappings using Assign Activities in the parent Page Flow.

8 Deploy both processes.

Correlation Keys and Sets

Configuring Correlation consists of the following steps:

- **"Create a Correlation Key" on page 88**

  A *correlation key* is a value that you can assign to a Page Flow, like 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.

- **"Add Correlation Sets" on page 90**

  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 wish to assign a Purchase Order number and an invoice number to a transaction, so that all information about the purchase and payment are associated.

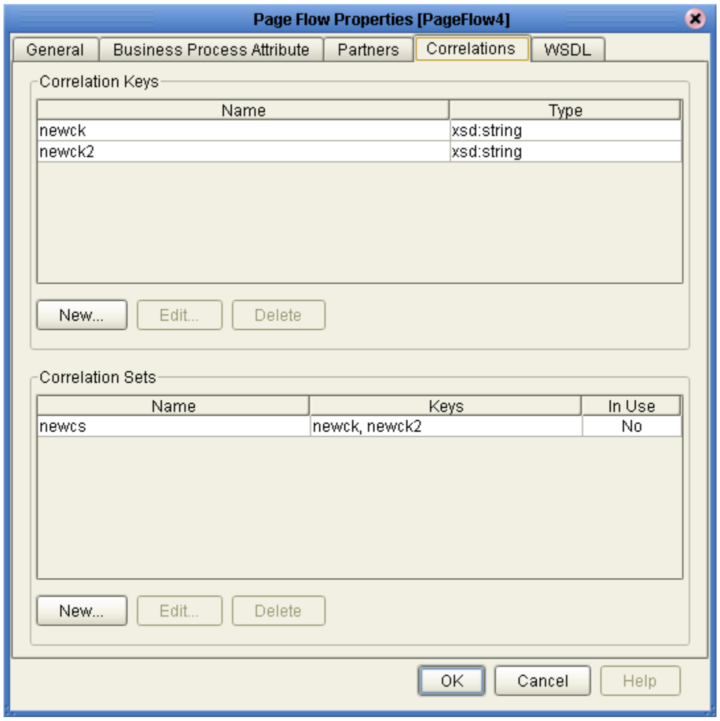
- **"Bind Sets of Correlations to Activities" on page 90**

  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).

Create a Correlation Key

1 Select the **Correlations** tab (see Figure 73).

Figure 73 Page Flow Properties: Correlations Tab



Correlation Keys

Name	Type
newck	xsd:string
newck2	xsd:string

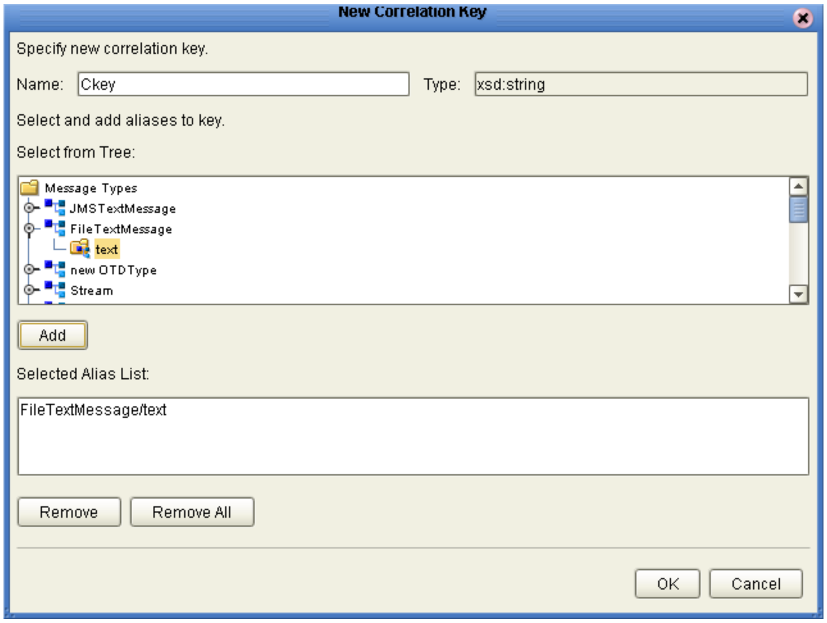
Correlation Sets

Name	Keys	In Use
newcs	newck, newck2	No

2 Select **New** from the **Correlation Keys** section of the dialog box.

The **New Correlation Key** dialog appears as shown in Figure 74.

Figure 74 New Correlation Key Dialog Box



Specify new correlation key.

Name: Ckey Type: xsd:string

Select and add aliases to key.

Select from Tree:

Selected Alias List:

FileTextMessage/text

3 From the **New Correlation Key** dialog box:

E Enter a **Name** (alias) for the Correlation Key

F 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.

4 Click Add to save the new alias to the **Selected Alias List**

5 Click **OK** to save your changes and exit the **New Correlation Key** dialog box.

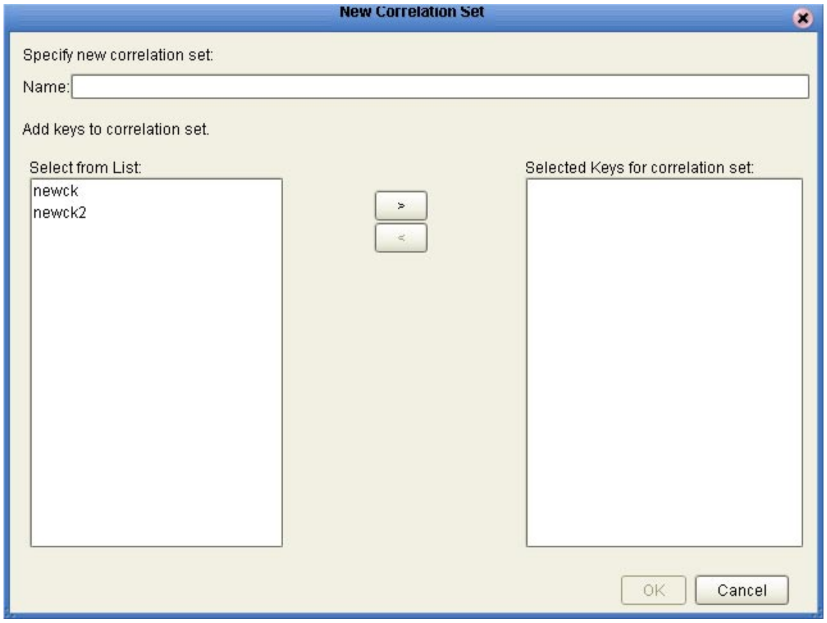
Add Correlation Sets

1 Select the **Correlations** tab (see Figure 73).

2 Select **New** from the **Correlation Set** section of the dialog box.

The **New Correlation Set** dialog appears as shown in Figure 75.

Figure 75 New Correlation Set Dialog Box


New Correlation Set
Specify new correlation set:
Name:
Add keys to correlation set.
Select from List:
newck
newck2
>
<
Selected Keys for correlation set:
OK
Cancel

3 From the New Correlation Set dialog box:

A Enter a **Name** for the new Correlation Set.

B Select **Correlation Keys** from the list to add to the Correlation Set.

C Click the arrow button to move your selections to the Correlation Set.

4 Click **OK** to save your changes and exit.

Bind Sets of Correlations to Activities

1 Select an Activity.

2 Select Show Property Sheet from the toolbar.

Locate **Use Correlations** and select **Yes.**

9.5 Page Flow Properties

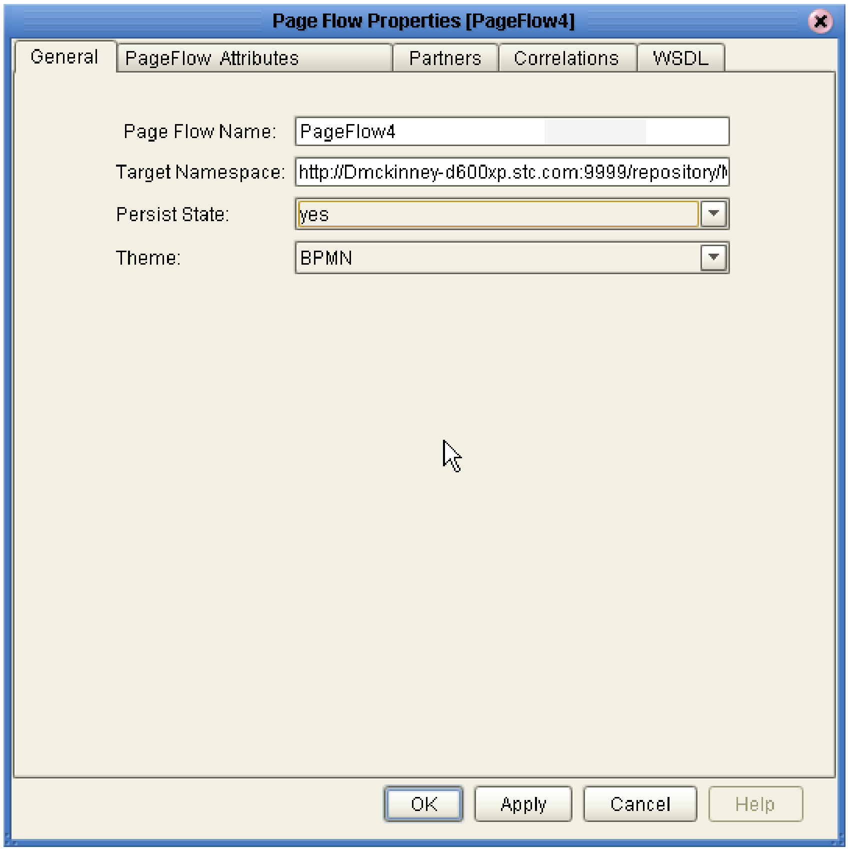
Each Page Flow has a set of properties, which enable the rapid creation and removal of Page Flow attributes. eVision uses this information to automatically create the appropriate Page Flow attributes and input/output structures, for use in the Business Rules Designer.

To edit Page Flow properties:

1. Right-click the **Page Flow** you want to edit.
2. Click **Open Business Property Dialog**.

   The **Page Flow Properties** dialog appears. See **Figure 76 on page 91**.

Figure 76 Page Flow Properties Dialog: General Tab



Page Flow Properties [PageFlow4]

Page Flow Name: PageFlow4
Target Namespace: http://Dmckinney-d600xp.stc.com:9999/repository/N
Persist State: yes
Theme: BPMN

General

The **General** tab is the first tab you see when you begin to edit a Page Flow property. You can change the Page Flow name, edit the target namespace and select the Persistence State from this tab.

Edit General Page Flow Properties

1. From the **General** tab, you can edit:

- **Page Flow Name**
- **Target Namespace** - address of the Page Flow.
- **Persist State** - See **"Persistence" on page 63** for more information.

2 Click **OK** to Save your changes and exit the **Page Flow Properties** dialog box.

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 OTDs and Collaborations are represented automatically in the Enterprise Explorer and are available for use in your 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.

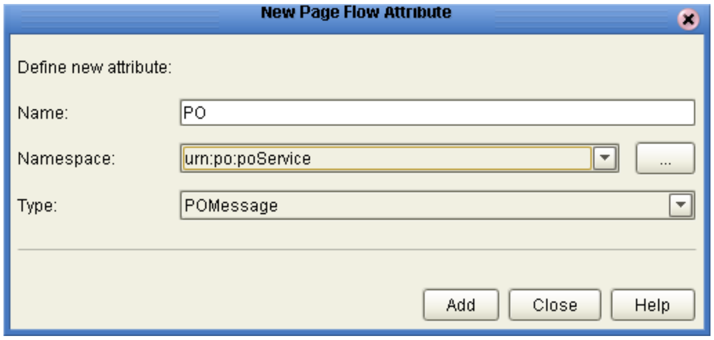
Create New Page Flow Attribute

1 Select the **Page Flow Attributes** tab. See Figure 78.

2 Select **New** to add a **New Page Flow Attribute**.

The **New Page Flow Attribute** dialog box appears. See Figure 77.

Figure 77 New Page Flow Attribute



New Page Flow Attribute

Define new attribute:

Name: PO

Namespace: urn:po:poService

Type: POMessage

3 From this dialog box, complete the following information:

- Enter a **Name** for the attribute.

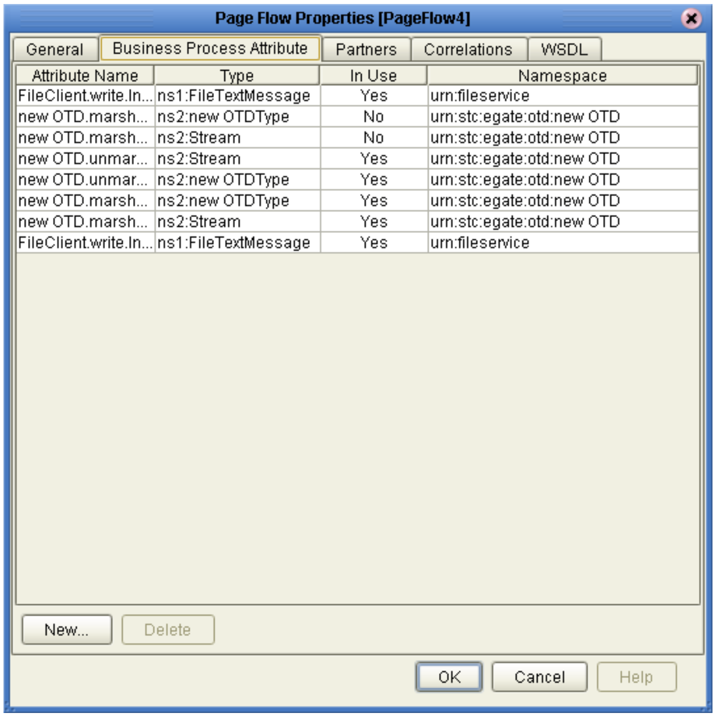
- Select or browse for an existing **Namespace**.
- Select an available **Type** for your attribute.

4 Click **Add** to Save the attribute or **Close** to return to the **Page Flow Properties** dialog box.

Edit a Page Flow Attribute

1 Select the **Page Flow Attributes** tab. See Figure 78.

Figure 78 Page Flow Properties: Page Flow Attributes Tab



Attribute Name	Type	In Use	Namespace
FileClient.write.In...	ns1:FileTextMessage	Yes	urn:fileservice
new OTD.marsh...	ns2:new OTDType	No	urn:stc:egate:otd:new OTD
new OTD.marsh...	ns2:Stream	No	urn:stc:egate:otd:new OTD
new OTD.unmar...	ns2:Stream	Yes	urn:stc:egate:otd:new OTD
new OTD.unmar...	ns2:new OTDType	Yes	urn:stc:egate:otd:new OTD
new OTD.marsh...	ns2:new OTDType	Yes	urn:stc:egate:otd:new OTD
new OTD.marsh...	ns2:Stream	Yes	urn:stc:egate:otd:new OTD
FileClient.write.In...	ns1:FileTextMessage	Yes	urn:fileservice

2 Select an existing attribute, and do one of the following:

- Select and double click the attribute name to rename it. (Some attributes cannot be renamed.)
- Click **Delete** to remove the attribute.

3 Click **OK** to save your changes and exit the **Page Flow Properties** dialog box.

9.5.1. WSDL Files

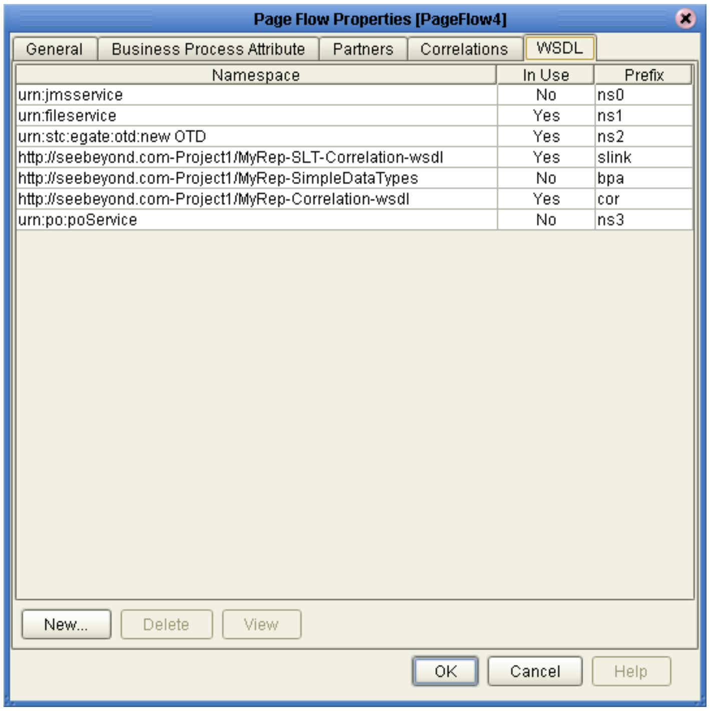
WSDL files are used to invoke and operate Web services on the Internet and to access and invoke remote applications and databases.

The WSDL tab is available from the Page Flow Properties dialog. From this dialog, you can upload a WSDL file with predefined Page Flow Attributes for use in your Page Flow.

To upload a WSDL file, do the following:

1 Select the **WSDL** tab (see Figure 79).

Figure 79 Page Flow Properties: WSDL Tab

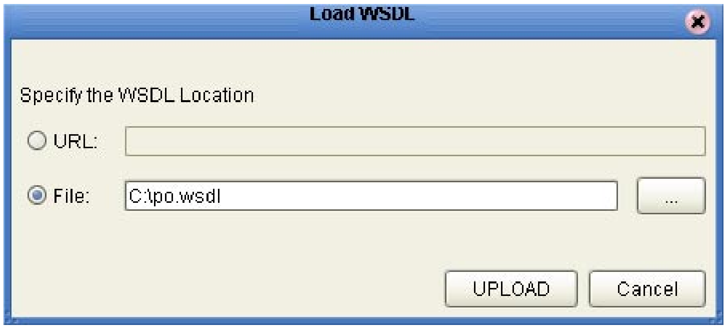


Namespace	In Use	Prefix
urn:jmsservice	No	ns0
urn:fileservice	Yes	ns1
urn:stc:egate:otd:new OTD	Yes	ns2
http://seebeyond.com-Project1/MyRep-SLT-Correlation-wsdl	Yes	slink
http://seebeyond.com-Project1/MyRep-SimpleDataTypes	No	bpa
http://seebeyond.com-Project1/MyRep-Correlation-wsdl	Yes	cor
urn:po:poService	No	ns3

2 Click **New** to upload a WSDL file.

The **Load WSDL** dialog box appears. See Figure 80.

Figure 80 Load WSDL dialog box



Specify the WSDL Location

○ URL:

● File: C:\po.wsdl

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

A Select **URL** or **File**, to choose where your WSDL file is located.

B Enter the path to the WSDL file.

You can also use the browse button to locate a File location.

4 Click **UPLOAD**.

9.6 Event Linking and Sequencing

This section describes how Event Linking and Sequencing (ELS) is implemented in this version of eVision. In some situations, you want to impose conditions on a set of Events, or process a group of Events together, or make a decision contingent on the receipt or non-receipt of all Events of a certain type. These situations can be handled using ELS.

You use ELS when you want to sort Events into separate labeled "buckets" and execute your business rules on a bucket-by-bucket basis rather than an Event-by-Event basis. The bucket label is called a Link Identifier, or Link ID; Events in the same bucket are linked.

Buckets are stored in memory, to facilitate processing each one as a unit. BPEL refers to these buckets as containers. As eVision retrieves a message or an event, it correlates the received message against a Page Flow instance. If found, eVision stores the message or event in the container for that Page Flow. Otherwise, it will instantiate a new instance of that Page Flow.

For example, you might have a Page Flow that deals with HL7 messages that have been broken up using a continuation pointer, and you have added logic that detects this condition and defers processing until the HL7 has been reassembled. In this case, the virtual bucket would be only be "full"—that is, all linking complete—when all HL7 messages for the same continuation pointer had been received.

9.6.1. ELS within eVision

eVision is invoked by messages or events sent to it via an eGate connector node such as an eWay or JMS. OnAlarm is used to set the expiration time. If time expires, then the loop condition is set to false to terminate the loop. If a message is received, then message counter is incremented and if the maximum number of messages have been received, then the loop is terminated. At the end, the date is written to a file.

Chapter 10

Binding and Deploying eVision Applications

When you have finished developing the Web pages that comprise a complete application, and have connected them, creating a Page Flow, you bind the pages to a Connectivity Map and deploy the pages to an Environment via Enterprise Designer and Environment Explorer.

When a Web page is saved to the Repository, the page appears in **Enterprise Explorer** as a **Show** icon. To create an activity from the Web page, you drag the **Show** icon to the **Page Flow Designer** canvas. You then manually connect the page with other pages using links, which represent the Page Flow and internal/external system interactions.

At run-time, the Page Flow mapping function populates Web page with data values and captures user inputs as outputs from the page. User interaction drives the display of the pages, while the pages send user responses to the backend systems.

This Chapter Includes:

10.1 Binding and Deployment

Binding and *Deployment* are procedures performed in Enterprise Designer using the Environment Explorer. These procedures are described in **"Binding an eVision Application to a Page Flow"** and **"Deploying an eVision Application" on page 97**.

10.1.1 Binding an eVision Application to a Page Flow

The steps to bind an **eVision** application to a Page Flow are as follows:

In Enterprise Designer:

1. Add the Page Flow instance to the **Connectivity Map Editor** (**CME**).
2. Drag the **eVision Web Connector** to the **CME**.
3. Connect the **eVision Web Connector** to the appropriate step in the Page Flow.
4. Double-click the connection to the **eVision Web Connector** to start the **eVision Configuration Editor**.

5 In the **Configuration Editor** dialog box, type the URL that will enable initial access to the page flow.

10.1.2 Deploying an eVision Application

The steps to deploy an **eVision** application to an **Environment** are as follows:

In Environment Explorer:

1 Create a new **Environment,** if necessary.

2 Add the following to the Environment:

- Integration Server
- **eVision Host**
- **Message Serve**r (if necessary)
- **External System** (if necessary)

3 Create a **Deployment Profile**.

4 Drag the **eVision Web Containe**r to the **eVision Host**.

10.2 Deployed Web Applications

After deployment, Web applications can execute assigned tasks, interact with backend systems, and populate a Worklist Viewer to allow users to view assigned tasks.

10.2.1 About Event-based Linking

A **Page Flow** can be encapsulated within a *while* loop, enabling *event-based linking*. Event-based linking allows pages to be branched to alternate execution paths within the application, when a message-based event is received by a page. In the looping **Page Flow** example shown in Figure 81, the **Page Flow** elements employed are **Web Start Page**, **Decision**, **While Loop**, **Link**, and **Event-based Decision**. For details on these elements, see **Table 11 on page 98**.

Figure 81 Simple event-based linking

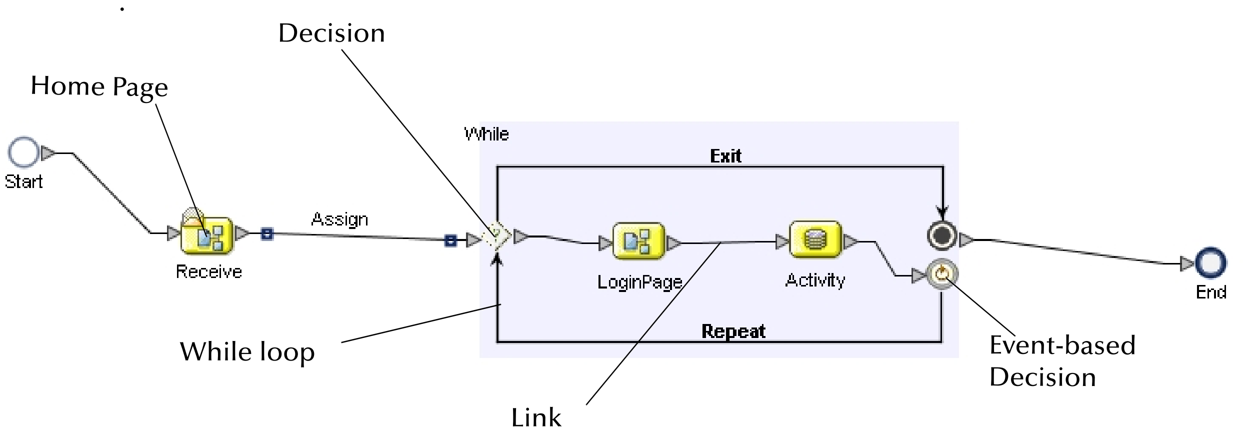


Table 11 Page Flow design elements

Element Name	Function
Dynamic Home page	The **Dynamic Home page** is the eVision equivalent of a Home or Index page, which contains the formatting instructions and JSP code for subsequent pages.
Decision	The **Decision** element allows one of several possible paths to execute, based on expression logic. This element is used to create expressions that determine the path of the Page Flow. It also contains the expression and connection names. Decision elements allow you to define expressions that are evaluated to determine the direction of the Page Flow.
While Loop	The **While Loop** element is a logical loop that allows you to create a continual looping process within a Page Flow.
Link	**Link** elements indicate the direction of the Page Flow by establishing directional connections between pages. When you select a link, a context menu allows you to configure how data is going to be passed to and from the underlying Web Service.
Event-based Decision	The **Event-based Decision** element invokes a link to another page or location (when there are two or more alternative possibilities) based on information that is passed to it from the previous page.

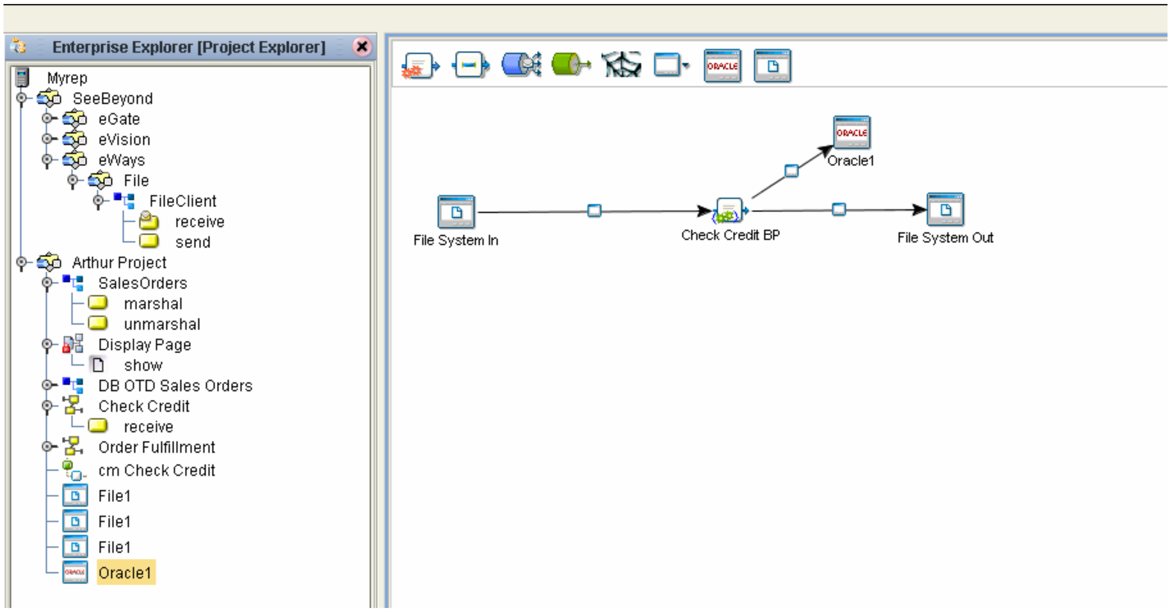
10.3 Page Flows in the Connectivity Map

The Connectivity Map represents Connectivity information. eVision uses the information in the Connectivity Map to establish and maintain connections to systems for the correct step in a Page Flow.

Creating a Connectivity Map that Includes a Page Flow

1. Drag the processing element to the Connectivity Map Editor
2. Drag the desired Web application from Enterprise Explorer to the icon
3. The “gears” in the icon will turn green to indicate the association
4. Add the external systems and components to the CME

Figure 82 Connectivity Map with Page Flow

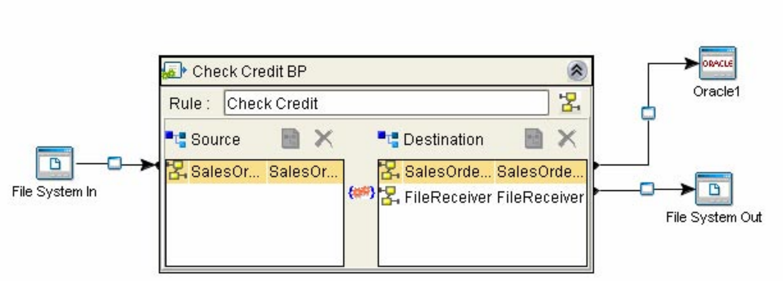


Opening a Page Flow in the Connectivity Map Editor

1. Double-click the Page Flow in the Connectivity Map to open the Binding Dialog.
2. Connect the Pages to the appropriate component.

- Receive elements will appear in the left pane.
- Invoke and Reply Activities will appear in the Right Pane.

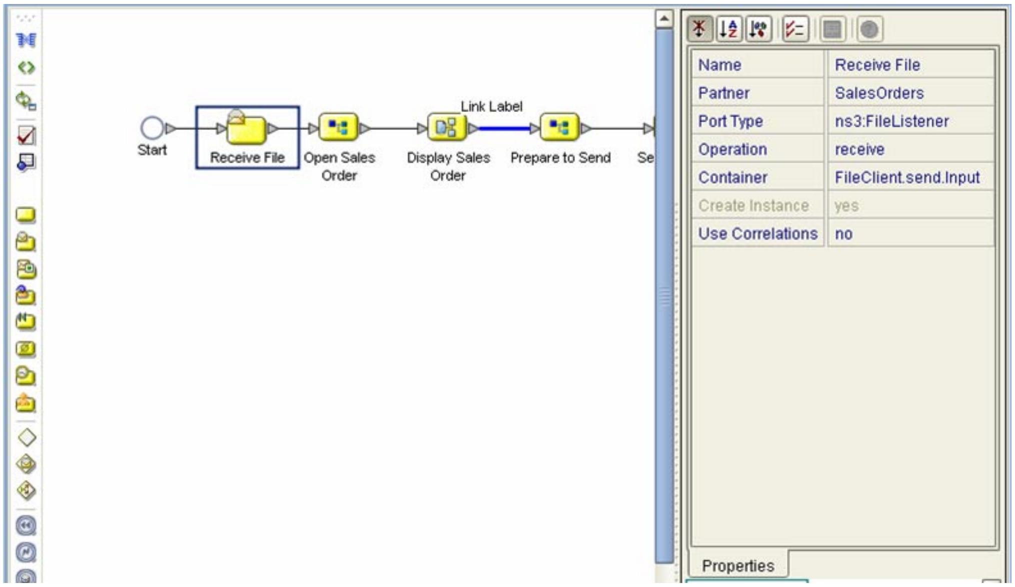
Figure 83 Connectivity Map: Page Flow Binding



How the Binding Information Appears In eVision

1. Select a Page in the Page Flow.
2. Select the Properties toggle.
3. The "Partner" is an abstracted ID for an external system that will appear in the Binding Dialog within the Connectivity Map Editor.

Figure 84 Binding Information



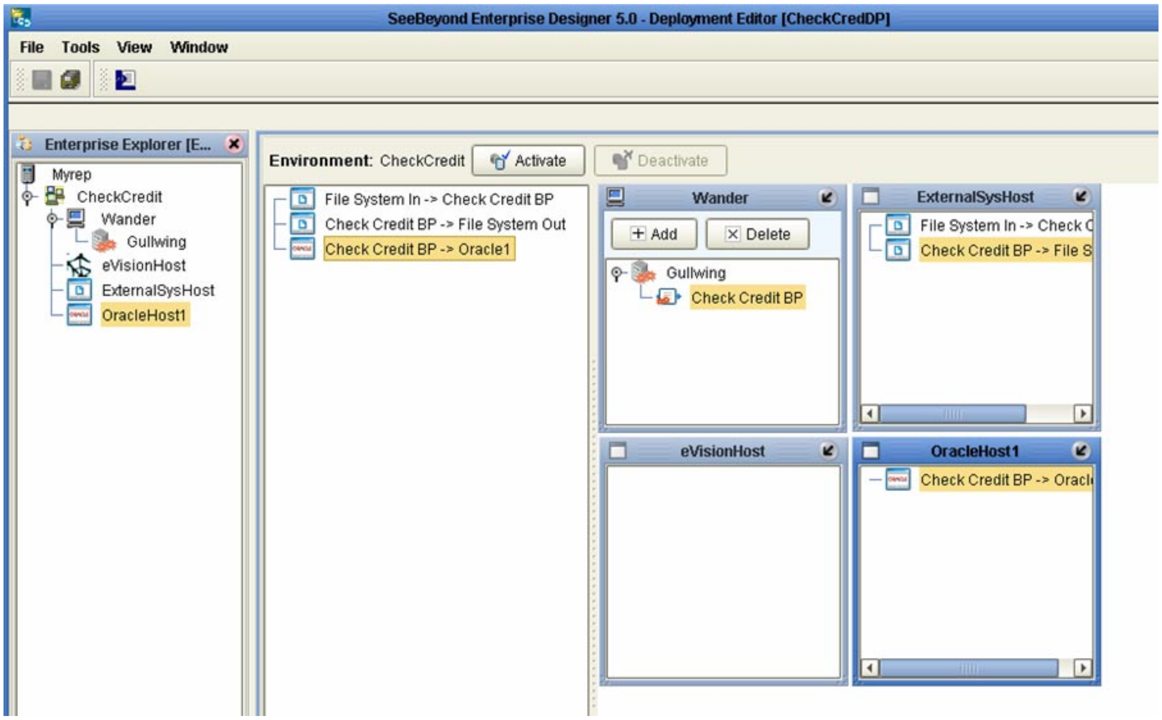
Name	Receive File
Partner	SalesOrders
Port Type	ns3:FileListener
Operation	receive
Container	FileClient.send.Input
Create Instance	yes
Use Correlations	no

10.4 Deploying a Page Flow

After creating your Deployment Profile, Environment, Logical Host, and other necessary Hosts, move to the Environment Explorer. The deployable components along with the Page Flow from the Connectivity Map will appear in the Environment.

Select **Activate** to complete the deployment of the components to the target hosts. Drag and drop the Page Flow to the desired Integration Server.

Figure 85 Deployment Editor


SeeBeyond Enterprise Designer 5.0 - Deployment Editor [CheckCredDP]
File
Tools
View
Window
Enterprise Explorer [E...
Myrep
CheckCredit
Wander
Gullwing
eVisionHost
ExternalSysHost
OracleHost1
Environment: CheckCredit
Activate
Deactivate
File System In -> Check Credit BP
Check Credit BP -> File System Out
Check Credit BP -> Oracle1
Wander
Add
Delete
Gullwing
Check Credit BP
ExternalSysHost
File System In -> Check C
Check Credit BP -> File S
eVisionHost
OracleHost1
Check Credit BP -> Oracl

Index